SAVE ENERGY
Grant agreement no.: 238882

DELIVERABLE

Deliverable D 4.2

Demonstrator for the energy efficiency management platform

Document ID	D4.2.31.08.10
Authors:	Nuno Martins (ISA)
Contributors:	ISA, CeTIM, ALFAMICRO
Workpackage:	WP4
Version:	7.0
Date:	31.08.2010
Project Coordinator:	ALFAMICRO
Contract Start Date:	01.03.2009
Duration:	30 months
Dissemination Level:	Private

Project co-funded by the European Commission within the ICT Policy Support Programme

Revision History

Date	Version	Author/Contributor	Comments
03-12-09	V01	Nuno Martins (ISA)	Document creation
29-01-10	V02	Pedro Oliveira (Alfamicro)	Comments to the structure and contents alignment
16-02-10	V03	Erwin Marges (CeTIM)	Code for integration development
22-02-10	V04	Nuno Martins (ISA)	Integration of all comments and revision of the document
29-02-10	V05	José Luís Malaquias (ISA)	Text revision, introduction, and conclusions.
26-08-2010	V06	Nuno Martins (ISA)	Revision, according to the Reviewers comments
31-08-2010	V07	ISA/ALFAMICRO	Final Version

Statement of originality:

This deliverable contains original unpublished work except where clearly indicated otherwise. Acknowledgement of previously published material and of the work of others has been made through appropriate citation, quotation or both.

Copyright

Table of Contents

Table of Figures

Table of Tables

Executive Summary

As explained in previous deliverables, the aim of the Save Energy Project is to show how making users aware of their energy consumption, in tandem with a healthy competitive spirit, can lead to potential net energy savings, as the users have a feedback on the effects of their day-to-day behaviour.

The five different pilots were purposely chosen to be quite heterogeneous, with different climates, different types of users, and very different building architectures. The goal was to show how different public buildings could be homogenized in a coherent energy saving benchmarking platform.

While it was not possible to achieve the goal of using a common hardware platform in all 5 different pilots, it was nonetheless considered to be a fundamental goal to integrate the data into a common data framework. To our knowledge, this unification of energy efficiency data from multi-vendor, multi-setup, multipurpose installations had never been tried in a European funded project, and it was thought to be not only a challenging goal, but also one that would prove to be useful in the dissemination phase as a case study on the integration of technologies.

The present document shows how ISA's iCenter middleware platform was used and adapted as a least common denominator that would aggregate multi-origin, multi-parametric data. The iCenter thus acts as a data sieve where data is categorized into different types, and where data can be made uniform to be used by upper common levels of the Save Energy architecture, such as the Serious Gaming platform, and other data presentation applications-

The following document intends to be a support document for the iCenter usage. The iCenter is an ISA service designed to receive data from remote metering locations. It has a tagging feature that will allow the project to document and homogenize the data from the five pilot locations. In the present document, an introduction to ISA's iCenter will be made, together with a description of its main functionalities. In the annexes section, there is a manual for data interaction between the pilot owner and the database.

Furthermore, ISA has developed for the SAVE ENERGY project an open version of the iCenter, from now on described as **iCenter Open Source** (OS). The code package will be published at SAVE ENERGY web page, along with the integration documents, so that the community can use and develop the proposed system. From our point of view, the release of such a software piece, can trigger the usage of a versatile middleware in other projects as well, the improvement of the existing platform.

1 Introduction

SAVE ENERGY will use a Serious Game providing an engaging, believable, virtual environment for citizens, consumers and policy makers to gain awareness, understanding and experience regarding energy efficiency issues and behavioural change, in both the immediate and longer terms.

The project will deliver a model that is simple, attractive and effective enough for users to actively and consistently engage with. Pilots will involve public buildings from 5 major European cities (Helsinki, Manchester, Lisbon, Luleå and Leiden) that will demonstrate the extent to which tangible emissions reduction is achievable.

All 5 pilots will use electronic sensors to measure energy usage, plug adapters between wall sockets and devices that will measure the energy consumed, and communicate to a local “gateway” system with a Web interface.

The information of a network of sensors will be gathered by a central server allowing data analysis, identification of consumption patterns and real time overview. The consequent analysis of the presented features will allow the partners to sketch action plans to optimize the energy usage and, hence, the carbon footprint. A given public space will become informed of their energy profile and be given advice on adapting behaviour.

Pilot implementation will follow the Living Lab methodology. The intended project involves large communities of motivated citizens co-creating ideas, decisions, and recommendations in an open innovation environment. The core group of Living Labs will collaborate during the project, and it will launch a Thematic European Network of Living Labs cities, focused on Energy Efficiency and Sustainability.

European-wide dissemination of results will be provided to key stakeholders to aid Policy change, market impact and broader knowledge in Energy Efficiency solutions.

The goal of this document is to provide a definition of the iCenter OS platform, that will have the role of the central server in order to standardize and harmonize information.

2 Architecture

As designed in the previous tasks of the SAVE ENERGY project, a common high level architecture was projected for all five SAVE ENERGY pilots.

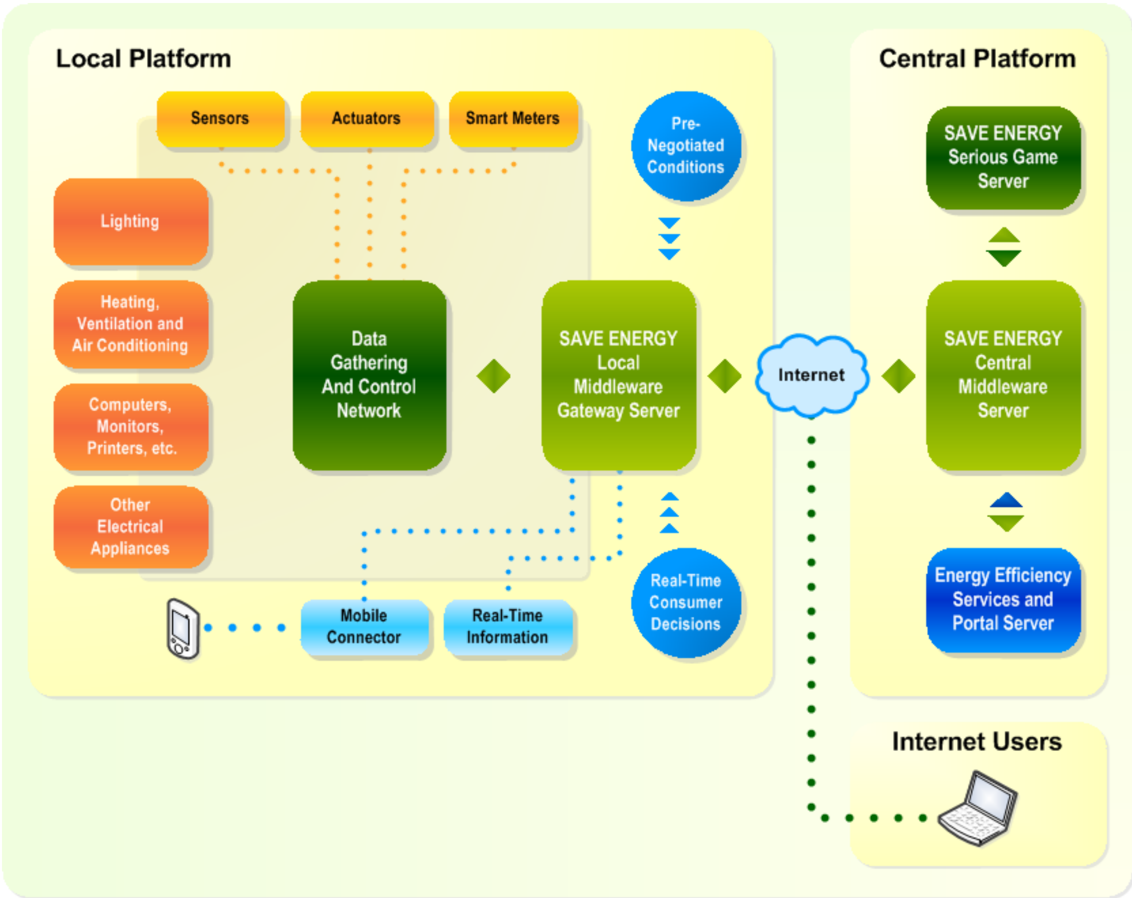


Figure 1 Common high-level technical architecture

The previous image, displays the envisaged high level architecture for the SAVE ENERGY pilots, the middleware layer that we are describing in the present deliverable sits in the Central Platform layer. The iCenter OS service will provide the ability to concentrate all gathered data in one physical spot.

ICenter is an ISA service, and comes to the project in an open source version, so it can work as a data harmonization solution. The iCenter OS will be the unifying middleware that will provide:

- Integration with the Pilots' local middleware;
- Interoperability with End-user Services;
- Interoperability with the Serious Game.

The architecture used in the iCenter OS solution has a classic layered design approach. At a higher level, it's composed of three layers, as shown below:

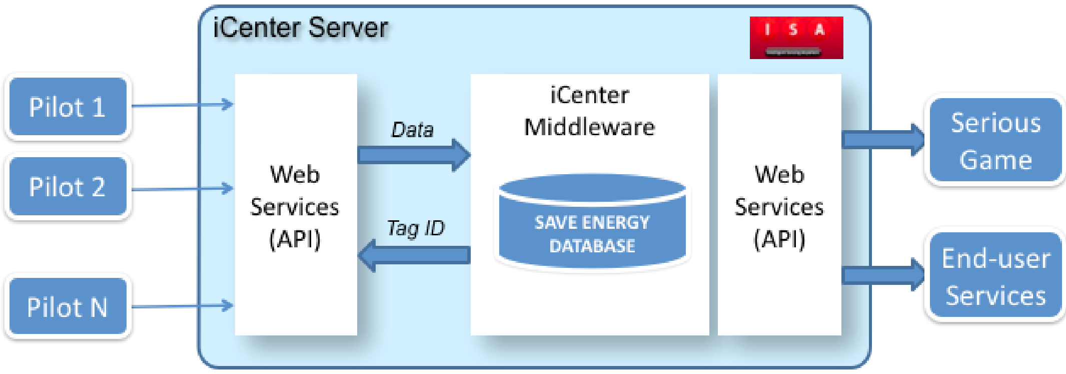


Figure 2 Central Platform: iCenter OS

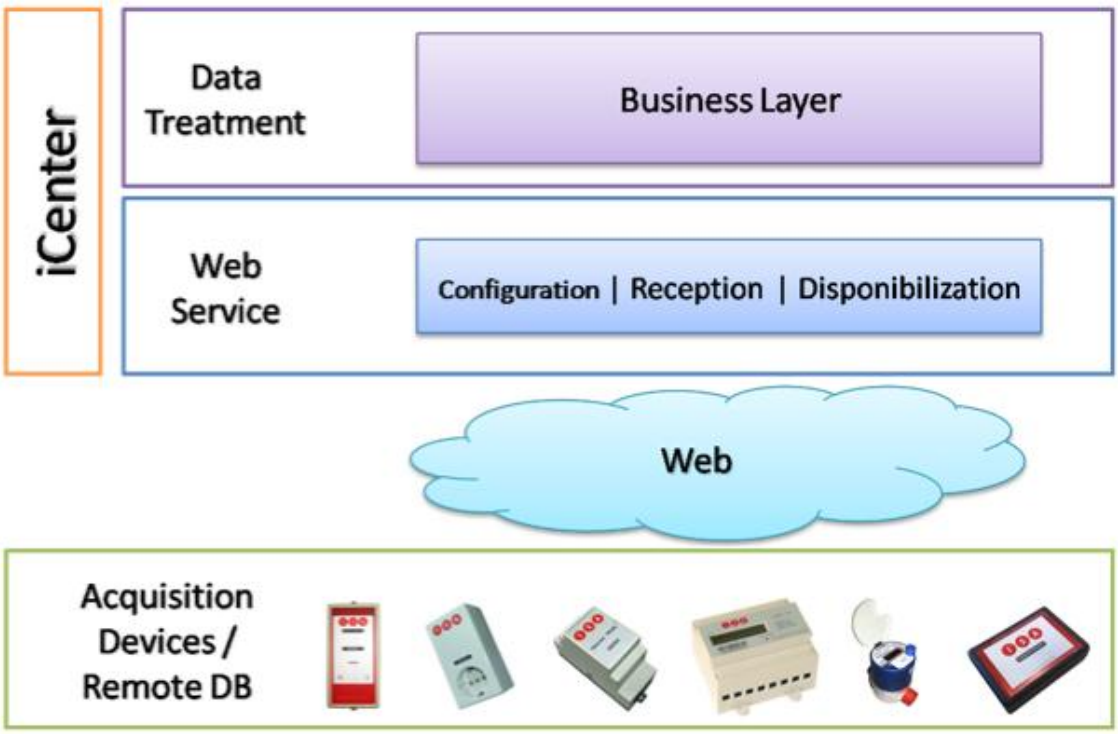


Figure 3 iCenter OS Architecture

The layers defined create multiple abstraction levels that provide flexibility to the software. A brief description of each layer and functionality follows:

Acquisition: this layer is the lower layer and is responsible for the data acquisition in the field. Here, the devices will measure various parameters such as energy, temperature, humidity, etc. This layer is also responsible for reporting the data to the next layer. This communication can be done by each acquisition device or by a central point, collecting the data and reporting it to the server. This layer is pilot-specific, even if a common general architecture was defined;

Web Service: this layer, as the name suggests, is a Web service the pilots can access for delivering data. Here, the system's users can configure the data structure, report, and also access data;

Business Layer: this layer is responsible for all the business logic, such as data unification, data aggregation, unit conversion, format adaptation, among others.

3 Data Structure

The data structure used for the iCenter OS solution to represent the measurement environment deployed in each place is a tree hierarchy with three levels, displayed below:

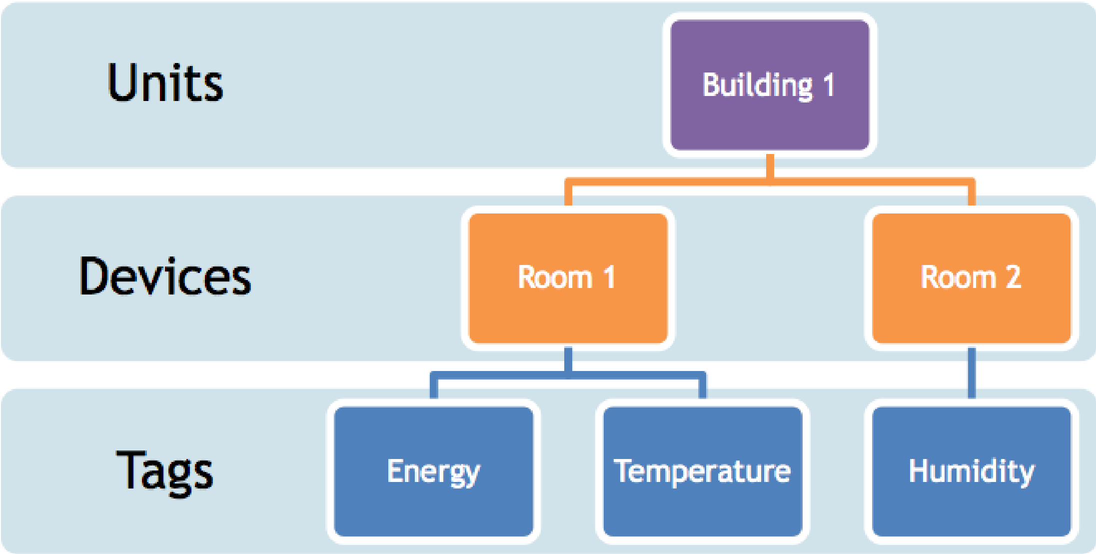


Figure 4 Data Structure

In the previous figure, the three levels are represented and a possible instance is also represented (note: the logical meaning of each level is the user's responsibility, the structure presented is only given as an example). The three levels are:

Unit: the unit is the first logical aggregation level. It usually represents the point where the data of a set of devices is concentrated and where the data is sent to the iCenter OS;

Device: this level is the second logical aggregation level. It can represent a hardware device (i.e. a single device with two sensors, such as temperature and humidity that would correspond to two tags);

Tag: this level is the most elementary level. This means that a tag represents a unique point of measure for a certain physical parameter in a well-defined measurement unit.

3.1 Integration of Objects

The objects used in the Web Service contain properties that may not be used in the Save Energy environment. The mandatory are described below (a description of the meaning of each property can be consulted in the specification of the web service):

- **Unit**
  - Id – Mandatory in edit;
  - Name – Mandatory;
  - Interval – Mandatory. Use 0 if not used in the system;
  - RemoteId – Mandatory. Use 0 if not used in the system;

- Serial – Mandatory. Use a string with at least one character

- **Device**
  - Id – Mandatory in edit;
  - Name – Mandatory;
  - TypeId – Mandatory; Consult Appendix
  - UnitId – Mandatory;
  - Interval – Mandatory. Use 0 if not used in the system.
  - RemoteId – Mandatory. Use 0 if not used in the system.
- **Tag**
  - Id – Mandatory in edit;
  - Name – Mandatory;
  - Type – Mandatory; Consult Appendix 2
  - DeviceId – Mandatory.
  - RemoteId – Mandatory. Use 0 if not used in the system

4 Web Service

4.1 Methods Available

The Web service available as an external interface to use the iCenter OS platform was built using the .Net WCF technology. The service is made available as a standard Web service through the http protocol, and it is compatible with any programming language or any operating system.

A specification of the full interface can be found in the document named “iCenter OS - Data Provision”.

There are several methods available in the Web service, as defined in the document. However, for the purposes of the Save Energy project, only the following methods will be needed and made available:

- **Hardware Management:**
  - Insert Unit
  - Edit Unit
  - Delete Unit
  - Insert Device
  - Edit Device
  - Delete Device
  - Insert Tag
  - Edit Tag
  - Delete Tag
  - Get Tags
- **Data:**
  - ObtainConsumptions
  - BulkInsertData
  - RequestSegmentedBulkInsertData
  - GetConsumptions
  - SegmentedBulkInsertData
  - CommitSegmentedBulkInsert
  - RollbackSegmentedBulkInsert
  - DeleteData

4.2 Security

The Web service security is ensured in two ways:

User authentication with a secure token based on username, password and a public shared key;

Using an SSL certificate encrypting the channel and communicating through https.

5 Front End

The iCenter database for Save Energy may be accessed at:

www.isa.pt/SaveEnergy

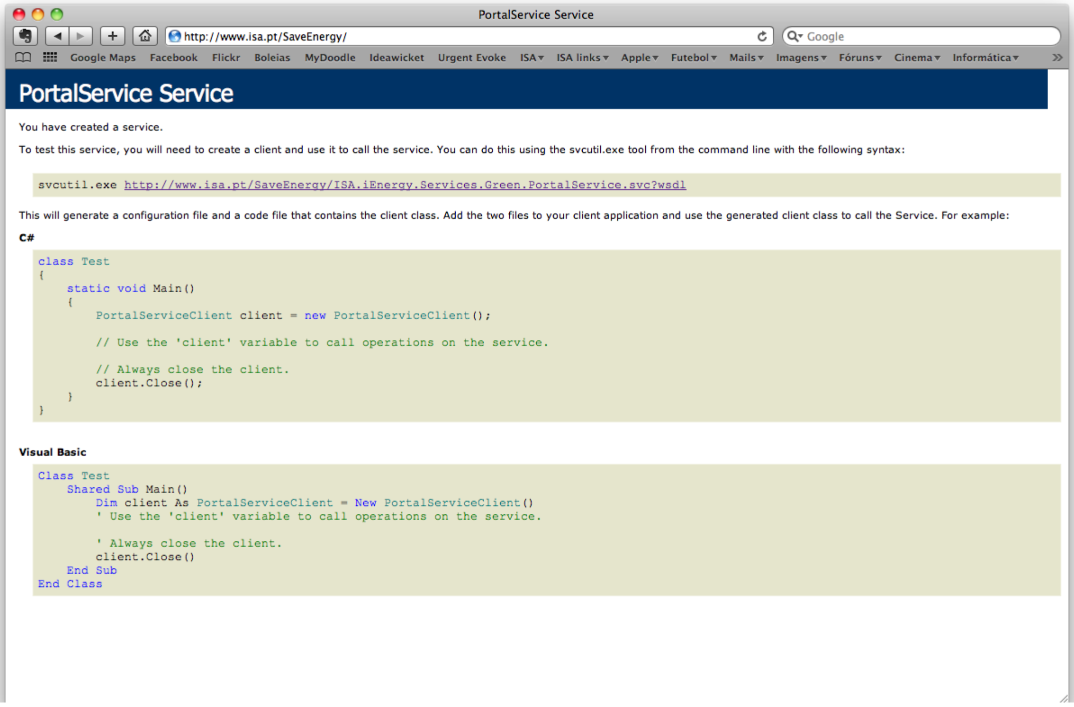


Figure 5 Front end of iCenter OS service web address

The login can be made from each pilot, using one of the five accounts created for the purpose:

- lisbon_pilot
- helsinki_pilot
- lulea_pilot
- leiden_pilot
- manchest_pilot

Apart from the five pilots, there are also two accounts for data downloading, namely for the Serious Games interface, and End-User Services:

- greenmyplace
- markoremes

6 Specification of Resource and Device Connector Layer

Initially thought having in mind a unified network for all five pilots, the developments of the project has turned into another paradigm, where five different ICT solutions were effectively deployed. This table allows us to summarize the five realities at each pilot, regarding the data collection and forwarding to the server. A major challenge that always comes to attention, when deploying an ICT solution for Energy Monitoring, is the versatility of the solution. The wireless technology allows us to enhance the possibilities in terms of scalability, reliability and installation easiness.

At the moment, each pilot has a wireless structure functioning for data acquisition:

Table 1 Pilots wireless used technologies

Pilot	Technology	Variables Measures	Comments
Helsinki	ZigBee meshed network. Protocols in use: ZigBee, BMS system, Manual measuremnts	Temperature, Local Electricity Consumption, CO2, Moisture, Illuminance	The data is collected by a ZigBee meshed network of wireless sensors, that forward the data to a TGG03 router
Lisbon	ISA integrated solution Protocols in use: 868 MHz proprietary, ZigBee	Electricity, Plugs, HVAC, CO2, Temperature, Humidity	The variables collected are using a wireless meshed network operating in a 868 MHz proprietary protocol and ZigBee for submetering.
Luleå	KYAB integrated solution Protocols in use:	Total electricity consumption, DHW, CW and DHWC	The metering is done using M-Bus and 1-wire that are aggregated in a SABER unit which is then forwarded via TCP/IP to a database
Leiden	Plugwise and AnyWi solution	Submetering, lightning, heating and windows	The Leiden pilot uses Plug meter for sub-metering tasks, this data is then crossed

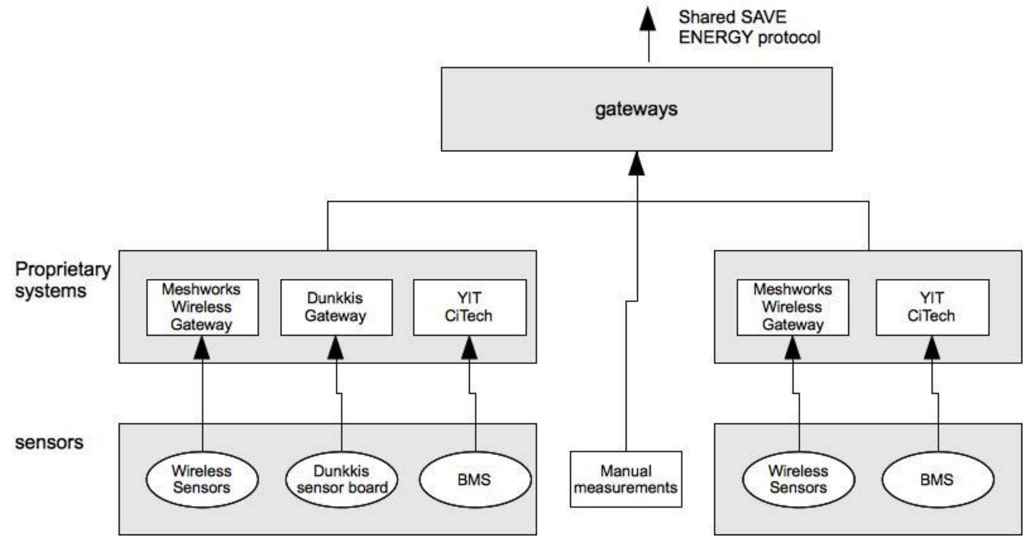
			with other variables using different protocols depending on the variable
Manchester	Schneider solution installed by Samelco	Ambient ligh, temperature and Total electricity	The fuse box of the monitored block is being monitored by a branch power meter and stored at a local mini PC.

At the moment all the networks are running and transmitting data to the local database server.

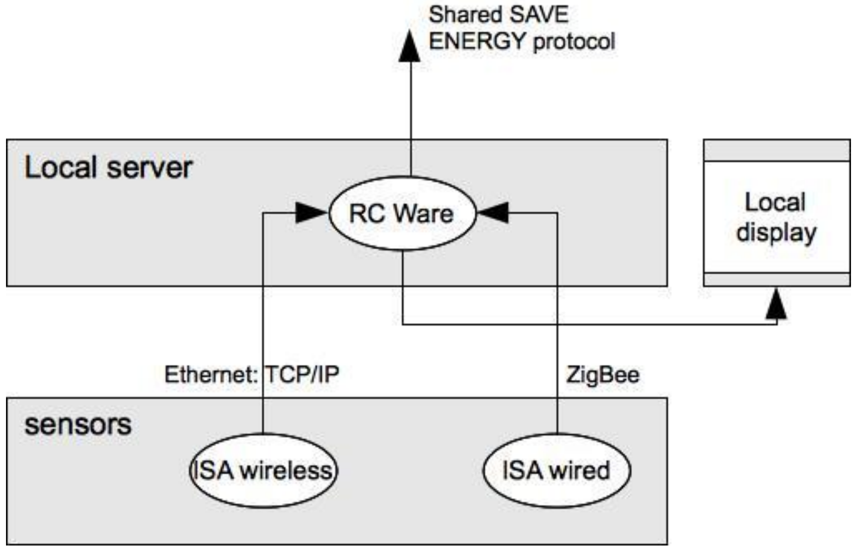
The inclusion of the study of such wireless networks, will demonstrate that, even with different ICT solutions, it is still possible to obtain data homogeneity. Although there are different, context-dependent, variables being measured, it is possible to harmonize the data. The collected data from the sensor networks allows the pilots a versatile environment where to gather the needed variables, in order to achieve the best performance indicators on Energy Efficiency.

The developments in all pilots have shown us how to generate a framework of interoperability with solutions that centralize the information in a local database/server.

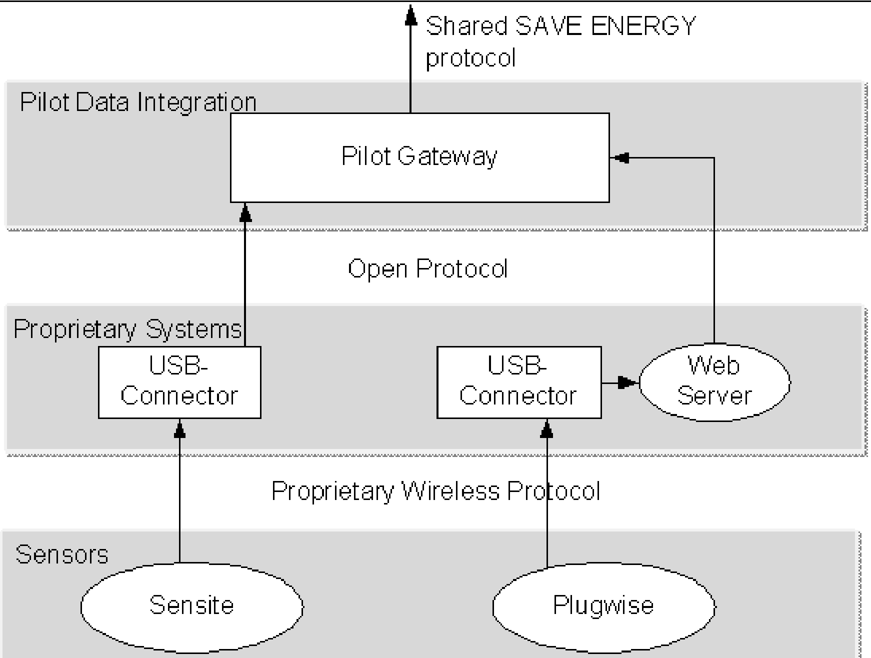
Table 2 Middleware protocol integration. From five pilots to one project



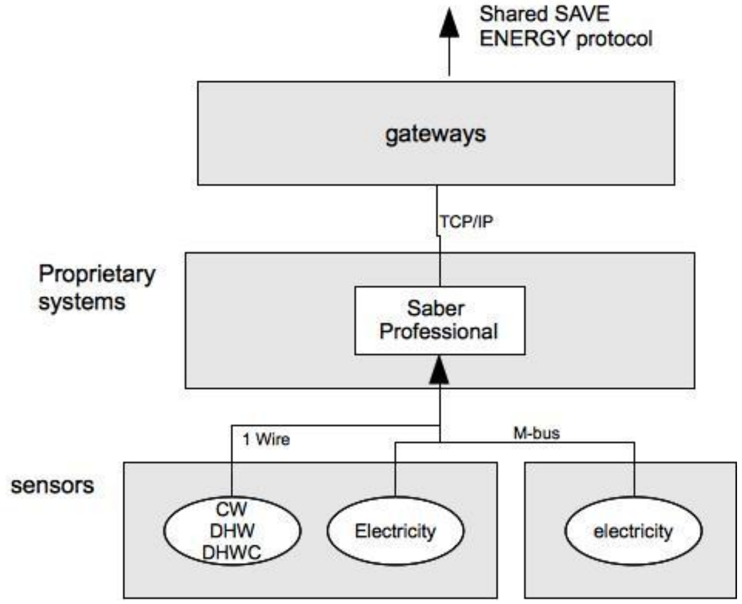
Helsinki communications and connectors schematics



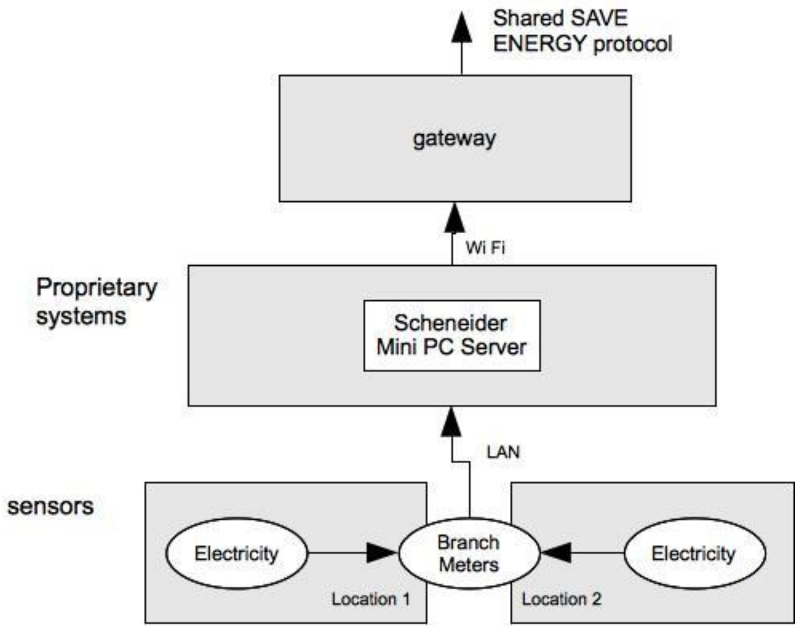
Lisbon communications and connectors schematics



Leiden communications and connectors schematics



Luleå communications and connectors schematics



Manchester communications and connectors schematics

The previous scheme, with multiple protocols for variable measurement is applicable to each pilot, and details may be found detailed at the Pilot Implementations Plan Deliverable description. At the time of this task a spreadsheet for each solution will be created in order to summarize the ongoing interactions and the respective pros and cons of each solution.

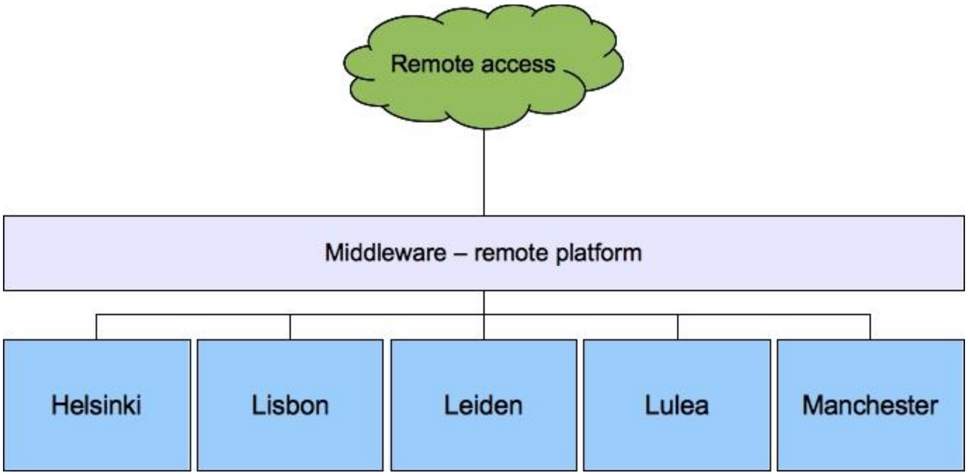


Figure 6 Middleware integration between pilots

At the moment, each sensor network and respective connectors are working and gathering information. In the following months, each solution will be refined.

7 User Representation Layer

The interface between the users and the ICT system will be one of the key elements to address during the behavioural change component of the project. It has been widely discussed that the way in which the information is presented, can be a critical trigger on how to induce energy savings. Thus, and focusing the current project on its founding principles, the user will be addressed throughout several platforms.

One of the most important features that will encourage the users to save energy will be the Serious Gaming platform.

The Serious Gaming will use real data, collected in all five pilots. This will create a connection between the user and his or her own pilot, therefore motivating and compelling the users to play the games, and be a part of the experience. The knowledge regarding the performance of the users' pilots, when compared to other pilots, will enhance a healthy competition and eagerness to play the games, while improving each pilot's performance.

One of the concerns regarding the present Task is the non uniform solution that has been deployed in all five pilots. Remember that, originally, a unique ICT solution was sketched for the whole project. Since it was not possible to address the envisaged ICT solution, five different ICT platforms were deployed in each pilot. This path allowed each pilot to address its own community of users in a very specific way, but it created a data uniformity problem. Since a school environment and its end users are not the same as the employees of a Municipality or the visitors of a Cultural House, different pilots required different solutions. Therefore, different ways of presenting the data and stimulating the savings are contemplated. In the following paragraphs we will approach and present several types of data display and interaction with the user.

Each pilot considers in its architecture a local server database, which will store the collected data. This unit will allow an immediate processing and forwarding of the data to the different types of displays. Afterwards, the same data is also provided to the iCenter OS platform. This flow of information will allow the Serious Gaming to be fed with near real-time data. This way, the pilot will not only have access to its own data, but to all pilots' experimental data.

All five pilots have been developing local display models and artefacts that will address the issue of conveying energy performance to the users.

7.1 The Helsinki Pilot

The **Helsinki pilot**, since its running in two schools, there are special needs to be attend, due to curriculum approaches and the group age. Nevertheless, the following materials have been developed and implemented:

Energyreview: 2008 energy consumption	Electrical energy	Heat energy	Cooling energy
Measured values	279 000 kWh	93 000 kWh	287 000 kWh

EE-value		January 2008 (example)
- 120	A	
121 - 150	B	129
151 - 190	C	
191 - 230	D	
231 - 300	E	
301 - 400	F	
401 -	G	
kWh/brm²/a		

Figure 7 Pihkapuisto elementary school

	Sähköenergian kulutus per oppilas	Energiankäytön CO_2-päästöt per oppilas
Tänään	2,9 kWh/hlö	2,0 kg*CO_2/hlö
Keskiarvo	3,6 kWh/hlö	2,5 kg*CO_2/hlö

Ma	Ti	Ke	To	Pe	Pisteet
28	29	30	31	1	350
4	5	6	7	8	250
11	12	13	14	15	150
18	19	20	21	22	100
25	26	27	28	29	350
1	2	3	4	5	300

Tiesitkö, että voit säästää energiaa: sammuttamalla valot ja sähkölaitteet lähtiessäsi

Figure 8 This simple display using the smiling faces per day gives the information of the electricity energy consumption of the whole school for all the pupils and teachers.

Figure 9 Pihkapuisto elementary school: mobile

7.2 Lisbon Pilot

For the Lisbon Pilot, the same local display will be used. The visual and daily stimulation of the users is one of the main drivers for the energy saving and performance improvement.

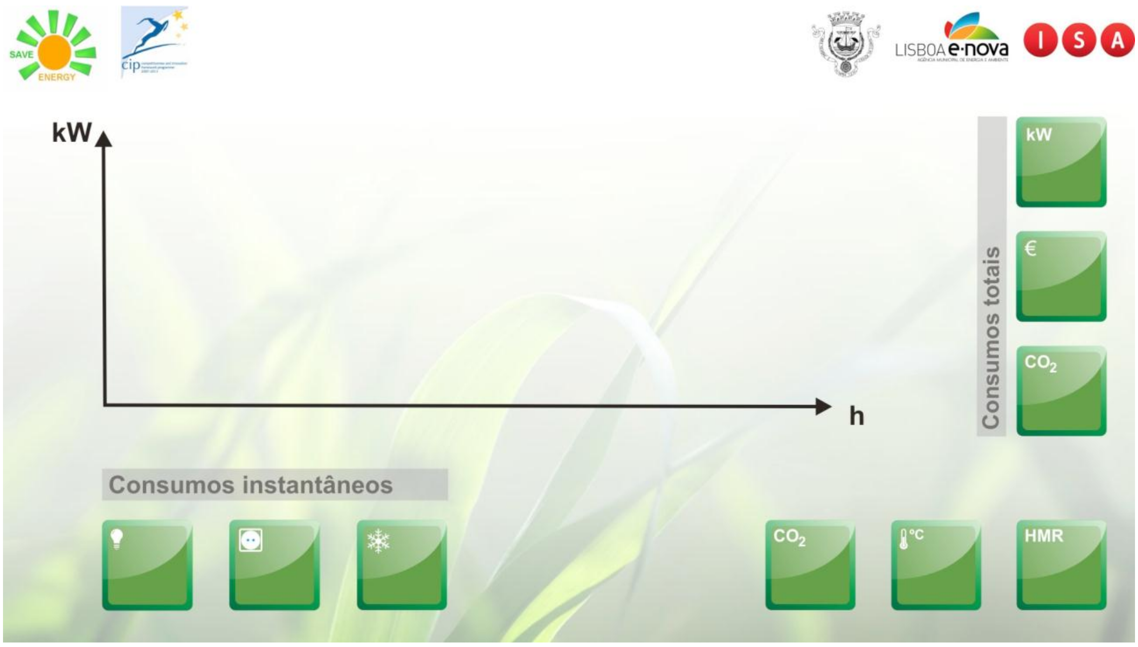


Figure 10 Lisbon Real Time Data Display

Along with the LCD continuously displaying the data, the Lisbon users will also be addressed with small applications that will run on their desk computers. The applet, will provide tips to the users accordingly with the Pilot performance.

Figure 11 Lisbon applet

7.3 Leiden Pilot

At the Leiden Pilot, energy consumption is made available for the users on a web portal. This portal provides graphical comparison on energy information of rooms and individual desks. Also, daily emails are sent to the users in order to inform them briefly about their energy usage and also to attract them into visit the web portal. The portal is developed in such a way that it can be viewed on modern mobile devices such as iPhones, and others.

Regarding the Leiden pilot, some analysis have been done, and the results disseminated:

Figure 12 Workstations in Leiden pilot, consumptions per room

Figure 13 Further analysis on consumptions in Leiden room number 1

7.4 Luleå Pilot

Luleå, has focused it's efforts also at the local real time data displays:

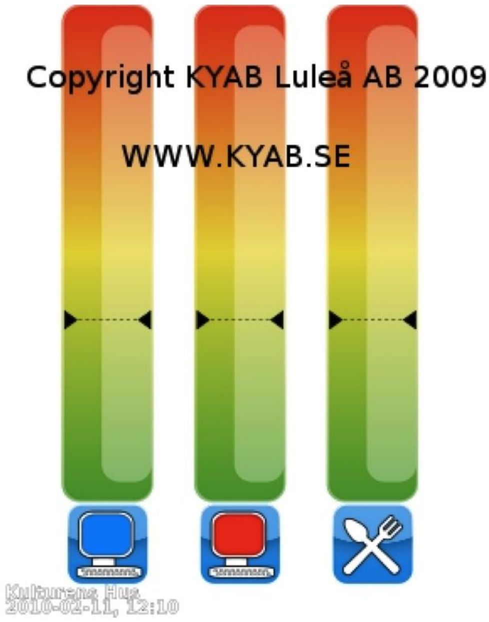


Figure 14 Local real time display. Luleå Culture House

Along with a static platform, efforts have been done to develop a mobile platform for data display:

Figure 15 Luleå mobile visualizer

At the moment, Manchester is tackling the last installation issues, but it has already made some developments at its own data display.

All the previous data display formats converge on the Serious Gaming platform. The harmonization of the data will be undertaken by the iCenter platform. The collected data will be forwarded to ISA servers, and then pulled by the Serious Gaming platform, thus creating a unified workflow for the data.

8 Conclusion and Final Remarks

The Save Energy consortium has proven to be very resilient, and capable of adopting contingency solutions when faced with important difficulties.

The needs of each pilot have proven to be too different from each other, and no vendor's solution was capable of handling all the differing requisites of each individual pilot. Thus, it was proven to be inevitable to adopt differing data acquisition technologies in each one of the pilots.

However, while proving to be a much more challenging task - to unite the pilots at the data level, rather than at the hardware level - it has proven as well to be a much more rewarding path. In effect, the chosen approach has proven that it is possible to build common energy efficiency solutions from a heterogeneous set of hardware solutions. From a European Energy Efficiency Industry viewpoint, it has been proven that different vendors can be made to work together towards a common goal - reducing energy consumption, in this case, by making users aware of their total energy usage.

The ISA iCenter OS middleware has proven to be a flexible reliable solution that is capable of handling a wide array of needs, receiving very large amounts of real-time data, and storing that data in a tagged form where, later, robot applications can apply transformations in order to unify it towards a common usage need.

The ISA iCenter OS has also proven to be capable of providing data access services to the applications above it, whether they be a Serious Gaming platform or different forms of user data presentation (Web portal, user applets, remote displays,...)

9 Technical Annexes

The technical annex contains the following information

- **PHP code for iCenter OS integration:** The code, developed by CeTIm, allows each pilot manager to create all the needed entities for data uploading.
- **iCenter OS Data Provision:** It's a general document created by ISA that summarizes the actions needed to interact with iCenter OS. It's a general and protocol agnostic document, it's a walkthrough for integration. The Data provision document can be found as separated document.
- **Test Report:** A test report is also provided, in a separated document, with an analysis to the supporting infrastructure of iCenter OS.
- **Code Package:** Along with the document, it will be published the code packinging for iCenter OS usage. You may found it at SAVE ENERGY project webpage

9.1 PHP integration:

In order to have a successful integration from a standard pilot and iCenter OS platform, we will address an example code for integration. Basically we're creating a step-by-step manual, which can be used by anyone who wants to get and integration from an ICT solution with ISA's iCenter.

All the following is intended to be used for a PHP integration.

The first and broader structure that need to be created is the *Unit:*

```
/*
 * createUnit creates a Unit data type in an Array
 *   $name = unit name
 *   $serial = unit serial code, should contain at least 1 character
  */
function createUnit($name, $serial)
{
    $newUnit = array(
        'id'=>null, // only used for return values
        'name'=> $name, // name of the unit
        'interval'=> 0, // not used, should be 0;
        'remoteid'=>0, // not used, should be 0
        'serial'=> $serial // should contain at least 1 character
    );
    return $newUnit;
}
```

Onwards there is the need for creating the next level entity, the *Device*:

```
<?php
/* This include contains functions to create arrays that can be sent to the ISA platform */

/*
 * createDevice creates a Device data type in an Array
 *    $name = device name
 *    $typeId = device type id, 1 = iClamp, 2 = netMeter Water, 3 = netMeter Gas
 *    $unitId = ISA id of the corresponding unit
 */
function createDevice($name, $typeId, $unitId)
{
    $newDevice = array(
        'id'=>null, // only used for return values
        'name'=> $name, // name of the unit
        'typeId'=> $typeId,// 1 = iClamp, 2 = netMeter Water, 3 = netMeter Gas
        'tariffId'=>null,//optional
        'stepId'=>null,// optional
        'unitId'=>$unitId,// mandatory
        'interval'=>0,// mandatory, but use 0
```

```
        'remoteId'=>0// mandatory, but use 0
    );
    return $newDevice;
}
```

Finally, with the *Unit* and the *Device* created, the only missing entity is the *Tag*:

```
/*
 * createTag creates a Tage data type in an Array
 * $name = tag name
 * $type = tag type; energy = 4, plog=5,water=6,gas=7,temperature=8,humidity=9,power=10
 * $deviceId = ISA id of the corresponding device
 */
function createTag($name, $type, $deviceId)
{
    $newTag = array(
        'id'=>null, // only used for return values
        'name'=> $name, // name of the tag
        'type'=> $type,//energy = 4, plog=5,water=6,gas=7,temperature=8,humidity=9,power=10
        'deviceId'=>$deviceId,
        'remoteId'=>0// mandatory, but use 0
    );
    return $newTag;
}
```

In order to successfully a Data Item you must invoke:

```
/* createDataItem
 * creates a data Item data type array.
 * tagId = ISA id of the corresponding tag
 * value = value of the measurement
 * date = timestamp of the measurement in format: 2004-12-01T00:00:00
 */
function createDataItem($tagId, $value, $date)
{
    $dataItem = array(
        'id'=>$tagId, // not sure though.
        'value'=>$value,
        'date'=>$date
    );
    return $dataItem;
}
?>
```

After the entities are created, the data upload and implementation is easily upload invoking the following functions.

```
<?php
/* This file contains the implementation of the ISA webservice, and offers simple functions
to add devices/units/tags and insert data
 It should be initated by:
 $ISAWebservice = new ISAImplementation($debugger, $ISA_wsdl, $ISA_username, $ISA_password);

 Then a device can be made by:
 //Insert Device
 $unit = createDevice("name", "1", "26");
 $ISAWebservice->insertDevice($unit);

 Bullk data can be inserted by:
 // Insert Bulk data
 $dataItems = array(createDataItem( 732, 10, '2004-12-01T00:00:00' ));
 $ISAWebservice->insertBulkData($dataItems);
*/

class ISAImplementation{
    private $debug;
    private $soap;
    private $token;

    /* constructor
     * $debug = true / false
     * $ISA_wsdl = address of ISA server
     * $ISA_username = username
     * $ISA_password = password
     */
    function __construct($debug, $ISA_wsdl, $ISA_username, $ISA_password)
    {
        $this->debug = $debug;
        $this->token = $this->getToken($ISA_username, $ISA_password);
        // connect to ISA webservice
        $this->soap = new SoapClient($ISA_wsdl, array('trace'=>$debug));

    }

    /* debug prints debug info if debug is enabled
     * $result = result of last soap call
     */
    function debug($result)
    {
        // check if debug is enabled
        if ($this->debug == false)
            return 0;

        // debug not enabled print debug information
        echo "**** Debug ****\n";
        echo $this->soap->__getLastRequestHeaders() . "\n"
            . $this->soap->__getLastRequest() . "\n";
        echo var_dump($result);
        echo "**** /Debug ***\n";
    }

    /* inserts a device to the ISA server
     * $device = device data type in array
```

```
     */
    function insertDevice($device)
    {
        // do the soap request
        $result = $this->soap->InsertDevice(array('token'=>$this->token, 'd'=>$device));
        $this->debug($result);
        if ($result->InsertDeviceResult != 'Success')
        {
            // insert failed
            echo "Insert device failed :" . $result->InsertDeviceResult . "\n";
            return -1;
        }

        // now get the id of the just created Device
        echo "Device id " . $result->id . "\n";
        return $result->id;
    }

    /* inserts a unit to the ISA server
     * $unit = unit data type in array
     */
    function insertUnit($unit)
    {
        // do the soap request
        $result = $this->soap->InsertUnit(array('token'=>$this->token, 'u'=>$unit));
        $this->debug($result);

        if ($result->InsertUnitResult != 'Success')
        {
            // insert failed
            echo "Insert Unit failed :" . $result->InsertUnitResult . "\n";
            return -1;
        }

        // now get the id of the just created unit
        echo "Unit id " . $result->id . "\n";
        return $result->id;
    }

    /* inserts a tag to the ISA server
     * $tag = tag data type in array
     */
    function insertTag($tag)
    {
        // do the soap request
        $result = $this->soap->InsertTag(array('token'=>$this->token, 't'=>$tag));
        $this->debug($result);
        if ($result->InsertTagResult != 'Success')
        {
            // insert failed
            echo "Insert Tag failed :" . $result->InsertTagResult . "\n";
            return -1;
        }
        // now get the id of the just created tag
        echo "Unit tag " . $result->id . "\n";
        return $result->id;
    }

    /* inserts bulk data t the ISA server
     * $dataItemArray = array of data item data types
     */
    function insertBulkData($dataItemArray)
    {
```

```
        // do the soap request
        $result = $this->soap->BulkInsertData(array('token'=>$this-
>token, 'data'=>$dataItemArray));
        $this->debug($result);
        if($result->BulkInsertMessageResult != 'Success')
        {
            // insert failed
            echo "Insert Bulk Data  failed :" . $result->BulkInsertMessageResult . "\n";
            return -1;
        }
        return 1;
    }

    /**
      Calculates the token that is used in communication with ISA server
    */
    function getToken($user, $pass) {
       $old_encoding = mb_internal_encoding();
       mb_internal_encoding('ASCII');
       $key  = "ISAiEnergy2009";
       $hmac = hash_hmac('sha512', $user.$pass, $key, true);
       $token = base64_encode($hmac);
       mb_internal_encoding($old_encoding);
       return $token;
    }

}
?>
```

The previously presented code provides the user with a complete set of tools to create, integrate, and upload the required data to iCenter OS.

Any more information may be obtained in the Data Provision Document, that is part of the annexes of the present document.

9.2 iCenter OS Data Provision

Consult supporting document - iCenter - Data Provision iEnergyOS.pdf (also available at SAVE ENERGY website)

9.3 Test Report

Consult supporting document - Test Report iCenterOS.pdf

9.4 iCenter OS Code Pack

You can download iCenter code package at SAVE ENERGY website:

http://www.ict4saveenergy.eu/