



Deliverable D32.3 *Factsheet #2* *“IoT Manager”*

Document Owner:	Jerry Dimitriou – SINGULAR
Contributors:	
Dissemination:	Public
Contributing to:	WP3.2
Date:	26/3/2013
Revision:	1.2

List of Abbreviations

IoT	Internet of Things
GEs	Generic Enablers
OMA	Open Mobile Alliance
ETSI	European Telecommunications Standards Institute
M2M	Machine-to-machine
NGSI	Next Generation Services Interface
GSN	Global Sensor Network
VSD	Virtual Sensors Description

Table of Contents

1	AVAILABILITY AND CONTACTS	4
2	ARCHITECTURE AND FUNCTIONALITIES	5
2.1	INTRODUCTION	5
2.2	REQUIREMENTS	5
2.2.1	<i>Functional Requirements</i>	5
2.2.2	<i>Non-Functional Requirements</i>	5
2.2.3	<i>Core Usage Scenario</i>	5
2.3	ARCHITECTURE AND IMPLEMENTATION NOTES	6
2.3.1	<i>Architecture overview</i>	6
2.3.2	<i>Internal components</i>	6
2.3.3	<i>External components</i>	8
2.3.4	<i>Technologies</i>	8
3	TECHNICAL INFORMATION.....	9
3.1	SERVICE RUNTIME REQUIREMENTS	9
3.1.1	<i>Operating systems</i>	9
3.1.2	<i>Application servers</i>	9
3.2	DEVELOPMENT ENVIRONMENT	9
3.3	TECHNICAL DETAILS	9
4	LICENSING	10
4.1	SERVICE LICENSE	10
4.2	THIRD PARTY LICENSES	10
5	TECHNICAL MANUAL	11
5.1	PREREQUISITE VERIFICATION	11
5.2	INSTALLATION	11
5.3	CONFIGURATION	11
6	USER MANUAL.....	12
6.1	ACCESSING THE PROTOTYPE	12
6.2	ADMINISTRATION UI	12
7	FUTURE PLANS.....	14
8	REFERENCES.....	14

1 Availability and Contacts

This table describes how to reach the prototype and the contact person in case of questions.

Version	1.0
Availability	Source is available from MSEE SVN repository
Accompanying specification and design document	Deliverable D32.1a: FI Platform Federation specifications and architecture – M12
Source control	svn://repo.nimbus-ware.com/MSEE/SP3/WP32/D32.3/trunk/iot-manager
Contact person	Jerry Dimitriou – ep6@singularlogic.eu

2 Architecture and Functionalities

2.1 Introduction

The purpose of the Internet of Things Manager (IoT Manager) is to operate as an abstraction layer for MSEE applications to reach devices in the MSEE ecosystems, providing access to sensors, actuators and other resources on embedded devices.

Taking into account the key requirements of pilot use cases for M18 prototypes, the first prototype of IoT Manager focuses on data acquisition and sensors management. The original IoT Manager architecture was designed with the intent to reuse Generic Enablers from the Internet of Things Services Enablement Chapter of FI-WARE project. However due to several factors pertaining to FI-WARE (technical architecture redesign, partners reshaping), access to FI-WARE Generic Enablers was not available for M18; therefore, given the focus on data acquisition for the first prototype, IoT Manager was redesigned to integrate with the Global Sensor Network (GSN) open source project (<http://sourceforge.net/projects/gsn/>). In terms of its API, IoT Manager still provides the same functionality originally designed. The IoT Manager design encapsulates the underlying implementation of data retrieval from remote sensors, therefore it will be possible in the future to switch from GSN implementation to another implementation of its backend services, while IoT Manager's API will remain the same and clients will operate normally.

2.2 Requirements

This module will provide to MSEE ecosystem services a gateway API through which IoT-enabled devices can be discovered, queried and managed.

2.2.1 Functional Requirements

Req. 1.1	Support devices management (registration, discovery etc)
Req. 1.2	Send commands from service to device <i>Note: Not implemented in the first prototype</i>
Req. 1.3	Collect readings from device
Req. 1.4	Receive status information from device
Req. 1.5	Device access must be authenticated and authorized <i>Note: authentication and authorization will be implemented in the second prototype, using SSO Utility Service</i>
Req. 1.6	Device access must be logged for service and platform administration purposes
Req. 1.7	Users must be provided with an administration user interface

2.2.2 Non-Functional Requirements

Req. 2.1	Performance
Req. 2.2	Robustness and fault tolerance

2.2.3 Core Usage Scenario

IoT Manager core usage is to provide MSEE ecosystems with access to federated IoT-enabled devices by means of a standard MSEE ecosystem service.

2.2.3.1 Device Management and Discovery

IoT Manager will track devices registered with the federated GSN middleware instances and provide browse and search facility so MSEE Ecosystem Services can dynamically discover devices they may be interested in for further interaction. Moreover, IoT Manager will provide a unified device addressing scheme, so MSEE Ecosystem Services may locate and communicate with devices across several federated GSN middleware instances.

2.2.3.2 Device communication

Service- device communication may happen according to the following use cases:

- Service requests data (service polls device)
- Service sends commands (service pushes to device)
- Service subscribes to updates from devices in order to receive data at the time they are available (pub/sub model)

Availability of the above modes of communication may vary with device capabilities, however IoT Manager will abstract these details from MSEE Ecosystem Services by providing a unified communication layer on top of the federated GSN middleware instances.

2.3 Architecture and Implementation Notes

2.3.1 Architecture overview

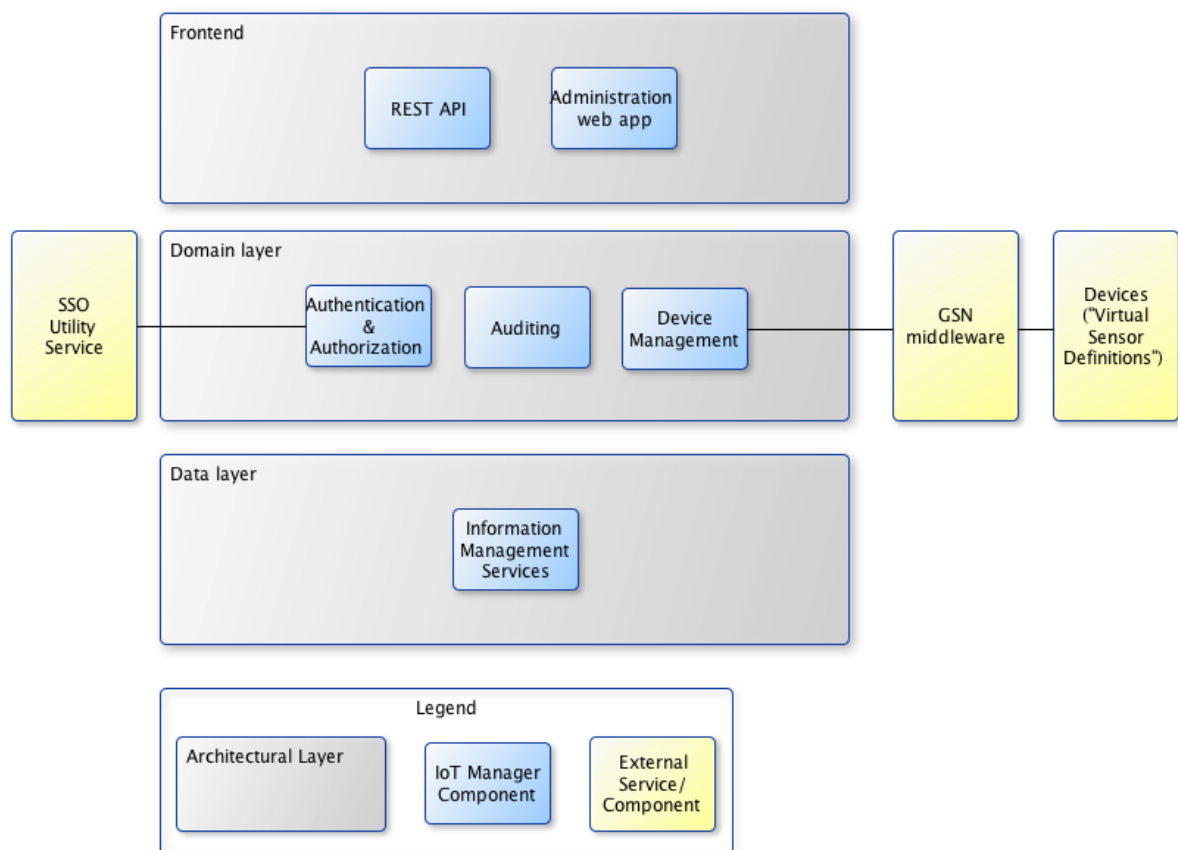


Figure 1 – IoT Manager logical architecture overview

2.3.2 Internal components

Device management

Device management interfaces with GSN middleware instances and supports all the device-related operations, involving discovery and registration of devices and reading of sensor values from devices. GSN provides the following integration points:

- A SOAP service that provides methods for
 - Management of Virtual Sensor Descriptions (VSDs)
 - Data retrieval, with advanced querying and data stream processing
- A callback-based HTTP streaming method
- A pure push, connection-oriented data push interface.

Device management: Device registration and discovery

Implements requirement 1.1

Devices in GSN middleware are specified via “Virtual Sensor Descriptions” (VSDs) in declarative XML. The Virtual Sensor is the core abstraction in GSN: it represents any IoT device or can serve as a processing element that obtains several input streams from other Virtual Sensors, applies some logic (e.g. a moving average algorithm implementation) and outputs a new stream of data.

Once registered, Device Management component may query GSN middleware to obtain registered devices and their available sensors.

Device Management: Collect readings from device

Implements requirement: 1.3

Data from devices will be obtained via the relevant SOAP methods of the GSN web service. Moreover, queries can be set and ‘subscribed to’, allowing for more complex processing to happen on GSN’s server-side.

Device Management: Receive status information from device

Implements requirement: 1.4

In GSN, device information are treated as regular virtual sensor data, and are available via the same SOAP methods for obtaining readings from sensors.

Auditing

Implements requirement: 1.6

Important events will be logged to persistent storage in order to provide full audit trail. These events include:

- Administration events such as changes to virtual sensors descriptions
- Device register/unregister
- Subscription updates
- Queries for data
- Commands sent to devices

Administration web interface

Implements requirement: 1.7

A web application will be provided that will support administration of IoT Manager instance:

- Register/unregister GSN instance available for federation
- Manage (create, read, update, delete) Virtual Sensor Descriptions per each GSN instance.
- View of instance's audit trail.

REST API

The IoT Manager exposes its core functionality via a REST API available to MSEE service developers that require access to remote sensors data. The REST API exposes the core business logic of IoT Manager, allowing the management of federated sensor networks, discovery and registration of sensors and data acquisition from remote sensors.

2.3.3 External components

IoT Manager uses GSN middleware open source project (<http://sourceforge.net/apps/trac/gsn>) to provide backend implementation of sensors data acquisition.

2.3.4 Technologies

The following table summarizes the tools and technologies which have been used to implement IoT Manager components per each layer:

Tools/Technologies	Role
MySQL	RDBMS for persistent storage of relational data
Apache Tomcat	Application server
JPA/Hibernate	Object-Relational Mapping tool for basic data access
Spring	IoC container
Apache CXF	SOAP and REST services client and server framework

3 Technical Information

3.1 Service runtime requirements

3.1.1 Operating systems

The IoT Manager service can be executed on any OS that supports Java (e.g. Windows, Linux or Mac OS X). It depends on at least one GSN middleware instance, which is also a Java service instance, therefore it is OS-independent as well.

3.1.2 Application servers

IoT Manager is executed in a servlet container such as Apache Tomcat or Jetty.

3.2 Development environment

IoT Manager uses Maven for build & dependency management

3.3 Technical details

Nature	Source code
Programming Language	Java
Development Tools	Maven
Additional libraries	Spring Apache CXF JPA/Hibernate GSN
Application Server	Apache Tomcat 7 or other servlet container
Database	MySQL

4 Licensing

4.1 Service license

To be defined.

4.2 Third party licenses

Third party software	Licence
ORACLE Java Runtime Environment 7	http://www.oracle.com/technetwork/java/javase/terms/license/index.html
GSN	GNU GPL v2.0 http://sourceforge.net/apps/trac/gsn

5 Technical Manual

5.1 Prerequisite verification

A MySQL database installation is required both for GSN and IoT Manager. MySQL can be downloaded from <http://www.mysql.com/downloads/>.

Also, since both GSN and IoT Manager run on the Java platform, the latest version of Oracle JDK 7 is required on the deployment machine.

A servlet container is required to host the IoT Manager. Apache Tomcat 7 has been tested successfully, however other servlet containers such as Jetty should work.

Finally, at least one GSN installation is required for IoT Manager to provide meaningful functionality. GSN source code or binaries can be obtained from <http://sourceforge.net/projects/gsn/>. Installation and configuration instructions for GSN are provided on GSN website (<http://sourceforge.net/apps/trac/gsn/wiki/install-gsn>).

5.2 Installation

IoT Manager is distributed as a single WAR archive, deployable to a servlet container. Consult your servlet container's documentation for exact instructions on how to deploy a WAR archive. For example, for Apache Tomcat 7, deploying IoT Manager is as simple as copying the WAR file to Tomcat's webapps folder.

5.3 Configuration

IoT Manager may require some configuration with regards to the database connection to MySQL server. In case you need to customize database access, open file `${TOMCAT}/webapps/iot-manager/WEB-INF/classes/database.properties` in a plain text editor and change the following lines appropriately:

Property name	Value
jdbc.url	JDBC URL of MySQL instance, including hostname and port on which MySQL is installed, as well as database name in the format below: <code>jdbc:mysql://HOSTNAME[:PORT]/DB_NAME</code> <i>Default: MySQL hosted on localhost with DB_NAME 'iot-manager'</i>
jdbc.username	Username of MySQL user with read access to database <i>Default: msee</i>
jdbc.password	Password of MySQL user <i>Default: msee</i>

6 User Manual

6.1 Accessing the prototype

The IoT Manager prototype provides an administration UI at <http://localhost:8080/iot-manater/admin>, assuming it is deployed on a default Tomcat installation at localhost.

6.2 Administration UI

The administration UI of IoT Manager provides basic functionality to manage federated GSN instances and their associated Virtual Sensors. Administration of GSN instances is displayed in Figure 2.

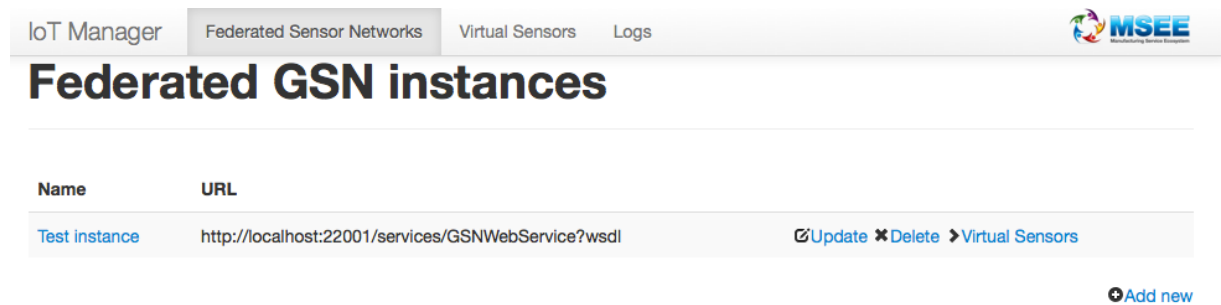


Figure 2 Administration of federated GSN instances

For each GSN instance, a name and service URL is provided, via which IoT Manager manages virtual sensors. User can perform all standard administration operations (create, read, update, delete) for GSN instances.

For each GSN instance, a list of its registered Virtual Sensors is provided. Each Virtual Sensor is configured via an XML file with a schema described in the following PDF:

<http://gsn.svn.sourceforge.net/viewvc/gsn/branches/documentations/book-of-gsn/chapters/ch-quickref/figures/vs-quick-ref.pdf>

Each sensor has a name and its own VSD XML which defines the internal structure of the sensor and its data streams.

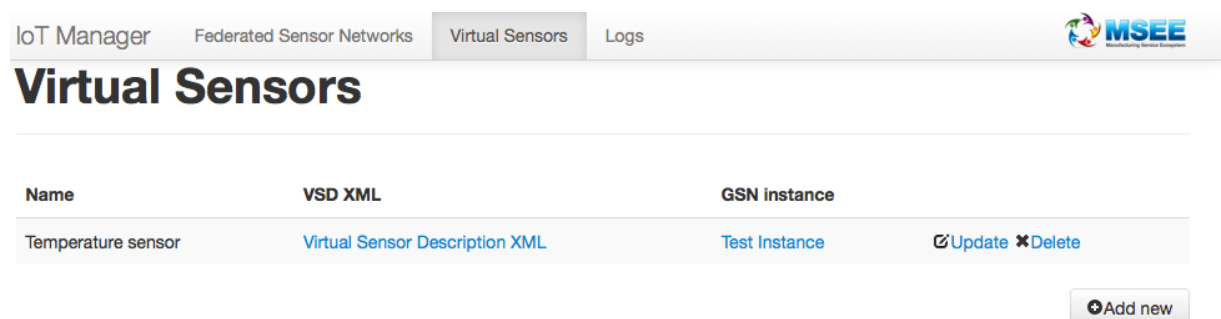


Figure 3 Virtual sensors management

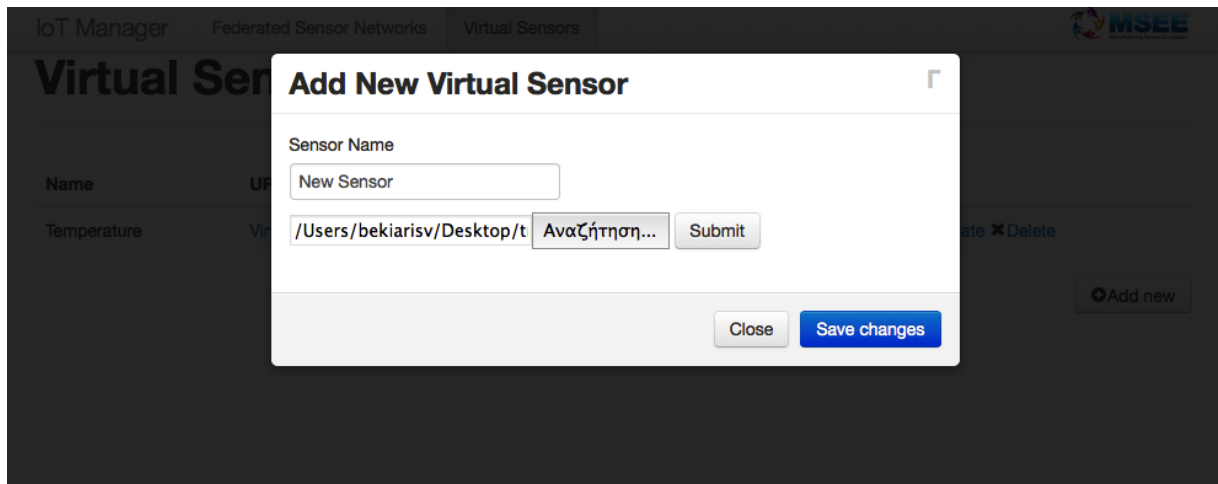


Figure 4 Adding a new Virtual Sensor Definition

Adding a new sensor, as long as required protocol adapters are already available for managing actual communication with the device, is a matter of providing a name for the Virtual Sensor and its XML VSD file.

Finally, the Logs view provides a complete view of actions that have occurred in IoT Manager.

7 Future plans

Plans for the second prototype of IoT Manager include:

- Implementation of access control, using SSO Utility Service infrastructure
- Re-evaluation of FI-WARE status and possibility to use FI-WARE GEs as alternative backend, including both sensor data acquisition as well as command & control

8 References

D32.1 – FI Platform Federation Specification and Architecture

GSN Wiki - <http://sourceforge.net/apps/trac/gsn>

GSN Book - <http://gsn.svn.sourceforge.net/viewvc/gsn/branches/documentations/book-of-gsn/main.pdf>

GSN WSDL reference -

<http://gsn.svn.sourceforge.net/viewvc/gsn/branches/documentations/misc/gsn-webservice-api.pdf>

GSN push service reference -

<http://gsn.svn.sourceforge.net/viewvc/gsn/branches/documentations/misc/gsn-datadistribution.pdf>