International Journal of Information Technology and Business Management

29th October 2012. Vol.6 No.1

ISSN 2304-0777 www.jitbm.com

DEVELOPING CLOUD ENERGY CONSUMPTION MONITORING SYSTEM FOR HOME ELECTRONIC APPLIANCES: INDONESIAN CASE

Teuku Aulia Geumpana / Joni Koentjoro / Nathalia Devina Widjaja
Bina Nusantara International University, Jakarta, Indonesia

Abstract --**This research is to measure electricity consumption for home electronic appliances using a cloud based application without modifying the default electricity meter from Indonesian National Energy Company. Different kinds of approaches like interviews, literature research and experiments were conducted to develop the prototype of the system. Arduino Board and Current Transformator were the hardware used to capture the currents transmitted from the electronic appliances. The application to store and interpret the data was built using PHP in the cloud platform. In the testing stage, the electronic appliances` energies were successfully captured and reported in the form of charts and graphs to the users. This application is hoped to give a smarter way for the users to decide how to control their energy consumption at home.**

***Keywords* -- Electricity consumption, cloud energy monitoring, energy preservation, greener IT, Arduino board, current transformator.**

1. INTRODUCTION

Based on 2050 ASEAN Electricity Demand Study, Indonesia will have 6.84% average growth for electricity demand up to 2050 [1]. Increase in energy demand is predicted to invite Indonesia along with other member countries of ASEAN to become a net importer of oil in the next 15 – 20 years [1,2]. With population of 240 millions, it will create huge demand in electricity consumption in Indonesia. The increase in electricity demand will lead to unnecessary cost spending due to lack of monitoring in electricity consumed. In Indonesia the needs to save electricity is considered crucial. Indonesia will always be power shortages as a continuous rising demand for electricity each year. Demand has surpassed the supply. In terms of electricity consumption, average watts / capita for Indonesia is 55.3 watts. Much lower when compared to America 1460 watts / capita and Germany 753 watts / capita[2,3]. And that's the fact from *Data National Socio Economic Survey from NSES/Susenas* that there are still 1/3 the people of Indonesia who have not had access to electricity[4,5]. The lack of electricity distribution has forced government of Indonesia to have a Rolling Blackouts[5].

Recently, Indonesia government has started the cost-saving programs by introducing a new package called 'STROOM'- as mentioned at the official website of PT PLN (National Energy Company), this is a prepaid electricity which acts just like the mobile phone recharge card so that people can have more control over their electricity spending[6]. However, this control does not give the user flexibility in knowing which devices that are consuming the most of the electricity. This may promote unnecessary electricity spending behavior and results to increase in cost and expenses. Therefore, ability to monitor the device electricity consumption may help user to avoid unnnecessary electricity usage. Parker in his studies revealed that if people can see their actual real-time electricity spending it will demonstrate a wide range of savings between 10% to 15% [3] as also discussed by Tribal Energies Research in electricity survey 2010 in Australia [7].

The aim of this research is to develop a user-friendly application to monitor electricity consumption for every home appliances and present it to an easy readable charts and graphs without having to modify the electricity meter from PLN. The application will be built on cloud-based technology to promote greener IT [8,9].

2. RESEARCH METHODOLOGY

An interview will be conducted with engineering department people from National Electric Company of Indonesia to gather requirements to see how the KWH (Kilowatt Hour) meter at home works. This is to understand the electricity computing pattern. Experiementation with Arduino boards will also be conducted to develop prototype system to measure energy generated by the electronic devices. To store and manipulate the data, an iterative system development implementation process will be carried through out this research. This electricity monitoring system will be deployed in a cloud based platform.

3. PROBLEM ANALYSIS

3.1 Existing Problem

The current KWH meter design from PLN cannot solve the main problem on how the residents can observe electricity consumption from each of their appliances at home. The current design only allows residents to get the bulk consumption value for the whole house. There are two kinds of solution that can be used to address this problem. The first solution is tampering the current electric meter by putting a measurement device and rearranges all of the cable before they go to different sockets across the house. The second one is by adding additional device in each of the appliances and connect them to a monitoring device which then we will able to monitor the individual electricity consumption.

3.2 Proposed System Specifications

In order to enable home residents to monitor the usage of electricity of their appliances, the system should have an additional device in each of the appliances and connects them to a monitoring device which can show its electricity consumption. The proposed system will have the following specifications:

- The system will be able to be applied on each of the electricity appliances at home according to users' preferences on which appliances they would like to monitor.
- The system will be able to send the reading over the cloud so users can monitor it wherever they are.

4. PROPOSED SYSTEM DESIGN

The proposed designs include system architecture, hardware selection consideration, connecting the hardware to the application which could monitor the energy consumption in a house, how to make sure each user had their own reading and finally hosting the application in cloud server.

4.1 General Design And Architecture of The System Prototype

Figure 1 System Prototype Architecture

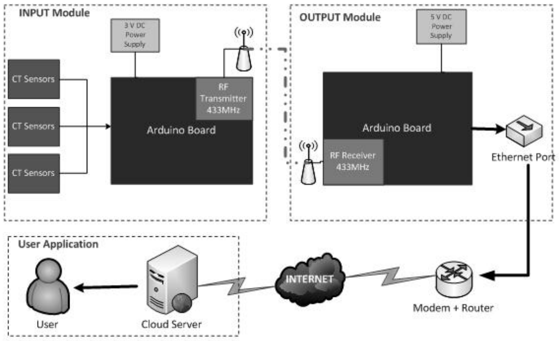


Figure 1 above showed the global architecture of the prototype. The architecture consisted of three important modules: input module, output module and user application.

The input module was responsible for sensing the environment which was the current usage for each appliances. The reading will be forwarded to the output module via wireless connection. The main output module consisted of microcontroller to process the reading, wireless receiver and Ethernet port. The wireless receiver would receive the data, convert it into JSON format and then transmitting the data using Ethernet cable to the application in the webserver. The webserver was hosted in the cloud hosting provider (in this case Windows Azure) to promote scalability and efficiency[10]. The application webserver than process the JSON data and the users would be able to determine each of the reading into the corresponding appliances and presented to the user in terms of charts. After that the data was recorded to MySQL database for historical analysis.

The application was built using PHP language and the data was saved in MySQL database. The application was expected to be deployed in a cloud platform. This is because cloud is such an ubiquotous technology that serve more flexibilities and efficiencies to the users. In addition, cloud has widely promoted easy understanding to people. Mike Vizard in the survey conducted by CompTIA (2011) mentioned that there is significant increase in understanding of cloud computing [11]. System charting will be using FLOT. As mentioned by R. Cheung at WebAppers, FLOT is a pure Javascript plotting library for jQuery [12]. The application had a multi-user capability, so it would require users to registers themselves before they used the application. The Energy Monitoring application has adapted Model View Controller (MVC) architecture. According to Fowler in his article about GUI Architecture, MVC provided more flexibility when adding new functionality as it separated functions to different modules; model (business logic), view (user interface) and controller (user input) [13].

For security features, access privileges were required to execute actions in controller. There are three types of privileges:

1. **Public** action – a non authenticated user can call this action.
2. **Read** access action – a user with a read authkey can call this action.
3. **Write** access action – only users with full authentication can call this action.

Full authentication requires either user login with a username and password, or a write access authkey. To be able to use the application users needed to have at least read privilege, it would allow users to see the graphs and charts, while write privilege was used to post data from the output board to the webserver or updating the feeds. The 'write' authentication key was coded in the output board. This was to ensure that only specific board from specific user could access the application. The AUTH key is generated using uniqid() function from PHP and then it is encoded using MD5 security. The 'write' authentication key could also be used to debug the application by copy paste the URL provided into the browser. That way the application could register the input and treated it as if it came from the output board.

Meanwhile, sessions are handled with php sessions. On user login or register the session id is regenerated with: session_generate_id(); URL rewriting has been implemented for added security. This was to prevent someone getting access to the file directly.

For example:
http://localhost/site/energon/panel/view instead of
http://localhost/site/energon/panel/view_panel.php
The application was equipped with customizable panel to show live/current reading and historical graph to see the appliance's usage in a period of time. The panel was customizable so user could choose which appliances that they wanted to monitor. The graph had several different time frames from Daily, Weekly, Monthly or Yearly.

4.2 Input Module

The input module was an Arduino board which consisted of three main parts, i.e. CT sensors, ATMEGA 328 microcontroller, and RF

Transmitter. CT sensors were used to read the electricity usage from appliances in terms of Current (Ampere); the microcontroller was responsible for all the data processing and calculation including converting the current into power (Watt) while the RF transmitter was used to forward the data into the output module. Several libraries from jeelab were used to program the input board, the manufacturer of many of the arduino plugin for example the Ethernet port, wireless transmitter, etc. Some of the libraries used are Arduino Library, RF12.h, Ports.h.

The first thing to do was to prepare the data structure to hold the data. There were two data structures needed to read and forward the energy reading. The first data structure was used to prepare how many sensors were going to be used in the board. For this research, two sensors input were used. In the view box 1 below is the proposed data structure:

View Box 1 Sensor Data Structure

```
class Sensor //create class of eload
              //variables  to allow two
              //sensors input monitoring
{
 public:
 double eload(int,double,int,int,int,int);
 private:
        //sample variables
        int lastSampeI, sampleI;
        // filter variables in calculation
        double lastFilteredI, filteredI;
        // power calculation variables
        double sqI, sumI;
};
Sensor ch1, ch2; //create three instances of
                 //Sensor to allow  two
                 //sensors input monitoring
```

The second data structure was used to hold the energy reading for each of the sensor input and the board power to indicate whether the battery was still in a good condition or needed to be replaced. The following view box 2 is the data structure to be sent over the wireless transmitter:

View Box 2 Energy load data structure

```
typedef struct {

   int ct1;  //current transformer 1
   int ct2;  //current transformer 2
   int supplyV; //voltage of emonTx battery

} ElectricLoad;
ElectricLoad eloadIn;
```

After the data structure is ready, the next step is to read the analogue pin where the current would be coming into the board. To read the value, the funtion used is: int analogRead(pin) where pin was the number of the analog input pin to read from and it would return the reading in integer format. If the analog input pin was not connected to anything the value returned by analogRead() will fluctuate based on a number of factors (e.g. the values of the other analog inputs, how close any other electrical interference to the board was, etc.), in which we called this as noise. The next step is to calculate the power with the following formula:

Relationship between real, reactive and apparent power (AP):

Real Power = AP x cos φ
Reactive Power = AP x sin φ
cos φ is the power factor

The final step is to send the power value to the output module using wireless transmitter. The wireless transmitter need to be waken up from the sleep state and then it could start receiving the package. Below in the view View Box 3 is the snippet code for RF Transmitter:

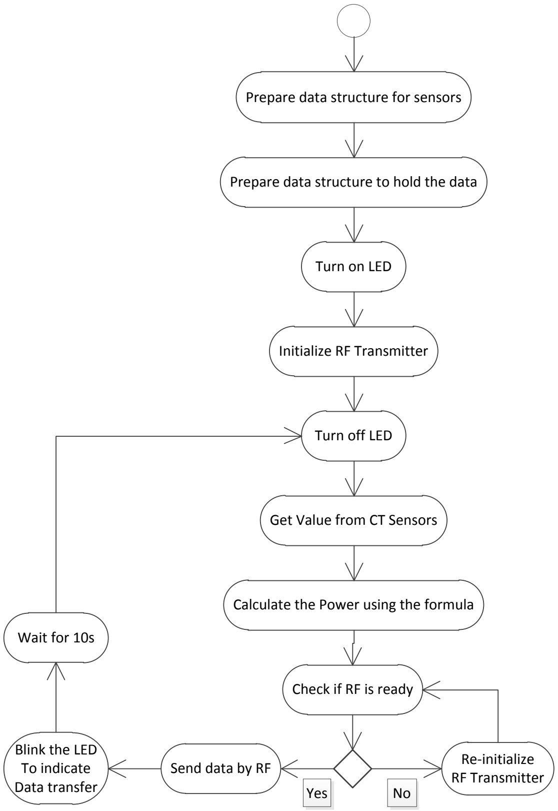
View Box 3 Transmist Package using RF Transmitter

```
static void rfSend() {

    //wake up RF Module
        rf12_sleep(RF12_WAKEUP);
        while (!rf12_canSend()) {
         rf12_recvDone();
        }
rf12_sendStart(0, &eloadIn, sizeof eloadIn);
//wait for RF to finish sending while
//in idle mode
rf12_sendWait(2);
// put RF Module to sleep
rf12_sleep(RF12_SLEEP);
```

From the snippet above, the RF12_SLEEP function, was to ensure that the wireless module did not draw energy continuously from the battery power, that way it could save more energy and the battery could last longer to power up the input module. The activity diagram for the whole input module process would be shown in the Figure 2 below.

Figure 2 Activity Diagram for Input Module

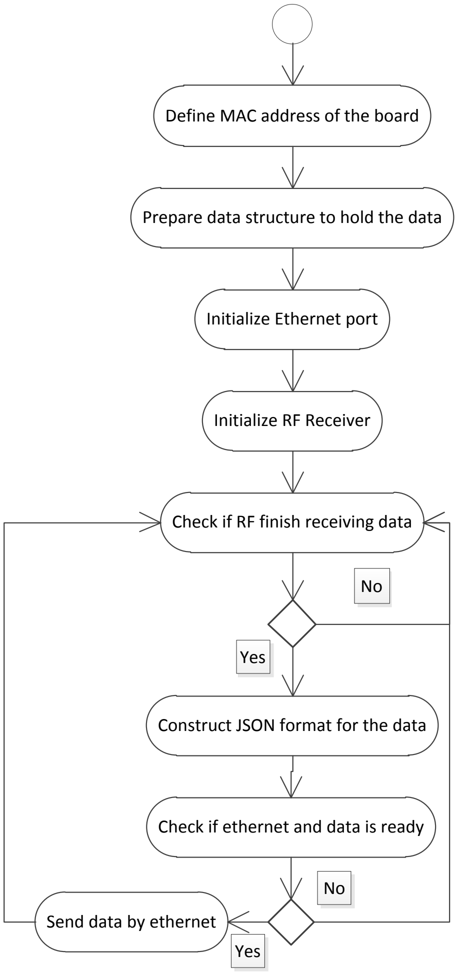

Prepare data structure for sensors
Prepare data structure to hold the data
Turn on LED
Initialize RF Transmitter
Turn off LED
Get Value from CT Sensors
Calculate the Power using the formula
Wait for 10s
Check if RF is ready
Blink the LED
To indicate
Data transfer
Send data by RF
Re-initialize
RF Transmitter
Yes
No

4.3 Output Module

The output module was also using an Arduino board which consisted of three main parts, i.e. *ATMEGA 328 microcontroller*, *ethernet port* and *RF Transmitter*. The microcontroller was responsible for all the data processing and communication redirection while the RF transmitter was used to receive the data from the input module. The ethernet port was used to send the data to the webserver to be read by the application so user could see it in graphical form.

To program the output board, again several libraries from jeelab were used, the manufacturer of many of the arduino plugin for example the Ethernet port, wireless transmitter, etc. Some of the libraries used are RF12.h, Ports.h, Ethershield.h, and NanodeMAC. The activity diagram for the output board will be shown in the Figure 3 below.

Figure 3 Activity Diagram for Output Board


Define MAC address of the board
Prepare data structure to hold the data
Initialize Ethernet port
Initialize RF Receiver
Check if RF finish receiving data
No
Yes
Construct JSON format for the data
Check if ethernet and data is ready
No
Send data by ethernet
Yes

5. DEPLOYING THE APPLICATION

To deploy and test the application, two different kind of servers were needed. The first one was localhost/offline server, this was the server that resided on the local computer and used merely for testing the application in details. It was done to test the individual module of the system prototype (input, output and the application itself). The second one was live/online server in which the application was tested in a real server to make sure the application run smoothly before it was launched to the market. The live server development environment was using Linux operating system with Apache Server and mySQL database.

6. IMPLEMENTATION AND TESTING

The test conducted would show the expected result and the actual result of the application. The test plan was done to make sure that the application could function according to the specifications. The test also involved all possible input data and should return the correct error handling.

7. SYSTEM SPECIFICATION

7.1 Software

Please refer to table 1 at the end of this paper for software needed for the development of the application and system prototype.

7.1.1 Hardware

Please refer to table 2 at the end of this paper for hardware used when developing and testing the system prototype.

7.1.2 Server

The server specification that was used when testing the system in online server.

Table 1 Server Specification

Category	**Type**
OS	Centos 6.2
Processor	Intel i7-2600 3.4GHz 8MB L2
RAM	16GB DDR3
HDD	2 x 3TB SATA 6GB/s 7200rpm

7.2 Testing

The module testing was aimed to test each of the modules independently and the test was conducted in a localhost to get more accurate result limiting the system from external environment. The module testing would include testing input, output and the application itself.

Table 2 Testing Result

Module Testing	Application Test	SUCCESS
	Input Module Test	SUCCESS
	Output Module Test	SUCCESS
Integration Testing	Reliability Test	SUCCESS
	Compatibility Test	SUCCESS
	Security Test	SUCCESS

8. DISCUSSION

8.1 Power Consumption

There were couples findings during the test on the system prototype. The first finding was that appliances had different power consumption according to the corresponding load at the particular time. For example computers that were idle drew less energy compared to when the computers were used to process something. Another observation were comparing when the appliances were turned off and in standby mode just like in Television. The difference between the two was not quite significant although in standby mode the appliances were still drawing power in small amount.

Another observation was to see the difference in power consumption from Air Conditioner with different setting, for example AC with the temperature below 20 degrees Celsius drew much more power compared with the one that were above 25 degrees Celsius. The wireless capability between the input and output board was quite reliable. It could sustain a distance up until 20 meters as long both the devices were in the same floor and it were still working when the input board were put in 1 floor away. The charting features of the application were very helpful to help users saw the power consumption in particular period of time. This could help user to raise their awareness in reducing their electricity consumption. This can also provide snapshot to help users in comparing their electricity usage each month or each year.

8.2 Development and Testing Difficulties

In developing the prototype and doing testing, some difficulties that happened were hardware calibration, compatibility issues and cloud uploading issue. The first difficulty was calibrating the input board in order to show the correct power usage. Because the inputs were analog, it produced ‘noise’ in the input sensor, so when the inputs were not connected to the appliances it already produced some values i.e. the ‘noise’. The next difficulty was the browser compatibility with the application. As the application is cloud based it relies on the browser. The application has been tested with various kinds of browsers including Firefox, Opera, Chrome, Safari and Internet Explorer. However in Internet Explorer the graph could not be shown but the other functionalities were working properly. The problem is related to rendering engines where the library was having problem with Internet Explorer **[**14**].**

To deploy the application using windows azure platform, many problems occurred that prevented the deployment.

1. In order to use Windows Azure, one must register his/her self and the country of location did not include Indonesia.
2. Windows Azure had recently supported PHP in their supporting application, so the library was still new. Even the Windows Azure SDK for PHP had just released on 3 December 2011.
3. MySQL was not yet natively supported by Windows Azure, even to use MySQL, $0.12/hour for one instance of MySQL need to be paid.
4. The URL rewriting needed to be inputted manually and we could not just copy the .htaccess file just like in the Linux based server.

Cloud hosting is basically the same with the traditional web hosting. The only difference is that cloud hosting offers flexibility and scalability. Flexibility means that the server can be resized or move anytime in real-time, while scalability offers server customization according to the users’ needs.

According to SingleHop (one of the cloud providers in US), cloud hosting had the same server mechanism in application development with the traditional web hosting, that means everything that can be deployed in the traditional web hosting, can be deployed on cloud hosting **[**15**]** similar statement also mentioned elsewhere [16].

Table 3 Software Specification

Operating Systems	**Local server Environment**	**Administration Tools/Language**	**IDE**	**Cloud Platform**	**Browsers**
Microsoft Windows 7 (32 bit and 64bit)	XAMPP 1.7.7	phpMyAdmin 3.4.5	Netbeans 7.0.1	Windows Azure SDK 1.4	Mozilla Firefox 9.0.1 (Windows and Lion)
OS X Lion	MAMP Pro 2.0	PHP 5.3.8	Microsoft Visual	Windows Azure MySQL PHP	Google Chrome

			Studio 2010	Solution Accelarator	16.0.912.75 (Windows and Lion)
				Windows Azure SDK for PHP v4.1.0Arduino 0.23	Internet Explorer 9 for Windows
					Safari 5.1.2 for MAC
					Opera 11.60 for Windows

Table 4 Hardware Specification

Category	**MAC**	**Laptop Windows**	**PC**
OS	OSX Lion/Snow Leopard	Windows 7 - 64 Bit	Windows 7 - 32bit
Processor	Intel i5 - 1.6Ghz/2.7GHz	Intel i5 - 2.4Ghz	Intel CoreDuo - 2.4Ghz
RAM	4GB DDR3	4GB DDR3	2GB DDR2
HDD	128GB SSD	4TB HDD	500 GB HDD

9. CONCLUSION AND RECOMMENDATION

9.1 Conclusion

For the hardware part, Arduino board was used to connect with the external environment that read the electricity usage and then process the data and post it to the webserver for further processing into meaningful graphs and charts. The application itself is cloud-based, built using PHP language and used MySQL database for datastorage in the webserver.

The first limitation was the wireless range in which it can accommodate for a maximum distance of one floor. The second limitation was the ‘noise’ in the input sensors. Because the input board used analog read, noise cannot be omitted, and the only way to solve the noise problem was by having the right calibration amount. The application has served its purpose as an energy monitoring system in which it is expected to be able to help people control their electricity usage.

9.2 Recommendation and Future Work

For future work, it is good to combine the system with remote control system, so users are not only capable of seeing their appliances but they can also turn off remotely if they accidentally left the appliances on while leaving the house. Therefore, it is good to develop the mobile version of the application and explore another options of input board so that it will result in more precise reading.

10. REFERENCES

[1] N. Utama, K. Ishihara, Q. Zhang , "Green Energy and Technology, 2011," *Zero-Carbon Energy Kyoto 2010*, Vol. Part I, : Springer, p. 32-39, 2011.

[2] M. Wijaya, T. Tezuka , "Electricity Saving Potential in Indonesian Households: A Techno-Socio-Economic Analysis ," *4th International Conference on Sustainable Energy and Environment (SEE 2011)*, 27-29 February 2012.[Online]. : http://bit.ly/VfazMn. [Accessed November 8,

2011]

[3] D. Parker, D. Hoak, A. Meier, "How Much Energy Are We Using? Potential for Energy Demand Feedback Devices," *Proceedings of the ACEEE Summer Study on Energy Efficiency in Buildings*, August 13, 2006 .[Online]. :http://www.aceee.org/proceedings-paper/ss06/panel01/paper19. [Accessed November 9, 2011]

[4] Alpen steel , "Pemadaman listrik bergilir di Indonesia," *Unpublished*, [Online]. :http://www.alpensteel.com/article/106-225-pemadaman-listrik/2192--pemadaman-listrik-bergilir-di-indonesia.html. [Accessed November 9, 2011]

[5] Edwin, "Jero: Indonesia akan Selalu Kekurangan Listrik ," *ROL: Republika Online*,16 July 2012.[Online]. :http://www.republika.co.id/berita/nasional/umum/12/07/16/m78il3-jero-indonesia-akan-selalu-kekurangan-listrik. [Accessed November 8, 2011]

[6] Prepaid Electricity (2011) PT PLN (Persero). [Online]. http://www.pln.co.id/eng/?p=501

[7] Tribal Energies. Australian's View on Electricity Price Increases. Tribal Energies Research. October, 2010.[Online]. http://tribalenergies.com.au/

[8] N. Widjaja, M. Mariani, K. Imam, " IT Professionals Awareness: "Green IT International Comparison Study *", Proceedings of the Annual International Conference on Green Information Technology, Singapore,* 25-26 October 2010.

[9] S. Pino, P. Bhatia, " Working 9 to 5 on climate change: An office Guide ," *World Resources Institute*, no. 1-56973-530-1, December 2002.[Online]. :from http://pdf.wri.org/wri_co2guide.pdf. [Accessed November 8, 2011]

[10] G. Reese, *Scaling a Cloud Infrastructure" in Cloud Application Architecture; Building Applications and Infrastructure in the Cloud*, Edition of book, California: O'Reilly Media, Inc, 2009.

[11] M. Vizard. Trends in Cloud Computing. CTO Edge. August 31, 2011. [Online]. http://www.ctoedge.com/content/trends-cloud-computing?slide=2

[12] R. Cheung. Flot Pure Javascript Plotting Library for jQuery in WebAppers. December 2006. [Online].http://www.webappers.com/2007/12/06/flot-pure-javascript-plotting-library-for-jquery/

[13] M. Fowler, GUI Architecture, 18 July 2006, [Online] . http://www.martinfowler.com/eaaDev/uiArchs.html#ModelViewController

[14] L. Palacio. For an Energy-Efficient Millennium: SAVE2000. The SAVE Programme. European Commission Audiovisual Library. 1999. [Online]. http://www.ademe.fr/partenaires/odyssee/pdf/save2000.pdf

[15] SingleHop - Private Cloud. 2011. [Online]. http://www.singlehop.com/cloud-hosting/private-cloud.php#

[16] Infinys System Indonesia. 2011. [Online]. http://www.infinyscloud.com/id