TAMPEREEN TEKNILLINEN YLIOPISTO
TAMPERE UNIVERSITY OF TECHNOLOGY

MOHAMMAD MAHMUD HOSSAIN

PORTLET-BASED PRESENTATION OF ENERGY KPIS IN SOA-ENABLED MANUFACTURING FACILITIES TARGETING HOLISTIC ENERGY MANAGEMENT

Master of Science Thesis

Examiner:
Professor Dr. Jose Martinez Lastra
Examiner and topic approved by the Council meeting of the Faculty of Engineering Sciences on 15th January 2014.

ABSTRACT

TAMPERE UNIVERSITY OF TECHNOLOGY
Master's Degree Programme in Machine Automation
MOHAMMAD MAHMUD, HOSSAIN: Portlet-based presentation of energy KPIs in SOA-enabled manufacturing facilities targeting holistic energy management
Master of Science Thesis, 65 pages, 30 Appendix pages
April 2014
Major: Factory Automation
Examiner: Professor Dr. Jose L. Martinez Lastra
Supervisor: Anna Florea
Keywords: service oriented architecture, energy management system, energy efficiency, key performance indicators, web services, web portal, portlet.

One third of global energy consumption is directly attributed to manufacturing industry. The demand of efficient energy usage in manufacturing is prompted by scarcity of traditional energy resources and rising energy price. Structures of current manufacturing facilities are complex and distributed with large number of assorted devices involve in manufacturing, utilities, lighting and HVAC systems within organization. Adoption of Information and Communication Technologies (ICT) allow near real time information acquisition from these assorted devices. In this thesis work, an energy management portal is designed and developed to present online and historical energy information to hierarchical user groups in manufacturing enterprise.

The thesis consists of background study, design and implementation. The current standardizations on Energy Management System (EMS) are studied to follow a systematic approach towards design and implementations. Role of KPI in manufacturing is studied from EMS point of view through previous research and implementations. Relevant technologies and architectural concepts are explored, analyzed and compared including Service Oriented Architecture (SOA) and web portal infrastructures.

A generic framework for presentation layer of EMS is designed targeting SOA-enabled manufacturing facilities which utilizes existing modular systems i.e. MES, ERP and so on. Result of implementation shows that, portlet based web portal draws SOA concept to User Interface (UI) of EMS where portlets act as pluggable UI component of flexible and configurable dashboards of hierarchical user groups. Presentation of online and historical energy KPIs from heterogeneous sources is achieved in different portlets using DPWS and WS. Adoption of implemented energy data representation techniques and role-based dashboards increase energy awareness among user groups in factory and support decision-making for achieving energy efficiency.

PREFACE

First of all I would like to express my gratitude to Professor José Luis Martínez Lastra for the opportunity of doing this thesis work under his research group (FAST). FAST has given me the confident of self-learning ability and team work as well. I want to thank all members of FAST for providing the great working environment. A special appreciation goes to my supervisor Anna Florea. I cannot imagine of completing this work without her guidance and support. I would also like to thank Dr. Andrei Lobov and Dr. Corina Postelnicu for their excellent courses that have been basis of this thesis work.

I want to thank all my fellows Anton, Juha, Sohail, Ville, Ahmed, Luis and Peyman for their supports during and after work. I would also like to thank all my friends I met here in TUT. A very special thank to Khuki, my closest friend and more than that. Thank you for your accompany and the joys you brought to me. Thanks to my childhood friends Shad and Bakar for consistently giving me inspiration from distance.

I want to thank my mother Nazma Begum for everything. I could never make this long journey happen without her love and courage. Thanks to my all of my family members. At last but not the least my all respect and gratitude to my elder brother Mynul Hossain Faruk. The person who replaced my fathers' position and provided me everything I needed to come in this position. He has been my ideal, inspiration and everything. Thanks to the creator for giving me such Brother.

Tampere, April 22nd, 2014
Mohammad Mahmud Hossain

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACRONYMS

API	Application Programming Interface
CEP	Complex Event Processing
DAO	Data Access Object
DPWS	Device Profile Web Service
DSL	Domain Specific Language
EDA	Event Driven Architecture
EIP	Enterprise Integration Patterns
EMS	Energy Management System
EMP	Energy Management Portal
ERP	Enterprise Resource Planning
ESB	Enterprise Service Bus
EUP	Energy Use Parameters
HTML	Hyper Text Markup Language
HTTP	Hyper Text Transfer Protocol
HVAC	Heating, Ventilation, and Air Conditioning
IEA	International Energy Agency
ISO	International Organization for Standardization
Java EE	Java Enterprise Edition
JDBC	Java Database Connectivity
JSP	Java Server Pages
JSR	Java Specification Requests
JSON	JavaScript Object Notation
KPI	Key Performance Indicator
KRI	Key Result Indicator
MVC	Model View Controller
MES	Manufacturing Execution Systems
OEE	Overall Equipment Effectiveness
REST	Representational State Transfer
RTU	Remote Terminal Unit
SOA	Service Oriented Architecture
SOAP	Simple Object Access Protocol
UML	Unified Modelling Language
UDDI	Universal Description, Discovery and Integration
URL	Uniform Resource Locator
UI	User Interface
WS	Web Service
WSDL	Web Service Definition Language
XML	Extensible Markup Language
XSD	XML Schema Definition

1. INTRODUCTION

The global energy consumption is growing rapidly, where there has not been any significant innovation or discovery in conventional energy resources for last few decades. According to the World Energy Out-look 2012 published by the International Energy Agency (IEA), the price of electricity will increase by 15% between 2011 and 2035 [1]. Besides, "Energy-smart Europe on the horizon" campaign of European Union sets three goals for 2020; cutting greenhouse emissions by 20%, drawing 20% of energy from green sources and reducing energy use by 20%, which requires energy efficient industries in Europe [2]. Industrial sector in Europe consumes 25.3% of total energy; similarly it is 32% in US [3], [4]. Due to the requirements of competitive global markets, effective factory Energy Management System (EMS) is a vital topic in recent research.

EMS plays an important role to improve energy efficiency in manufacturing facilities. This thesis work focuses on presentation of energy information through EMS. The chapter begins with describing some previous research and implementations as background of implemented energy management solution. A quandary of current energy management systems is defined to improve. The work is illustrated by setting objectives for designing intended energy management system. Then, methodology approach is stated to attain the objectives. Finally, the outline of the thesis is given at the end of this chapter.

1.1 Background

EMS is a coordinated platform of four energy applications; procurement, conversion, distribution and utilization in order to cover requirements [5]. The International Organization for Standardization (ISO) introduced a structured background for EMS in ISO 50001: 2011. It is describing standardized approaches including policy development, measurement, improvement, and system optimization. This standard is based on Plan - Do - Check - Act (PDCA) to improve organization's energy efficiency, leading to energy savings and cost reduction [5].

Although current energy management systems are acclimatizing with ISO 50001 standards, they are available as entire package of solution with devices and applications. It causes incompatibility in integration with MES and ERP systems coexist in manufacturing facilities. In contrast, open standards and technologies are recent trend in manufacturing industry. Service Oriented Architecture enables the integration of device level services with enterprise systems. In order to allow flexibility, reusability and reconfigurability in manufacturing systems, a framework was developed in SOCRADES project by implementing SOA paradigm at device and application level [6]. SOA enabled EMS allows real-time decision making based on on-line and offline information coupled with business processes [7].

EMS needs to be cross-domain solution combining energy information of facilities, manufacturing and other utilities to manage entire energy activities within the enterprise [10]. Information of these heterogeneous sources can be accumulated as form of energy KPIs.

Energy KPIs helps personnel to improve energy awareness by making organizational changes in management, operations and development [11]. However, an outline is needed to integrate cross layer data flows and present accumulated KPIs to right personals within organization.

Web portal architecture has been a popular choice for cross layer data integration and multilevel access control in complex ERP solutions [12]. This work facilitates a similar approach for EMS. Architecture of portlet based web portal allows EMS to integrate services from different layers to presentation layer enabling proper holistic visibility and hierarchical user role management, where portlets are used as pluggable user interface components.

1.2 Problem definition

Most of modern manufacturing facilities are distributed and assorted in structure. Thus Energy performance within organizations not only influenced by equipment and devices dedicated to manufacturing processes but also buildings, lighting system, HVAC system of the facility and personnel influence patterns as well. An open standard solution is still missing in research and implementations to present overall energy information within manufacturing enterprise in flexible and configurable platform.

1.2.1 Justification for the Work

A framework of holistic EMS with multilevel presentation dashboards is required to ensure energy aware product design, customization, process algorithm, maintenance, monitoring and management.

However, success of holistic EMS is constrained by complexity in manufacturing systems. Major challenge of holistic EMS is cross-layer data integration. A set of energy KPIs also needs to be identified in vertical and horizontal level of manufacturing sites to improve energy awareness. In addition, appearance of energy information should be in a meaningful way that the presentation turns into solution of the energy optimization [15]. Thus, it supports each user group to set target, analyze trend and evaluate decisions.

The implementation of SOA paradigm allows aggregation of energy information from device level to higher level applications. Adaptation of SOA with CEP implementation helps extracting formulated energy KPIs from events and processes. As architectures of SOA and EDA with CEP implementation are studied in MES and ERP systems

[7]; this work focuses on data aggregation in presentation layers from EMS point of view.

Furthermore, thesis also facilitated portlet based web portal architecture that allows EMS to integrate services from different layers to presentation layer, which enables proper holistic visibility and hierarchical user role management.

1.2.2 Problem Statement

"How to design and organize hierarchical dashboards in manufacturing facility to present energy KPIs collected from heterogeneous sources in order to support holistic energy management?"

1.3 Work Description

1.3.1 Objectives

The main purpose of the thesis is to design supporting tool for making energy efficient decisions within manufacturing facilities by presenting energy information among different entities through hierarchical dashboards of Energy Management Portal. In order to achieve the goal, the whole work can be derived into subsequent objectives:

- Develop a method to integrate heterogeneous sources to get energy information within manufacturing enterprise.
- Design communication rules and service consumption methods between EMS and other existing modular systems i.e. MES and ERP.
- Identify a set of energy KPIs for holistic EMS to present offline energy information.
- Design and implementation of Energy Management Portal to fit in EMS scope by presenting online and offline energy information.
- Design and implementation of platform independent responsive dashboards targeting different user groups in organization.

1.3.2 Methodology

- **Research on Energy Management System**
    - A research is conducted to comprehend instruction and approaches standardized by International Organization for Standardization in ISO 50001.
    - A study on KPIs and its role is done to understand role of energy KPIs in holistic energy management systems. The study is conducted by exploring existing standards on KPIs in manufacturing facilities and from previous researches and implementations mentioned in scientific articles and journals.
    - A research is conducted to realize utilization of SOA paradigm in Energy Management System for manufacturing facilities. It is done by studying previous researches and implementation of SOA paradigm in Factory Automation Systems

and Building Automation Systems. The study is extended by reviewing existing commercial EMS solutions.

- An exploration on existing portlet containers is carried out to compare their capabilities, architecture and feasibilities for implementation of targeted EMS. Development platform, flexibility and configurability are considered as key features during exploration process.

▪ **Design and Implementation**

An EMS needs to be developed with flexible customization capabilities in interface level and configurable hierarchical user management system. The hierarchical dashboards of EMS present energy information from heterogeneous sources within the context of its user groups. The following steps are done to realize the EMS application.

- **Implementation of EMS**
  In order to enable flexible dashboard management and user management in UI level, tools and techniques are selected and illustrated to implement EMS.
- **Development of portlet applications**
  Portlets are pluggable UI component in portlet based web portals. A set of portlet applications is developed and deployed in EMS application to present historical or online energy KPIs to different user groups. Each portlet is designed and developed considering its user group and role in EMS.
- **Design of Dashboards**
  Dashboards for different user groups are designed utilizing developed portlet applications. Design of dashboards is enabled in UI to serve the purpose flexibility and re-configurability.

1.3.3 Assumption and limitation

The work is performed under the following assumptions and limitations:

▪ Performance of device level services and readings are assumed to be correct and reliable.
▪ Other existing components i.e. MES and ERP are capable of publishing information using web services.
▪ Some information were aggregated for shorter period, but simulated for longer period in order to illustrate broader depiction of production line and role of corresponding user groups.

1.4 Thesis Outline

The rest of thesis is organized as follows: a literature review is presented in chapter 2 arranging theoretical background, concepts, technologies and tools used in this thesis. Chapter 3 describes methodology approach for the development of Energy Management

Portal. The implementation of EMS, presentation techniques of energy information in hierarchical dashboards is described in chapter 4. Chapter 5 demonstrates result of implementation for used test bed. Finally chapter 5 concludes on presented work with future scope.

2. LITERATURE AND TECHNOLOGY REVIEW

This chapter focuses on literatures and technologies behind the methodological approaches and implementation of EMS in this thesis work. Firstly, it illustrates steps of current standardization on EMS. Then a background study on KPIs in manufacturing industry is presented with state of art. Architecture of SOA paradigm is studied with state of art as an approach to accumulate energy KPIs from various sources of manufacturing systems. Finally structure of portlet based web portal is discussed in details to draw SOA in presentation level of EMS.

2.1 Standardization on Energy Management System

From numerous definitions it can be said that Energy Management System (EMS) is a framework of systematic approach to improve energy performance, energy efficiency, energy use and consumption in organizations [5]. Several researches were initiated to standardize the structure of EMS. An industrial standardized approach was proposed in [16]based on existing standards of The International Organization for Standardization (ISO). The suggested framework includes measurement, documentation and ongoing instruction to optimize systems.

The EN 16001:2009 Energy Management Systems standard was released by European Committee of Standardization to support organizations in management and setup of processes to improve energy efficiency [17]. EN 16001:2009 is structured on existing management standards such as ISO 9001 and ISO 14001, where ISO 9000 addresses various aspects of quality management systems and ISO 14000 deals with the fundamentals of environmental management for manufacturing and service industries [18], [19].

ISO issued a new standard in June 2011 to standardize the approaches for investigating energy usage within organizations [5]. This standard ISO 50001 officially replaced the old EN 16001 in Europe since April 2012.

ISO 50001 is based on Deming Cycle also known as (Plan- Do – Check - Act) PDCA cycle. Edward Deming promoted the cycle as an effective continuous improvement tool for processes in organization Figure 1.

From EMS point of view, PDCA cycle can be described as follows:

- **Plan:** Organization continuously records and analyzes information about energy consumption. This information establishes an energy baseline with energy Key Performance Indicators (KPI). Energy KPIs helps organization to assess energy performance within its scope. Eventually, organization sets objectives, targets and action plans to improve energy performance according to its policy. Energy planning also considers result of previous targets defining achievement for specific time frame [5]: 4.4.

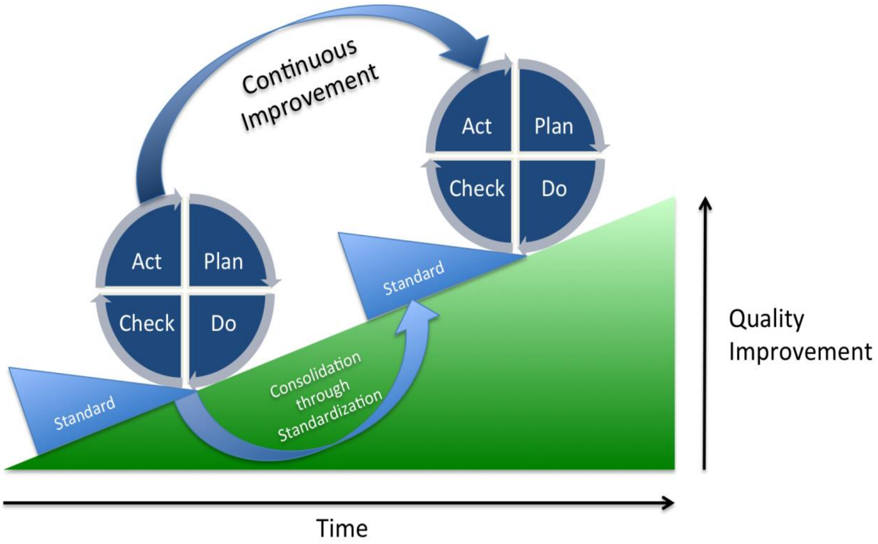

Figure 1: PDCA Cycle according to Edward Deming [source 20]

- **Do:** Organization implements the action plans in this phase. All entities working in the organization must be aware of their energy uses, role and significance in improving the energy performance. For example in manufacturing facilities, this includes machines working in processes, operator monitoring machines in shop floor, production manager scheduling and designing production plan, facilities managers, enterprise managers and so on. It ensures that operations associated with significant energy use are carried out effectively in each level of organization [5]: 4.5.
- **Check:** This phase reviews other three phases belonging to PDCA cycle. A better illustration is depicted in Figure 2 adapted from official site of ISO 50001. Figure 2 determines that energy is monitored, measured and analyzed to propose corrective and preventive actions. Eventually this proposal goes through organizations internal audit to check its effectiveness [5]: 4.6.
- **Act:** In this phase EMS deals with management review. It settles on actions to continuously improve energy efficiency and performance of EMS. Important aspect of this stage is; it evaluates targets and results to upgrade policy and set future objectives [5]: 4.7.

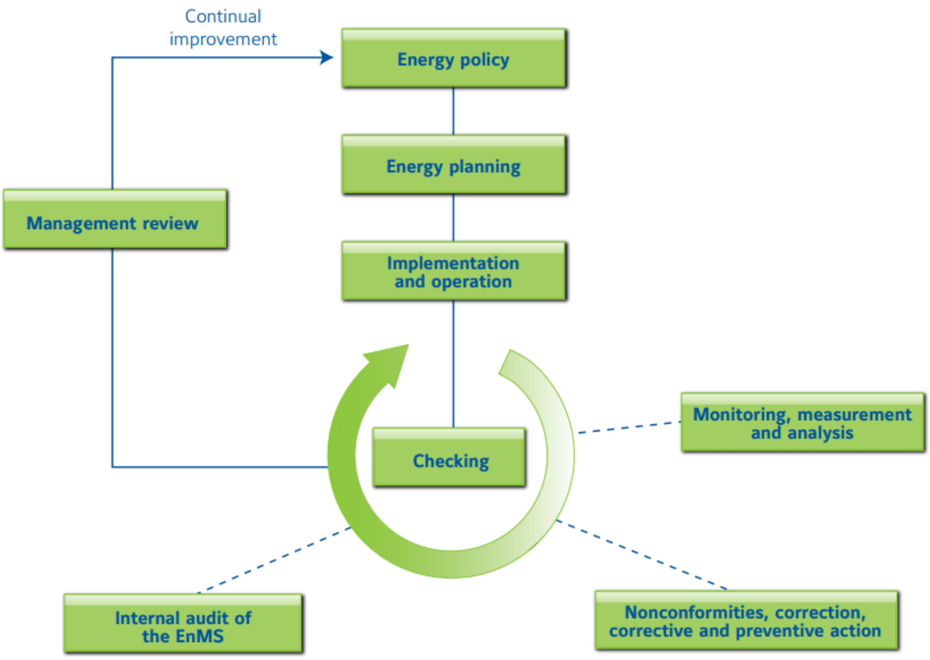

Figure 2: Energy Management System Model [5]

According to ISO 50001, successful implementation of EMS depends on participation and commitment from all levels of organization.

2.2 Key Performance Indicator (KPI)

Based on given definitions in [21]-[25], KPI is a metric or measure to quantify and evaluate performance in relation to targets and objectives of organization. Organizations utilize KPIs to set objectives, track progress, monitor trend and support interim decisions. A distinguishing explanation between KPI and other performance indicators like Key Result Indicator (KRI) and Performance Indicator (PI) is mentioned in [21]. It defined KPI as most critical indicator for current and future achievement of organizational performances. While, KRI explains from results perspective covering longer period and PI specifies what to do to improve performances. Figure 3 is adopted from [21] to portray performance measures clearly.

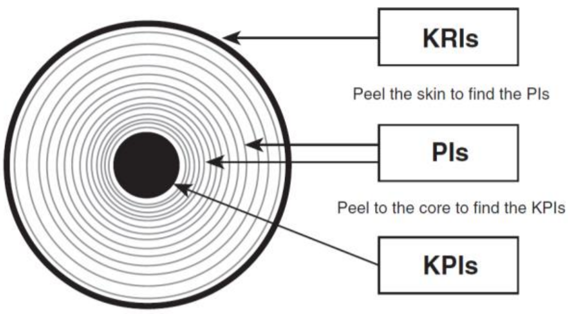

Figure 3: Three Types of Performance Measures [from 21]

2.2.1 KPIs Manufacturing Enterprise

ISO 22400 provides concept of KPIs in manufacturing enterprise [22]. KPIs in manufacturing enterprises are part of assessment and value creation process. Each manufacturing process consists of set of goals with set of resources and actions within precise timeline. Production systems mainly focus on achieving targets by manufacturing products by given time limit. KPIs provide measurable metrics to evaluate manufacturing processes, monitor trends, support decision making and manage the processes in efficient way [22]. Table 1 shows criteria of KPI in manufacturing enterprise adapted from ISO22400.

Table 1: KPI criterion from ISO22400 [22]

	Name	Description
1	Quantitative	A formal procedure is determined to Estimate the measurement.
2	Relevant	A description is provided to explain the purpose and effects on the objectives, following quantitative details.
3	Comparable	A means to compare the measurements between different time periods with proper unit to express the measurement.

KPIs supports following continuous procedures to improve management in manufacturing facilities [23]:

- Production planning and scheduling
- Upgrade, downgrade, maintain or replace manufacturing resources;
- Design and operate system configurations and applications;
- Improve or adapt architectures in system or application level;
- Other key performance drivers.

Therefore, KPI definitions in manufacturing site mostly contains following properties [23]:

- Trend – maximum, minimum, sum, average, count, standard deviation;
- Sampling –grouped by monitoring context;
- Time tag – current period, sliding interval, fixed interval, last completed period;
- Data filter – the target range value and the range definition.

Next sub section focuses more on energy KPIs to continue the study on holistic energy management system.

2.2.2 State of Art Energy KPI in Manufacturing Facilities

Energy performance of manufacturing facilities is complex and dispersed. In addition, KPIs of holistic EMS must be selected from heterogeneous sources of manufacturing facilities.

Two major sources of KPIs in manufacturing site are Building Automation System (BAS) and Factory Automation System. Heterogeneity of Factory Energy management systems is discussed in [7]. But the article focused more on architecture of system than identification of energy KPIs. Therefore, exploration on energy KPIs study is separated into two domains: BAS and FAS.

There are enormous numbers of researches and implementations have been done in KPI design and definition for FAS. Seven common KPIs for production monitoring are described in [24]as Count (good or bad), Reject Ratio, Rate, Target, Task Time, Overall Equipment Effectiveness (OEE) and Downtime. On the other hand, six case studies were briefly discussed in [25] to exemplify implementation and limitations of currently widely used KPIs in FAS. The research also listed set of KPIs with definition based on run-time data acquisition and extraction. Then it generalized methodology and implementation that can be adjusted in any manufacturing facilities. The study was tested in a simulated production line and claimed to achieve desired result for monitoring production KPIs.

The term Energy Use Parameters (EUP) is mentioned as metric functions to identify the state of the energy use in manufacturing process management [26]. EUP can be calculated from measured data in shop-floor. A general definition for EUP is presented in equation below, where Input Energy is total amount of consumed energy and Output means ready components, products or production for the same period.

$$EUP = Input\ Energy/Output$$

A review on the sustainable manufacturing indicators performed considering Automotive Companies [27]. The study considered three factors: environmental, social and economic performance to evaluate sustainable manufacturing. In order to integrate sustainability with manufacturing performance, a set of initial KPIs for automotive companies also proposed in [27].

Production activities in shop-floor are divided into different operational conditions [9]. And Energy consumption is classified in two types like; normal or waste. The study described how operators in shop floor can also contribute in energy performance improvement by meaningful energy information presented to them [9].

On the other hand, another set of KPIs has been developed and implemented to evaluate energy performance in buildings [28]. Although this paper is focused on commercial buildings, it can be adapted for manufacturing BAS as well. The study was intended to support stakeholders in energy efficient decision-making by presenting building energy consumption information.

In addition, The International Telecommunication Union (ITU) listed a large number of KPIs to improve environmental sustainability [29]. The list divided in five main groups of environmental KPIs: energy, GHG, emissions, waste, water and others. These KPIs are reported on generic structure of all organizations and their functionalities.

Several methods have been discussed in literature and practices to realize KPIs in factory EMS. However, the data integration coming from heterogeneous sources (i.e. MES, and ERP etc.) increases complexity of the system. Adaption of Service Oriented Architecture (SOA) emerged as a tangible solution for such complexity.

2.3 Service-oriented Architecture (SOA)

In recent years, SOA has become a popular choice in development of software systems. According to definition of World Wide Web Consortium (W3C) SOA is a conceptual model of system architecture that refers *‘A set of components which can be invoked, and whose interface descriptions can be published and discovered’* [30].

From numerous profound definitions, it can be said that SOA represents a system of multiple autonomous applications combined by orchestrating loosely coupled services [31] running on different endpoints. These services have open interface to allow other applications invoke independently and form different business processes. Main principles of SOA are listed in Table 1, adopted from [32].

Table 2: Principles of SOA [32]

Name	Description
Loose coupling	Services operate independently. They have full control over the encapsulated logic.
Interoperability	Implemented by adopting open standards that allows services to be developed in different platforms but communicate with each other.
Reusability	Services are reusable and can be shared with other services.
Discoverability	A key feature of SOA, services are publishable with unique interface and service consumers can find it to use them.
Composition	It enables building and modifying business processes to create new application or upgrade existing one.

SOA is preferred solution for complex systems where data is integrated from heterogeneous sources like hardware platforms, application formats or operating systems. Basic components of SOA are service providers, service consumers and service broker. In short, consumer requests broker for information about particular service. Then broker looks for service from available service providers and return the information to consumer. Then consumer requests provider for services and gets response.

The overall communication resembles client/service paradigm. Figure 4 illustrates relationship among these components.

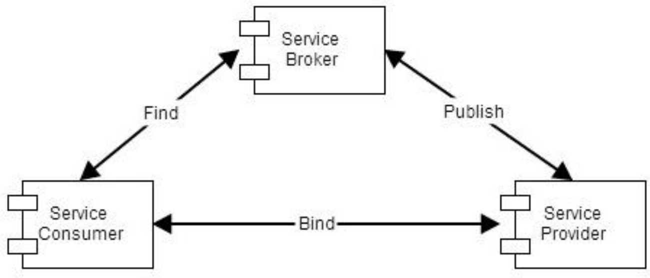

Figure 4: Basic Components of SOA

SOA can be implemented in many ways. Some of these standards were discussed and compared in [33]. Most of SOA standards and its protocol stack are presented in Figure 5.

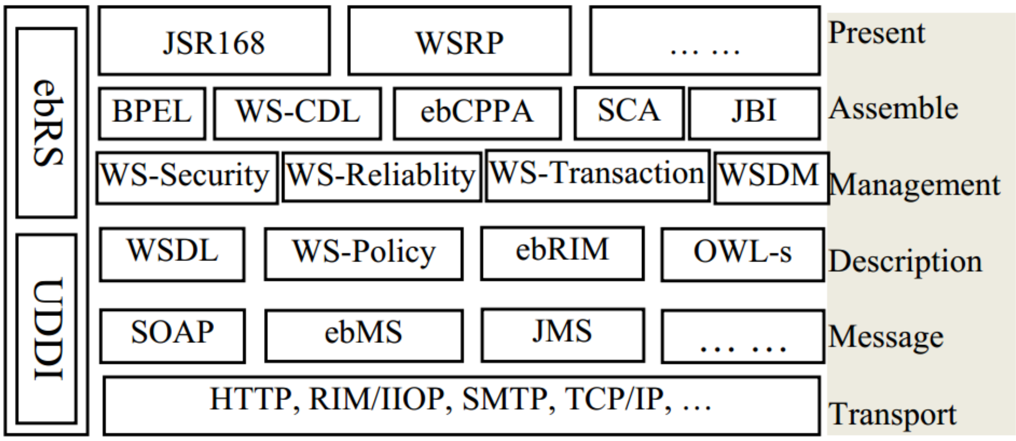

Figure 5: The SOA protocol stacks [33]

Most recommended platform of implementing SOA is Web Services set of standards [31][31].

2.3.1 Web Services:

"*A Web Service is a self-describing, self-contained, modular application accessible over the web. It exposes an XML interface, registered and can be located through a Web Service registry*" [33]. Web Services consists with set of standards. Among these standards, Web Services Description Language (WSDL) is used for service interfaces, service data types are realized by XML Schema Definition (XSD), messages are formatted by Simple Object Access Protocol (SOAP), then exchanged by Hypertext Transfer Protocol HTTP, Business Process Execution Language (BPEL) for service orchestration code and services are listed and discovered by Universal Description, Discovery and Integration (UDDI). Each Standard of WS is consistent with each other and creates a level of abstraction and functionality for WS based application development. Figure 6 illustrates some of these WS standards in stack.

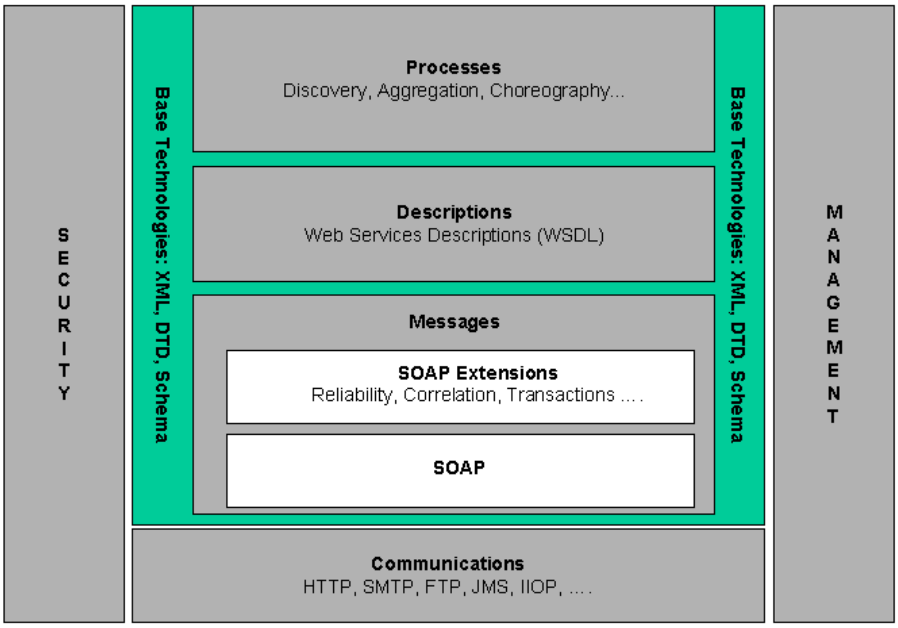

Figure 6: WS Standards stack [34]

Together with all these standards, WS guarantees loosely coupling, flexibility, reusability and compatibility of SOA paradigm [33]. Correlation between principles of SOA paradigm mentioned in Table 2 and features of web services is presented in Table 3.

Table 3: Alignment of SOA and Web Services

SOA Principle	Feature of Web Services
Loose coupling	WS is self-describing, self-contained and modular. Thus It is capable being loosely coupled and provide autonomous services
Interoperability	As XML is a platform independent messaging language, WS is interoperable.
Reusability	WS are reusable and can be shared with other services.
Discoverability	UDDI enables discoverability of WS using unique service interface of WSDL
Composition	Composition of Web services are choreographed using standard BPEL.

Because of these strong similarities many people confuse WS with SOA. But it is important to revise that SOA is an architectural model of building applications and WS is just one of many ways to implement SOA.

2.3.2 Device Profile for Web Services

Device Profile for Web Services (DPWS) is a standard Web Services profile that enables secure WS capabilities on resource-constraint network embedded devices [35]. It was developed to implement SOA in machine-to-machine communication. DPWS is based on Web Services standards protocol including IP, TCP, UDP, HTTP, SOAP, XML and WSDL Figure 7. It allows devices to send or receive messages using Web services. Devices can also describe a Web service by providing WSDL file or interact with other services using its description. WS-Eventing specification of DPWS enables subscription and receiving events from a Web service.

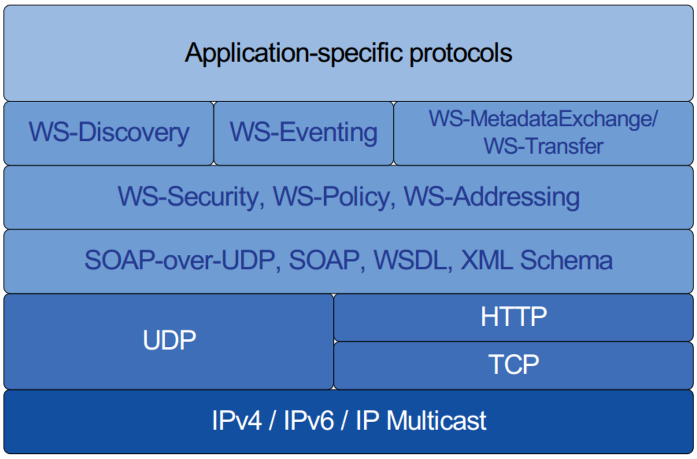

Figure 7: Devices Profile for Web services as protocol stack [30]

DPWS enhanced implementation of SOA in heterogeneous environment. Several European projects immensely contributed to develop DPWS. In SIRENA (2003-2005) project, DPWS implementation was developed focusing embedded devices. In this project, SOA was implemented in embedded devices targeting four different domains: industrial, telecommunication, automotive and home automation [36]. SODA (2006-2008) project aimed to implement SOA in a comprehensive, scalable and easy to deploy eco-system with Service Oriented Architecture for Devices (SOA4D) in wired and wireless factory communication network [37]. The approach of SOCRADES (2009-2009) can be reviewed as the implementation of the SOA paradigm to next generation industrial automation systems. It provided basic concept of cross-layer interaction from the lower levels to the enterprise level of the ISA95 [38].

2.3.3 State of Art Energy Efficiency and SOA in Manufacturing

Since energy efficient factory has become a focusing topic in recent research areas, many implementations and research papers are proposing SOA as an effective tool to implement energy competence in manufacturing enterprises. Approaches to maximize energy efficiency in future factories are outlined in paper [39]. It presented architectural issues of cross-domain holistic energy management. The paper identified cross layer SOA and WS, as unifying technology to implement Cross-Enterprise Energy Efficiency. Components in heterogeneous shop-floors also integrated using Device Profile for Web Services (DPWS).

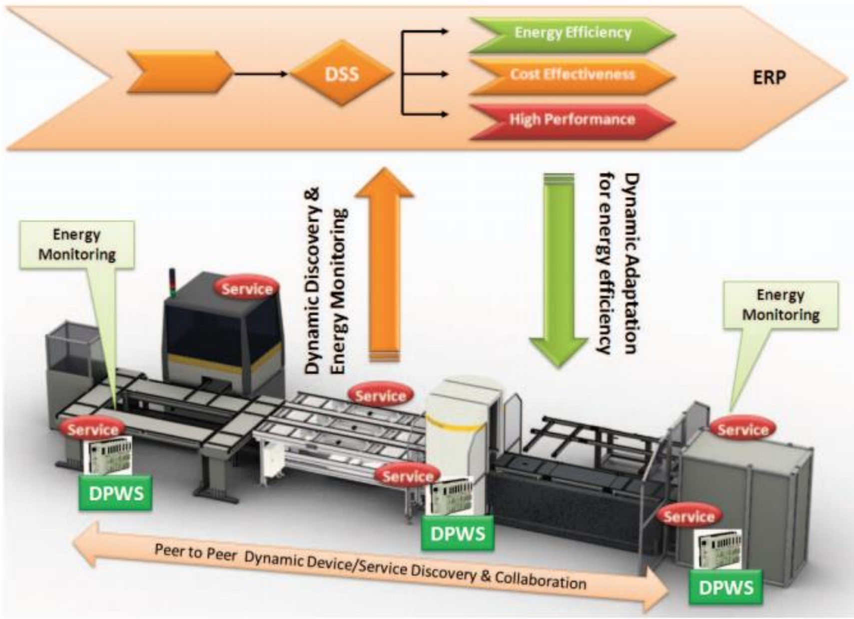

Figure 8: Energy Aware Business Processes and dynamic shop-floor adaptation [39]

Adoption of cross layer architecture using SOA supports seamless re-engineering of business processes [40]. The study [40]proposed approaches to enable near real-time production control decisions introducing two directional optimizations. AmI-MoSES implemented a similar architecture for monitoring energy consumption in manufacturing SME in order to support energy efficient optimization combining data/information with processes [15]. The platform allowed building different context-aware software services for EE optimization.

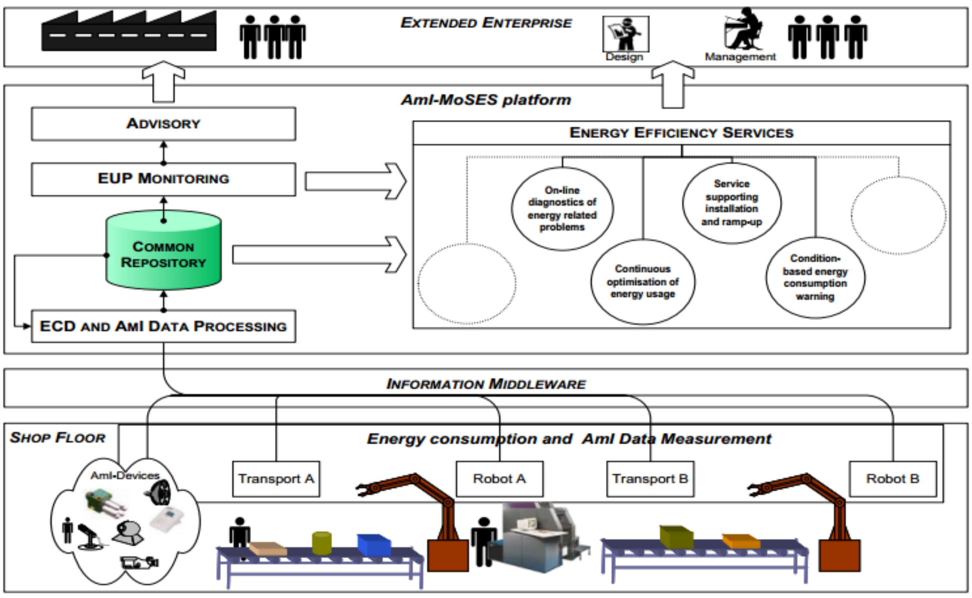

Figure 9: AmI-MoSES System Concept [41]

Cross-layer architecture for energy management was implemented for a manufacturing enterprise in [7]. This architecture is based on Service Oriented Architecture (SOA), Event Driven Architecture (EDA) paradigms and Complex Event Processing (CEP). A Data Acquisition Engine (DAE) has been developed for data retrieval and processing in EMS. The engine is responsible for data aggregation from BAS and FAS systems. It also computes Key Performance Indicators (KPI) and accumulates in the following (BAS or FAS) databases Figure 10.

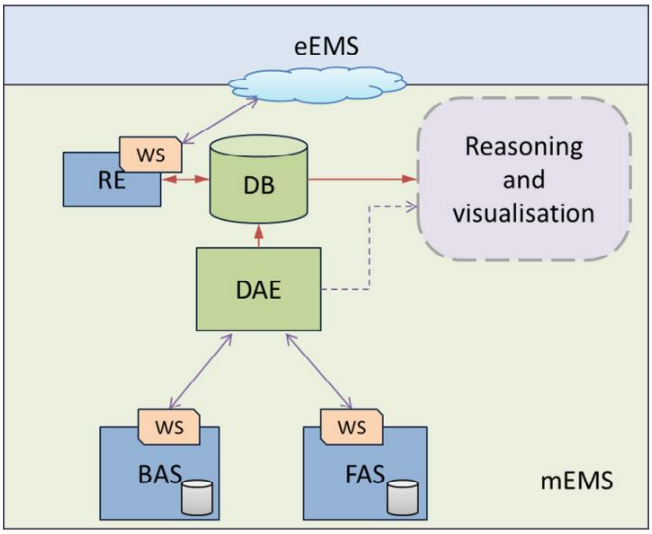

Figure 10: Data retrieval and reporting in cross-layer EMS [7]

EMS is presented as "System of Systems" (SOS) in [42]. SOS approach proposed to divide EMS architecture into four vertical layers [43]; linked data wrappers of existing systems, linked energy cloud consisting linked data space at the information-level, support services and energy management applications Figure 11. Communications interface between these layers are implemented adapting SOA paradigm.

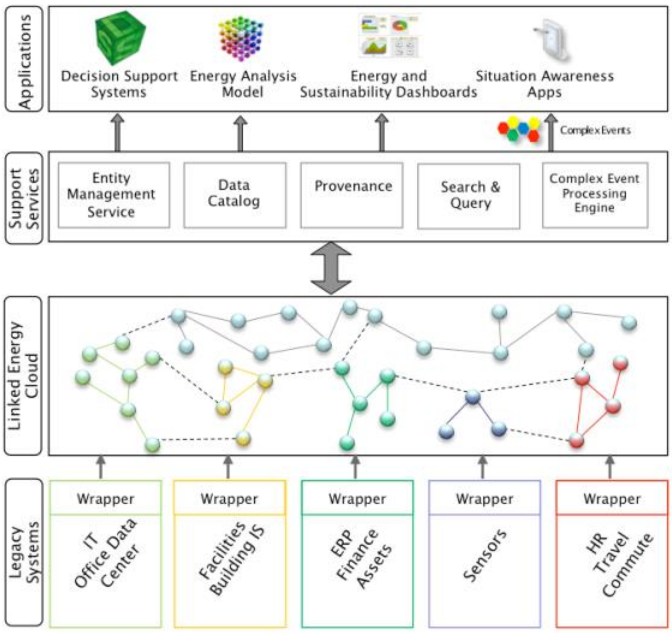

Figure 11: SOS concept of EMS [43]

2.4 Review of Energy Management Solutions

There are several commercial EMS currently available in market but very few information and documentation are available in public. Each of these EMS has its strong and weak sides. However, without direct feedback from enterprises or hands on user experience it is hard to review on these EMS solutions. EMS Solutions like Enerize, e&eco-F@actory , EnergyCAP[1] Energy Management System, SAP Energy Management System[2] and Power logic Energy Management System[3] are studied from existing demo and there official website during thesis implementation. Following sub sections review on Enerize E3 and e&eco-F@actory in details because these two solutions had enough online material to draw an assessment.

2.4.1 Enerize E3 Factory Energy Management System

Enerize E3 is an industrial EMS focusing visualization of energy information within in-plant production activities [44]. Enerize E3 uses KPIs to identify energy saving points and proposes KPIs as management criteria for energy efficient decisions.

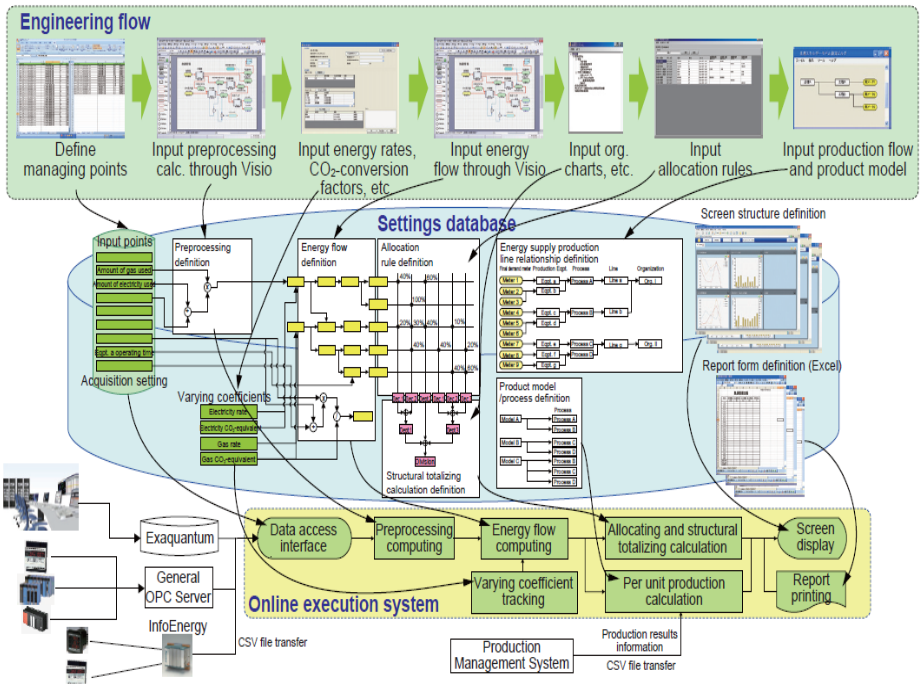

Figure 12: Functional Software Configuration Enerize E3 [44]

However flexibility in new KPI definition is not mentioned in its solution. It recognizes energy flow within manufacturing enterprise to identify excessive energy

---

[1] EnergyCAP Energy Management Software [Accesed online on January 15th 2014] Available: http://www.energycap.com/

[2] SAP Energy Management Software [Accesed online on January 15th 2014] Available: http://www.sap.com/solution/lob/sustainability/software/overview/highlights.html

[3] Power Logic Energy Management Software [Accesed online on January 15th 2014] Available: http://www.powerlogic.com/literature/PL_product_range_overview_3000BR602R109.pdf

uses in energy supplying equipments, energy consuming equipments and production activities. However this solution skipped BAS integration to monitor energy consumption by facilities. Enerize E3 calculates and visualizes energy costs and CO2 emissions by each department or equipment which ensures energy aware decisions by corresponding bodies. The system offers web based solution with different dashboards to each department considering their role and activities. Figure – shows functional software configuration of Enerize E3. A case study of Enerize E3 implementation is described in [45].

2.4.2 e&eco-F@actory

e&eco-F@actory is a set of hardware and software solution of EMS offered by Mitsubishi Electric. It interrelates and manages production information and energy information by combining EMS and MES. The objectives of e&eco-F@actory are; reduce equipment standby time, shorter tact time, shorter lead time, shorter production time, pre-production time loss and reduce frequency troubles [46]. This solution followed PDCA cycle by dividing approaches in four continuous operations: measurement, visualization, reduction and management of energy. However, solution does not offer retrofitting hardware components for existing factories. As a result, management of energy in shop floor is possible when the whole system is bought from Mitsubishi.

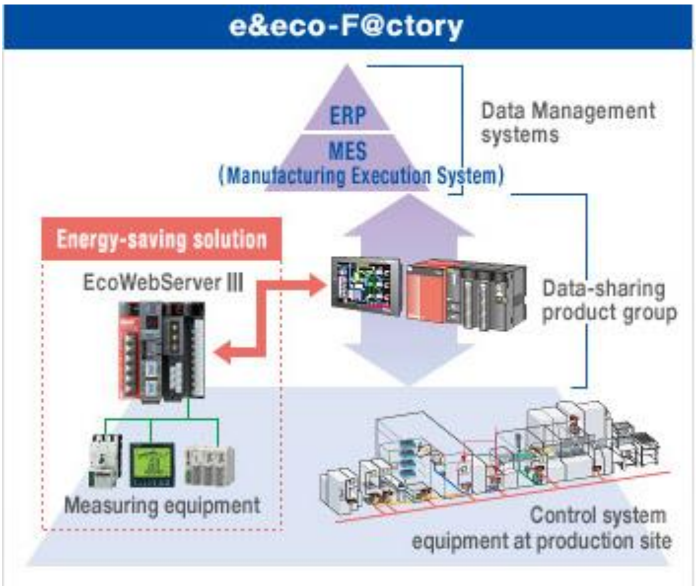

Figure 13: e&eco-F@actory Model [46]

2.5 Portlet applications

Model of SOA allows developing applications from existing services. Portlets enable plug and play behaviour of SOA at user interface layer. This section describes JSR (Java Specification Requests)-000286 Portlet Specification in order to explore its feasibility for Energy Management Portal.

2.5.1 Portlets

Portlets are Java component defined as "*pluggable user interface component that provides specific content, which could be a service or information from existing information systems. Portlets provide the user interface of the portal by accessing distinct applications, systems, or data sources and generating markup fragments to present their content to portal users*" [47]. IBM referred portlets as "*reusable web modules that provide access to web-based content, applications, and other resource*" [48]. Portlets are organized and displayed within portal. Portlet generates markup fragment such as HTML, XML, or WML in a portlet window. A portal page displays a group of non-overlapping portlet windows.

Portlet has a lifecycle within portal architecture. Life cycle defines how a portlet is loaded, instantiated and initialized. It also manages procedures of handling requests from client and rendering response [49].

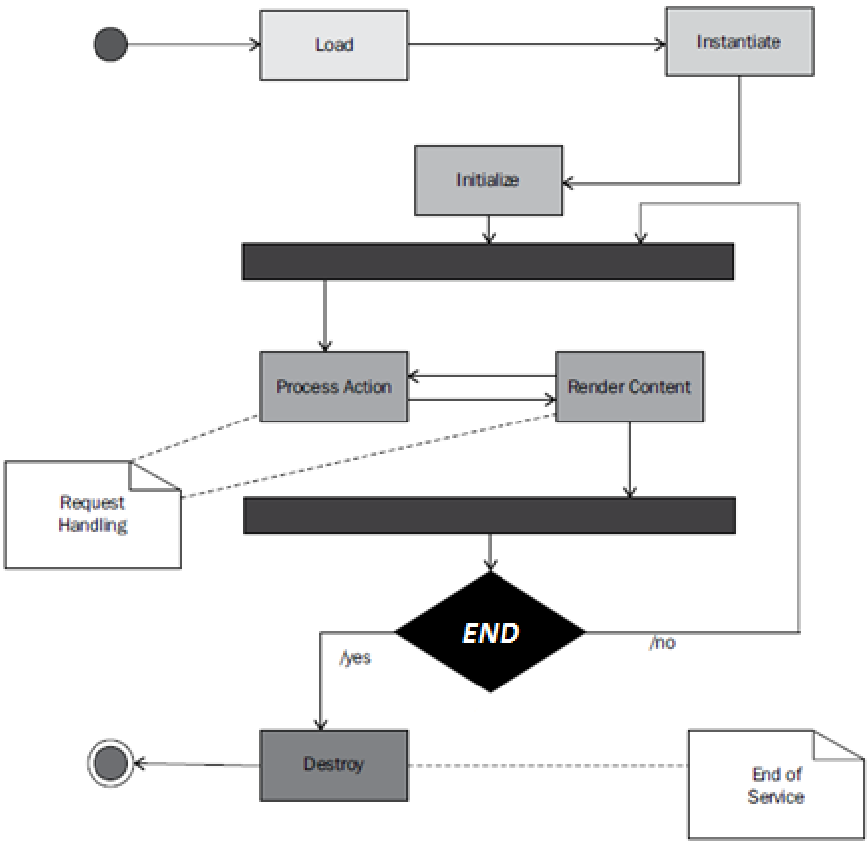

Figure 14: Portlet Life Cycle [49]

- **Loading and instantiation:** The loading and instantiation take place when the portlet container initiates/deploys portlet for the first time. It typically also occurs when server starts/restarts.
- **Initialization:** Portlets are initialized when user enables the portlet in a portlet window.

- **Process Action:** Process action prompts on several events: load of a portal page containing portlet, initialization of portlet within a page or users action to update content.
- **Render Content:** Render content is a consequent phase of process action in order to update contents of portlet.
- **Destroy:** When user removes portlet from service

Portlet vs. widgets

Widgets also called as gadgets can generate contents from different data sources on a web page. Widgets can be developed using Java Script and XML. At a first glance many people confuse widgets with portlets. But architecture of widget and portlets are completely different.

Portlet has a sophisticated API to regulate session management in server-side, internal communication between portlets, request processing and so on. Infrastructure of a portal provides lifecycle management of portlet instance, pooling, content caching, security, and single sign-on features to web portal [47]. Contents generated by portlets can be shared only within Portal with proper access privilege. On other hand, widgets directly interact with data source to generate content Figure 15. Content of widgets has limited security management and mostly accessible through any web page. Until now widgets are not well explored for complex integrated systems. Figure 15 shows how portal server generates portal page by aggregating content of portlets.

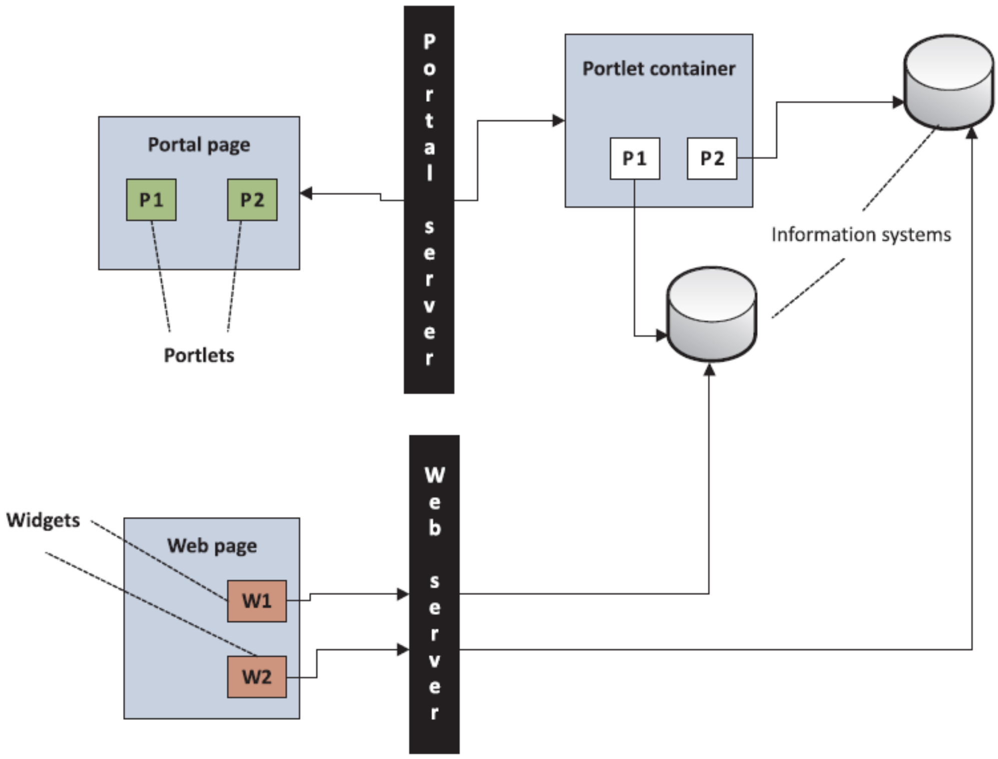

Figure 15: Portlet vs Widgets [47]

Portlet vs. Servlet

Servlet is defined [50] as "*A Java technology based web component, managed by a container that generates dynamic content. Like other Java-based components, servlets are platform independent Java classes that are compiled to platform neutral bytecode that can be loaded dynamically into and run by a Java enabled web server. Containers, sometimes called servlet engines, are web server extensions that provide servlet functionality. Servlets interact with web clients via a request/response paradigm implemented by the servlet container*".

There are many similarities in servlets and portlets. Both of these are Java technology based web components with container and life cycle mechanism [51]. Following table indicates difference between portlets and servlets:

Table 4: Difference between Portlets and Servlets

	Portlets	Servlets
Render Method	Generate markup fragments of cHTML, XML or WML in render method and Portal infrastructure aggregates fragments of different portlets to a portal page.	Generate complete page in render method
Access	Portlet can only be accessed via a portal page. Portal uses specific portlet API to access the portlets URL. Web clients communicate with portlets through portal page.	Servlets have attached URL and it can be accessed directly by web clients (Browser).
Multiple Instantiations	Same page capable of holding multiple instances of a single portlet.	Single page hold single instance of servlet
Window States	Portlets have three different window states: View, Edit and Help in user interface layer	Servlets have view state only.
Persistent configuration and customization	Portlets support persistent configuration and customization	Servlets can read persistent configuration. It cannot customize it.
Miscellaneous	Portlets unable to set HTTP headers in response. Portlets are not capable of setting character set encoding at rendered response nor can it maneuver URL of client request to the portal.	Servlets can set HTTP headers in the response; It can also manipulate URL of client request.

2.5.2 Portlet Container

"*A portlet container runs portlets and provides them with the required runtime environment. A portlet container contains portlets and manages their lifecycle. It also provides persistent storage for portlet preferences. A portlet container receives requests from the portal to execute requests on the portlets hosted by it.*" [51]. Portal container is an extended modification of servlet container. It comes with all the functionalities of servlet container with additional capabilities of handling portlet instances. Runtime environment of portal container uses PortletContext which is similar to ServletContext object. PortletContext allows portlets to share data between them and with servlets in the same environment. The structure diagram of Figure 16 illustrates relationship between portlet, portlet container and portal server.

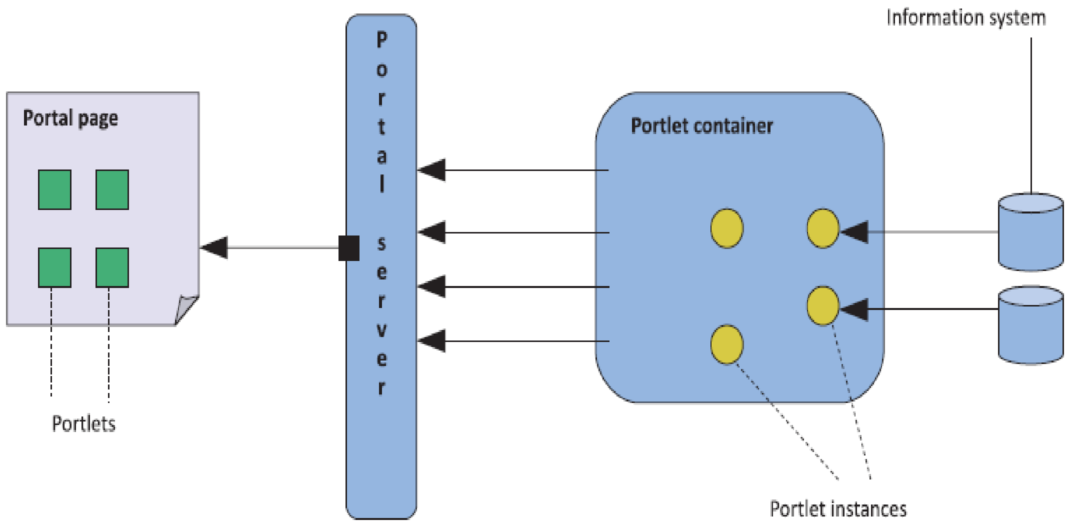

Figure 16: Relationship between portlets, portal server and portlet container [47]

2.5.3 Portal Server

Portlet container is not in charge of generating portal page. It only generates markup fragments. Portal Server handles requests from portal page to portal container. It also aggregates responses from different portlets and generates portal page [51]. Portal server facilitates administration, single sign on, content management, user management, configuration and data aggregation from different sources to single portal page in portal architecture. The sequence diagram of Figure 17 illustrates life cycle of a request within portal architecture.

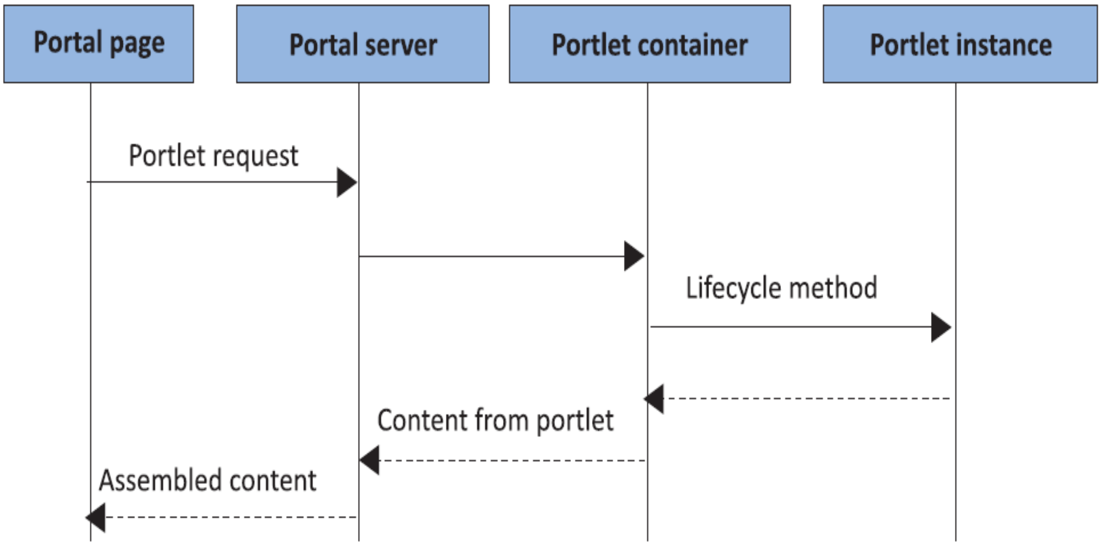

Figure 17: Sequence Diagram of request/ response within Portal [47]

2.5.4 Portal Servers

Following subsection shortly introduces some of available portal infrastructures. Portal infrastructure is software component that includes portlet container, portlet server and additional features of web portals.

Liferay

Liferay is a portal infrastructure to deploy and develop composite portal applications. Liferay adapted SOA in its core architecture. Portlets inside Liferay are loosly coupled and able to communicate with each other. External applications can also interact with portlets using APIs via REST, SOAP, RMI, XML-RPC, XML, JSON, Hessian, Burlap, and custom-tunnel classes [52].

Enterprise Layer

The enterprise layer of Liferay is consists of enterprise functionalities of Portal Management, Content Management, Workflow Management, Document Management, User Management and Security Management.

Service Layer

Model Driven Architecture approach is followed in Lifray's service layer. Liferay adapted Platform Specific Model (PSM) utilizing EJB, Spring Framework, Hibernate Layer and Web Services. This collection of technologies enabled Service Builder Layer which enabled seamless data integration from heterogeneous sources.

Figure 18: logical Architechture of Liferay [52]

eXo

eXo is another Portal Platform. It is built on concept of platform-as-a-service (UXPaaS). EXO has also been popular for building and deploying transactional web-sites and creating gadgets and dashboards. eXo allows building complex applications with its native API and REST architecture. eXo has a web based IDE for building applications, gadgets and mashups. Figure summarizes architecture of eXo [53].

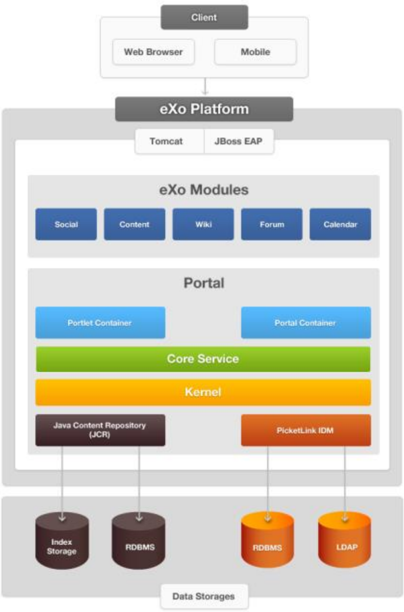

Figure 19: eXo architecture [53]

Although in this thesis EMS is developed using Liferay, eXo can be feasible solution for such complex applications as well.

Alfresco ECM

Alfresco ECM is also a popular portal infrastructure. But its architecture and model is more focused on intranet management, document management and content management for medium or large corporations [54]. The architecture of alfresco is presented in following figure.

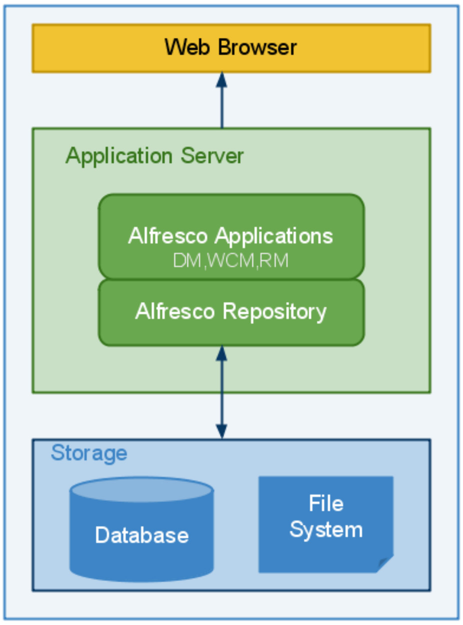

Figure 20: Architecture of Alfresco [54]

This short review is done in order to familiarize reader with some current portal infrastructure technologies in practice. Eventually, Liferay has been chosen to develop EMS for its SOA paradigm, open source license, clear documentation and active community support. More about these technical approaches will be further discussed in following chapters.

3. METHODOLOGY

A variety of research and implementations of EMS are studied as theoretical background in previous chapter. However, methodological approaches of this work are quite novel for EMS domain. This work enabled SOA architecture in user interface level by implementing EMS using portal infrastructure. Furthermore, it followed current EMS standards ISO: 5000. This chapter particularly describes the selected technologies and tools for the design process of the Energy Management Systems.

3.1 Application Architecture

Overall application architecture of designed EMS is presented in Figure 21. EMS is developed on existing application layers in SOA enabled manufacturing facility.

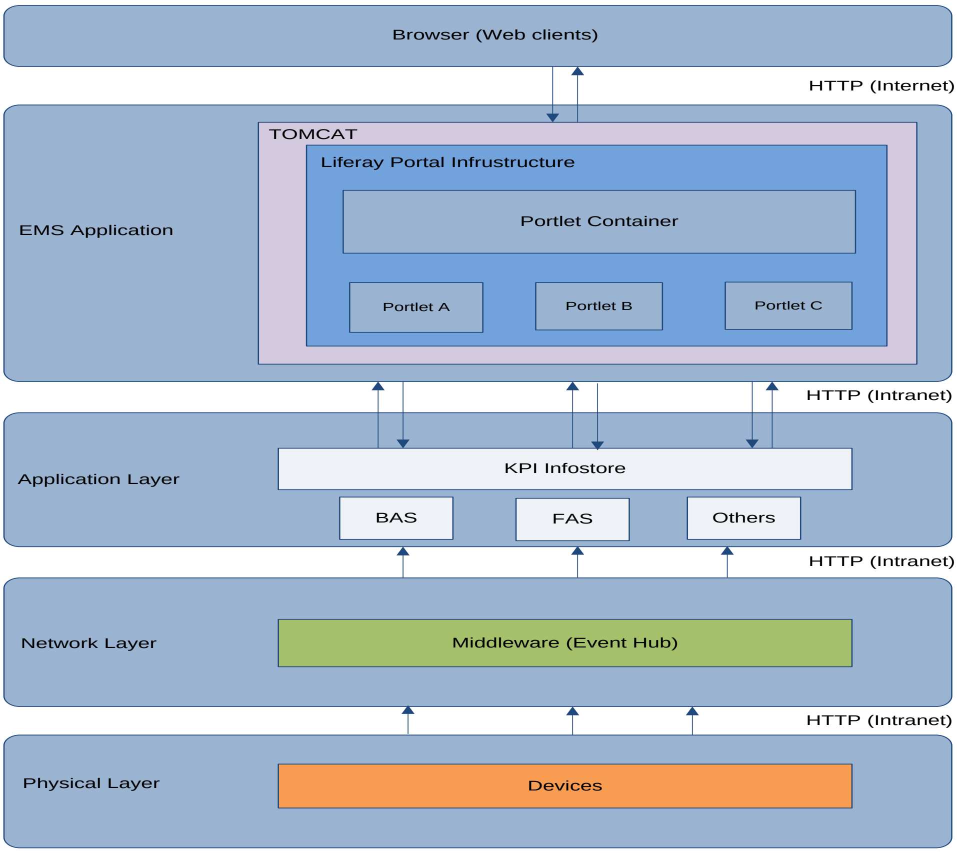

Figure 21: Application Architecture EMS

Physical Layer in portrayed EMS architecture represents all the devices and components directly involve in energy consumption. Network layer represents a middleware application (Event-Hub) that accumulates data from devices and opens a single interface for subscription to other existing modular systems i.e. BAS, MAS and KPI info-store. EMS is also subscribed to Event-Hub to get online energy information. Application layer represents all the systems typically exist in SOA enabled manufacturing facility. EMS Application represents the complete energy management system built as implementation part of this thesis work.

3.2 Tools and Frameworks

3.2.1 Liferay

Liferay version 6.1 is used for implementing EMS in this work. Liferay is already introduced as portal infrastructure in previous chapter. After a research on existing portal infrastructures, model, trends and features, Liferay is selected for this thesis work. In this section, features of Liferay are described to facilitate Liferay as energy management portal infrastructure.

There are two ways of building site in Liferay: community and organization. Liferay portal can encapsulate multiple numbers of community sites and organizational sites.

Community

Community displays group of pages and applications to its users with common interest. Each community contains public and private pages. Public pages are visible to non-registered visitors, while private pages are only visible to registered users. Community site may encapsulate multiple organizations but organizations can't encapsulate community sites [52].

Organization

Organization is designed to target companies or institutions with hierarchical user groups. An organization can be assigned as parent or child of another organization. An assigned user of organization inherits permissions and associations of his given organizational role. Organization also has public and private pages. However unlike community, access to contents and applications of pages are assigned to different hierarchical user groups. As a result, same page displays different contents to different user groups. Consequently, organization structure is chosen to implement EMS, which allows hierarchical user and application management in manufacturing enterprise [52].

User

User represents physical users of portal. Each user has account with login, password and other additional information. User may belong to a user group. User groups represent a common role that allocated to all of its users. Teams also correspond to group of users but within an organization or community [52]. Figure 20 illustrates permission model of users in Liferay.

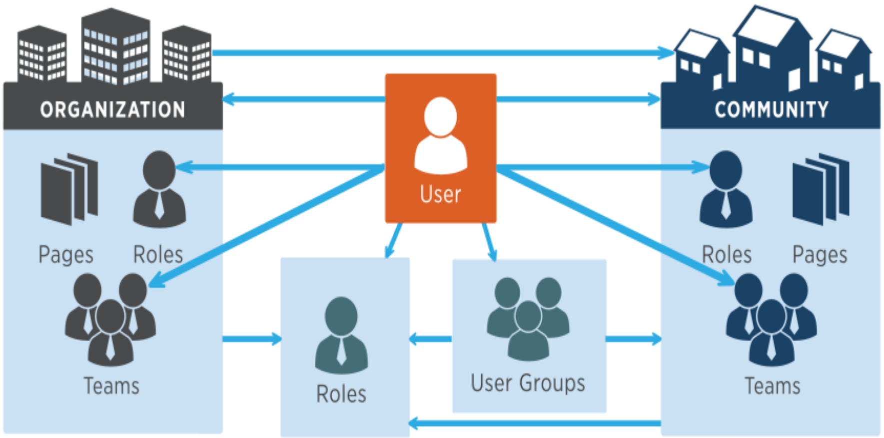

Figure 22: Liferay Permission model of user [55]

This built in feature of Liferay helped developer to avoid concentration on user management and organization management systems in implementation level.

User Role Management

User role management feature of Liferay allows multi-level user role hierarchy in EMS. Organizations have administrator role for their portal. A user with administrator role has the permission to create different roles and assign users in different roles of the organization.

These processes of assigning users to permissions are arranged in user interface level of control panel in Liferay, which allows flexibility and re-configurability in user management of EMS. Figure 23 explains permission and user management in Liferay using entity relationship diagram. It is worth to mention that resource_ table of figure 21 represents drag and droppable objects of portal. It can be portlet, application, or content that is allowed to put in portlet window of portal page. All the portlets developed for EMS belong to resource_entity and organization represents manufacturing enterprise itself. If manufacturing enterprise has facilities in multiple locations, it is possible to implement all this location as child organization. And organization role represents different user groups within organization: for example production managers, facilities manager and so on.

Due to this huge built in facilities of Liferay as portal infrastructure, implementation of EMS becomes more focused and research narrowed its concentration on portlet development, where Liferay solved hierarchical dashboard management, user management and portal administration features of EMS portal.

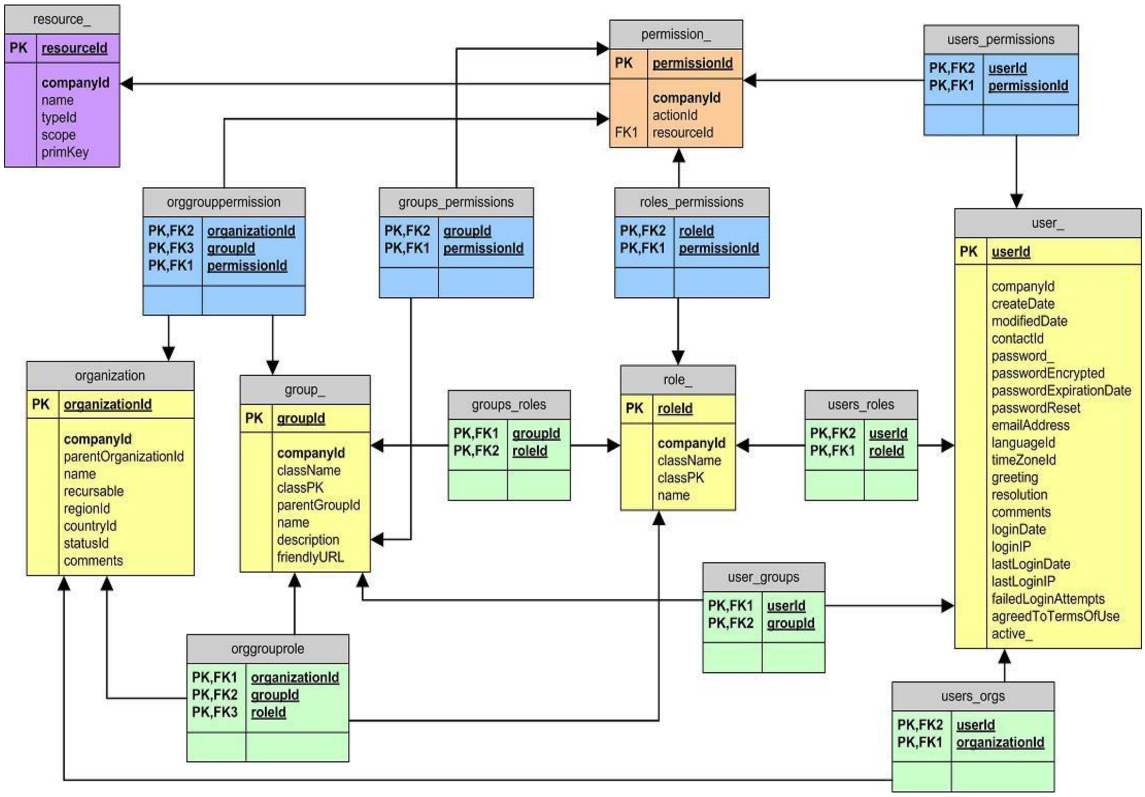

Figure 23: Entity Relationship Diagram of user management in Liferay [56]

3.2.2 Portlet Development Tool

Liferay has a development environment called Plugins SDK [57]. It is possible to develop portlets from command line and file editors like Emacs, Vim or even Notepad. However, There is an extension of Liferay IDE for Eclipse that extends its functionality to facilitate developing all types of Liferay plugins. The extension is also based on Plugins SDK. In addition, an extension to NetBeans called Portal Pack also available for developing plugins of Liferay. In this thesis work, Eclipse has been used as IDE for portlet development of EMS.

3.2.3 Portlet Development

A generic portlet configuration and structure is discussed in this section, therefore it is not repeated in all the implementation of portlets. Then portlet wise classes and model will be elaborated in each portlet application described in Implementation chapter.

A Portlet is consists with three minimum basic components:

1. Java sources
2. Configuration Files
3. Client side files

In Liferay's SDK environment portlet files follow a standard directory structure illustrated in Figure 24.

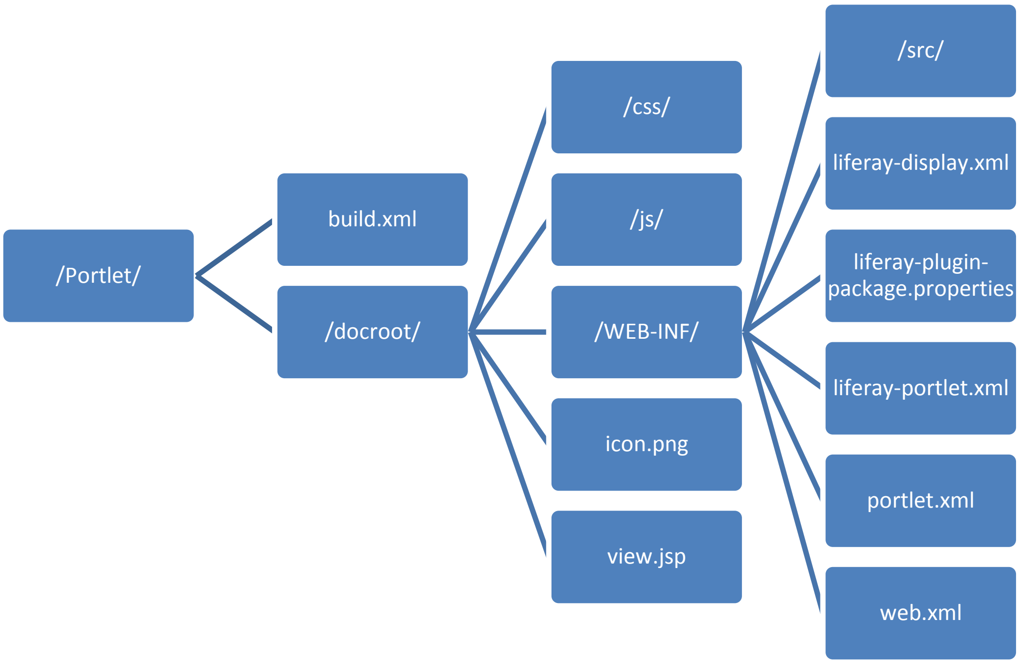

Figure 24: Portlets Directory Structure

MVCPortlet framework is used to configure portlets. Each view fragment of MVCPortlet uses separate JSP for example; view.jsp for view mode. Java Source classes are stored in the docroot/WEB-INF/src/ directory. And XML Configuration files are stored in the docroot/WEB-INF folder. portlet.xml file is standard JSR-286 portlet configuration. But rests of Liferay's specific configuration files in Figure 24 are necessary to deploy portlet in Liferay portal server.

Figure 25 presents structure of portlet.xml that follows JSR-286 specifications:

- The portlet-name element represents canonical name of the portlet. It should be unique in portlet application. It is also referred as the portlet id within Liferay Portal.
- The display-name element is the short name that is displayed in drag and drop menu. The display name does not need to be unique.
- The portlet-class element includes the main class that handles all invocations to the portlet.
- Initialization parameters of the portlet are included in init param element.
- Expiration-cache parameters specify output lifetime of the portlet. If the value is -1 then output never expires.
- The supports element describes supported mime-types. Every portlet must support a view mode.

```
<?xml version="1.0"?>

<portlet-app xmlns="http://java.sun.com/xml/ns/portlet/portlet-
app_2_0.xsd" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://java.sun.com/xml/ns/portlet/portlet-
app_2_0.xsd http://java.sun.com/xml/ns/portlet/portlet-app_2_0.xsd" ver-
sion="2.0">

	<portlet>
		<portlet-name>PortletName</portlet-name>
		<display-name>Portlet Name</display-name>
		<portlet-class>
			fi.tut.fast.ems.activity.controller.PortletClass
		</portlet-class>
		<init-param>
			<name>view-template</name>
			<value>/html/_PortletName /view.jsp</value>
		</init-param>
		<expiration-cache>0</expiration-cache>
		<supports>
			<mime-type>text/html</mime-type>
			<portlet-mode>view</portlet-mode>
		</supports>
		<resource-bundle>content/Language</resource-bundle>
		<portlet-info>
			<title>Portlet Name</title>
			<short-title>Portlet Name</short-title>
			<keywords></keywords>
		</portlet-info>
		<security-role-ref>
			<role-name>administrator</role-name>
		</security-role-ref>
		<security-role-ref>
			<role-name>guest</role-name>
		</security-role-ref>
		<security-role-ref>
			<role-name>power-user</role-name>
		</security-role-ref>
		<security-role-ref>
			<role-name>user</role-name>
		</security-role-ref>
	</portlet>
</portlet-app>
```

Figure 25: Portlet.xml configuration

- The resource-bundle is a flexible element to implement multilingual portlet application.
- Portlet-info contains information about portlet.
- The security-role-ref element is used to define security role reference of the portlet. It characterizes which role can access the classes within application.

While portlet.xml describes JSR-286 standardizations, Liferay has its own classification described in liferay-portlet.xml. This file locates in the same directory of portlet.xml.

```
<?xml version="1.0"?>
<!DOCTYPE liferay-portlet-app PUBLIC "-//Liferay//DTD Portlet Application
6.1.0//EN" "http://www.liferay.com/dtd/liferay-portlet-app_6_1_0.dtd">

<liferay-portlet-app>
	<portlet>
		<portlet-name>PortletName</portlet-name>
		<icon>/icon.png</icon>
		<instanceable>false</instanceable>
		<header-portlet-css>
			/css/productionportlet.css
		</header-portlet-css>
		<footer-portlet-javascript>
			/js/productionportlet.js
		</footer-portlet-javascript>
		<css-class-wrapper>
			productionactivity-portlet
		</css-class-wrapper>
	</portlet>
	<role-mapper>
		<role-name>administrator</role-name>
		<role-link>Administrator</role-link>
	</role-mapper>
	<role-mapper>
		<role-name>guest</role-name>
		<role-link>Guest</role-link>
	</role-mapper>
	<role-mapper>
		<role-name>power-user</role-name>
		<role-link>Power User</role-link>
	</role-mapper>
	<role-mapper>
		<role-name>user</role-name>
		<role-link>User</role-link>
	</role-mapper>
</liferay-portlet-app>
```

Figure 26: liferay-portlet.xml configuration

The liferay-porlet.xml enhances standard java portlet specifications to comply with Liferay standard.

- The portlet-name element contains the same canonical name that is declared in porlet.xml.
- The icon element provides path to icon image of the portlet
- The instanceable element indicates multiple instantiations ability of the portlet. If it is true then portlet can be instantiated multiple times in same page.

- header-portlet-css and footer-portlet-javascript elements contain path to .css and .js files consequently. This files are intended to be loaded when portlet loads in a page

On the other hand, liferay-display.xml file contains information of category the portlet that appears in the dockbar from where user can drag and drop the portlet to intended pages.

```
<?xml version="1.0"?>
<!DOCTYPE display PUBLIC "-//Liferay//DTD Display 6.1.0//EN"
"http://www.liferay.com/dtd/liferay-display_6_1_0.dtd">

<display>

	<category name="FASTORY">
		<portlet id="PortletName"></portlet>
	</category>
</display>
```

Figure 27: liferay-display.xml file configuration

3.2.4 Apache Camel Framework

Apache camel is a java based open source integration framework to integrate different enterprise applications in SOA paradigm. It is based on common Enterprise Integration Patterns (EIP). EIP was developed to standardize integration process of enterprise applications [58]. The core of Camel is a routing and mediation engine, which is responsible for moving a message between different applications based on the route's configuration. Route's configuration is formed using EIP and a Domain Specific Language (DSL) [59]. Route is defined as integration path between different endpoints, where path represents from sources to destinations. Endpoints are identified by Unified Resource Identifier (URI). Camel uses "CamelContext" object which is a Camel runtime system. Each application typically has one "CamelContext" that contains routes and components, where components are responsible for creating endpoints [60].

Apache Camel version 2.10.3 is used to implement middleware (Event-hub) in SOA-enabled manufacturing facility, which receives all messages from energy meters, CAMXstates and THL sensors and machine vision systems and reroute them to other enterprise applications. One of portlets of EMS uses Camel framework to consume online energy information from Event-Hub. Then portlet utilize Direct Web Remoting (DWR) library to stream the consumed messages to browsers.

3.2.5 Direct Web Remoting (DWR)

DWR is a Java based open source remote procedure call (RPC) library to call Java functions from JavaScript or vice versa [61]. DWR 2.0 introduces "Reverse Ajax", an effective way to implement event driven communication between servers and web browsers. Reverse Ajax of DWR generates JavaScript API which allows seamless communication between Java functions running on server side and updates to arbitrary groups on browsers. Common supporting methods of DWR are: Comet, Polling and Long Polling. Therefore, DWR consists of two parts: A server side Java Servlet that generates requests and sends responses back to the browser and A Javascript library that communicate with server side functions and update in browsers accordingly.

DWR version 2.0.10 is used in EMS to get near real time energy information of manufacturing site in web browsers. Then the information is presented using High-charts. More about Highchart is discussed in following subsections.

3.3 JAX-WS

While Camel and DWR are used to integrate online information flows, JAX-WS is used for offline service compositions and consumptions of SOA-enabled factory. JAX-WS is a Java based Web Services Framework that provides tools and infrastructure for creating and consuming web services [62]. It defines WSDL to Java mapping or vice versa, which determines operations between Java methods and WSDL through SOAP requests and responses [63].

JAX-WS version 2.2 is utilized in several portlets of EMS to invoke services from other applications like KPI generator, MES application and BAS as well.

3.4 HighCharts

Highchart is a HTML5 and JavaScript based charting library for building interactive charts and diagram in client side of web applications. HTML5 enabled the use of standards like Scalable Vector Graphics (SVG) and Canvases which allow Highchart to dynamically update diagrams and charts based on clients request or asynchronously as well [64].

Since EMS is intended to develop as a platform independent application with capabilities of browsing from office computers to phones and tablets where interactivity also plays an important role, Highchart version 3.0 is selected as supportive tools for data representation in browsers which supports area, angular, bar, column, error bars, funnel, pie, line, spline, scatter, gauges, columnrange, bubble, box plot and many other chart types. Graphical presentation of energy information helps user groups to understand big data at glance realize trends and take decisions promptly. A detailed description about implementation of these methodological approaches is presented in next chapter.

4. IMPLEMENTATION

Followed by Methodological approaches demonstrated in previous chapter, this chapter begins by describing a generic architectural model of EMS and how it fits in ISO: 50001 standardizations. Then it introduces manufacturing Test-bed for the implementation of EMS used in this work. After that, a set of KPIs is identified. Finally, how the identified KPIs are implemented in different portlets is elaborated.

4.1 EMS Architecture

As studied numerous research and implementations in previous chapters, energy consumption in any manufacturing industry can be shown by following illustration:

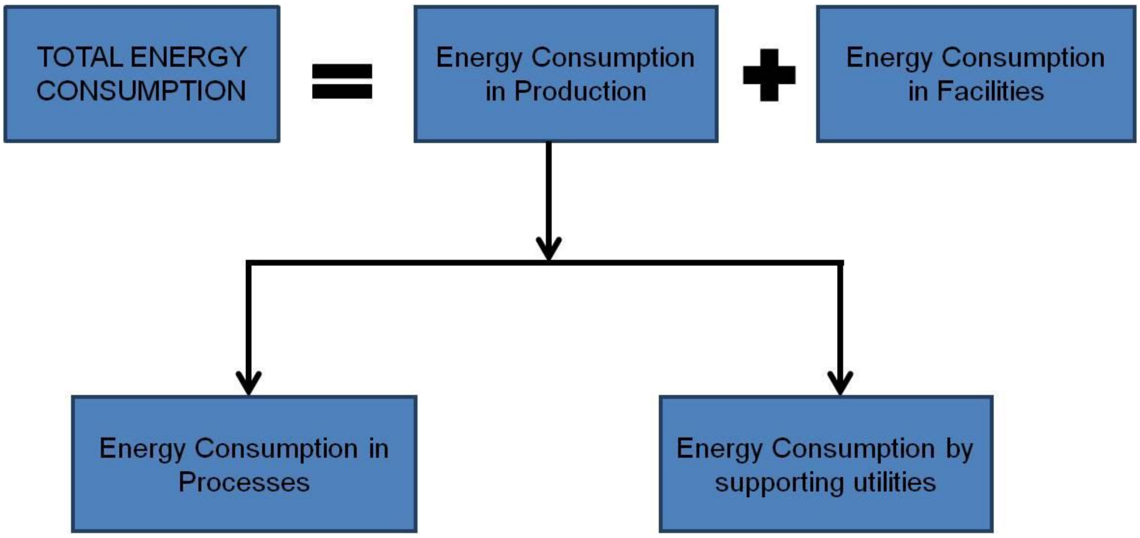

Figure 28: Total Energy Consumption in Manufacturing Facilities

Consumption of energy in production can be categorized as “Energy Consumption in Processes” (consumption by Conveyors, Controllers and Robots in different states) And “Energy Consumption by supporting utilities” (consumption by workshop lighting, machine vision, compressed air flow etcetera). On the other hand, energy used by space and heating, air conditioning, cooling and office equipments is part of facilities management. Certainly, it is necessary to upgrade existing equipments and introduce energy efficient machineries. However, in order to see significant and long term achievement, ISO: 50001 approaches are implemented in this work by modifying PDCA cycle in following strategy illustrated in Figure 29.

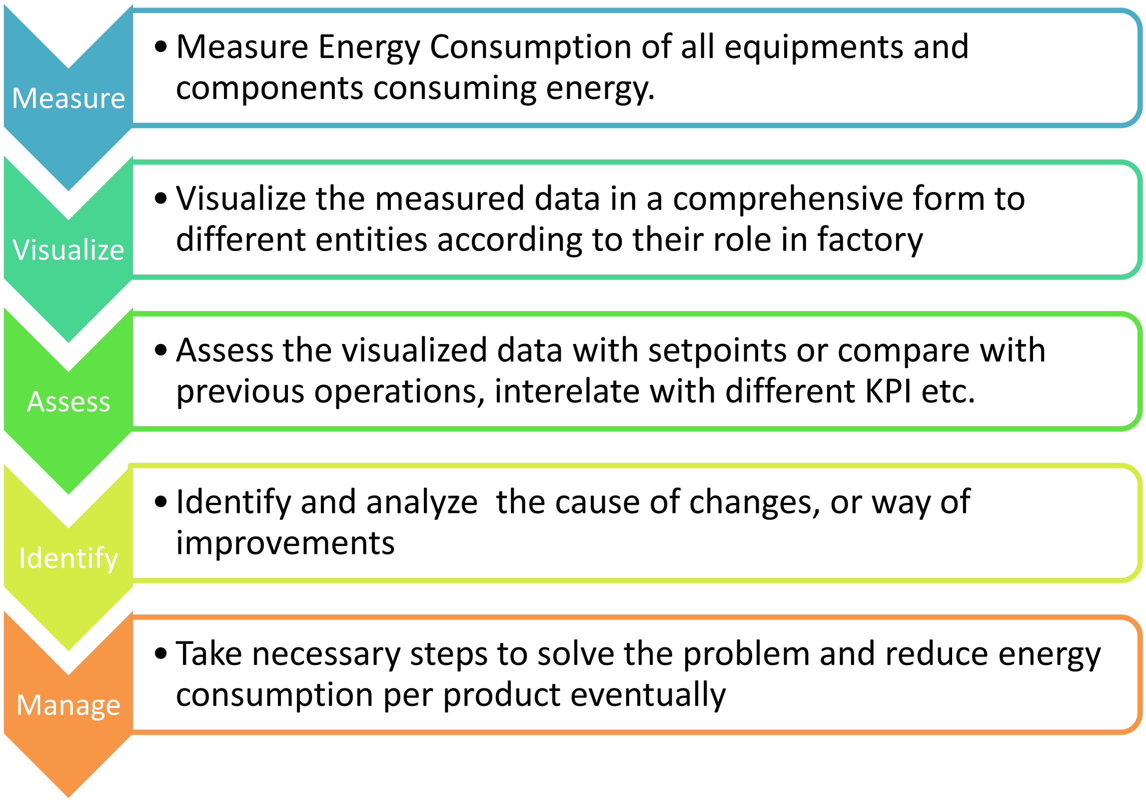

Figure 29: Steps implemented in EMS

- **Measures**: In this step, a technique is developed to measure energy consumption of all equipments in manufacturing facility. It not only includes equipments directly involve in manufacturing processes but also supporting utilities, lighting, HVAC and other components responsible for energy consumption. It is implemented in existing facilities by retrofitting energy meters with DPWS capability. In SOA enabled factories, DPWS is already widely used in shop-floor devices. Thus, existing SOA paradigm and its FAS application infrastructure is used to accumulate energy information and save in FAS and BAS databases as another service. In addition, DPWS also allows integrating shop-floor with EMS and KPI info-store directly.
- **Visualize**: In this step, Firstly, Role and activities of different user groups are studied. Studying about responsibilities and activities of different user group with ISA 95 standard provided a clear idea about hierarchical user groups in manufacturing facilities [65]. In this implementation three user groups are identified to design dashboards: Production Managers, Facility Managers and Operators. As portal infrastructure allows dashboards design steps in UI level, new user group creation and configuration will not require any further development of the system. EMS portlets communicate with FAS, BAS, KPIs info-store and middleware applications to collect energy information. And collected information is presented to different dashboards through graphical presentation using

highcharts API. Adaptation of ISA 95 standards allocated right information to the relative user groups based on their role and activities.

Although five steps are depicted in Figure 29, Measure and Visualize steps are the actual implemented phases of EMS. Remaining three steps are organizational acts by its user groups. Thus, proper visualization of energy information is needed, with the help of identified KPIs to support individuals in order to monitor trend and assess overall energy activities in organization.

- **Assess**: In this step, different user group assess energy performances based on visualized energy KPIs.
- **Identify**: Based on the assessment in previous step, individuals identify reasons like cause of excessive energy consumptions or way to develop energy efficiency within his scope.
- **Manage**: Eventually based on identified points individuals develop or reschedule their activities to improve energy efficiency. This act of each individual eventually leads to ultimate energy efficiency in organization.

The architecture of EMS is designed to implement the steps mentioned in Figure 29. Figure 30 shows the generic physical model of EMS that can be adapted in any manufacturing facilities. 1 and 2 in this figure symbolizes all the equipments and components that publish their energy information using DPWS through wireless or Ethernet. Event hub 3 is a web application running on a local server as middleware that receives all incoming data flow and maintain its reliability. It also provides a single interface to FAS, BAS and EMS to communicate with shop-floor devices. 4 is presenting application servers with databases belong to factory automation systems and building automation systems which subscribe all corresponding data flows and accumulate in local servers. EMS portal server in 5 installs portal infrastructure and host all developed portlet applications. A potlet is developed to communicate with event hub for near real time energy information. Similarly different portlets are developed targeting different user groups, where offline information is accumulated from FAS and BAS databases through web services. Number 6 shows user groups working in manufacturing enterprise; portal owner assigns an administrator who gets the permission of assigning other users in different user groups. Finally, users in different groups see corresponding dashboards designed for them.

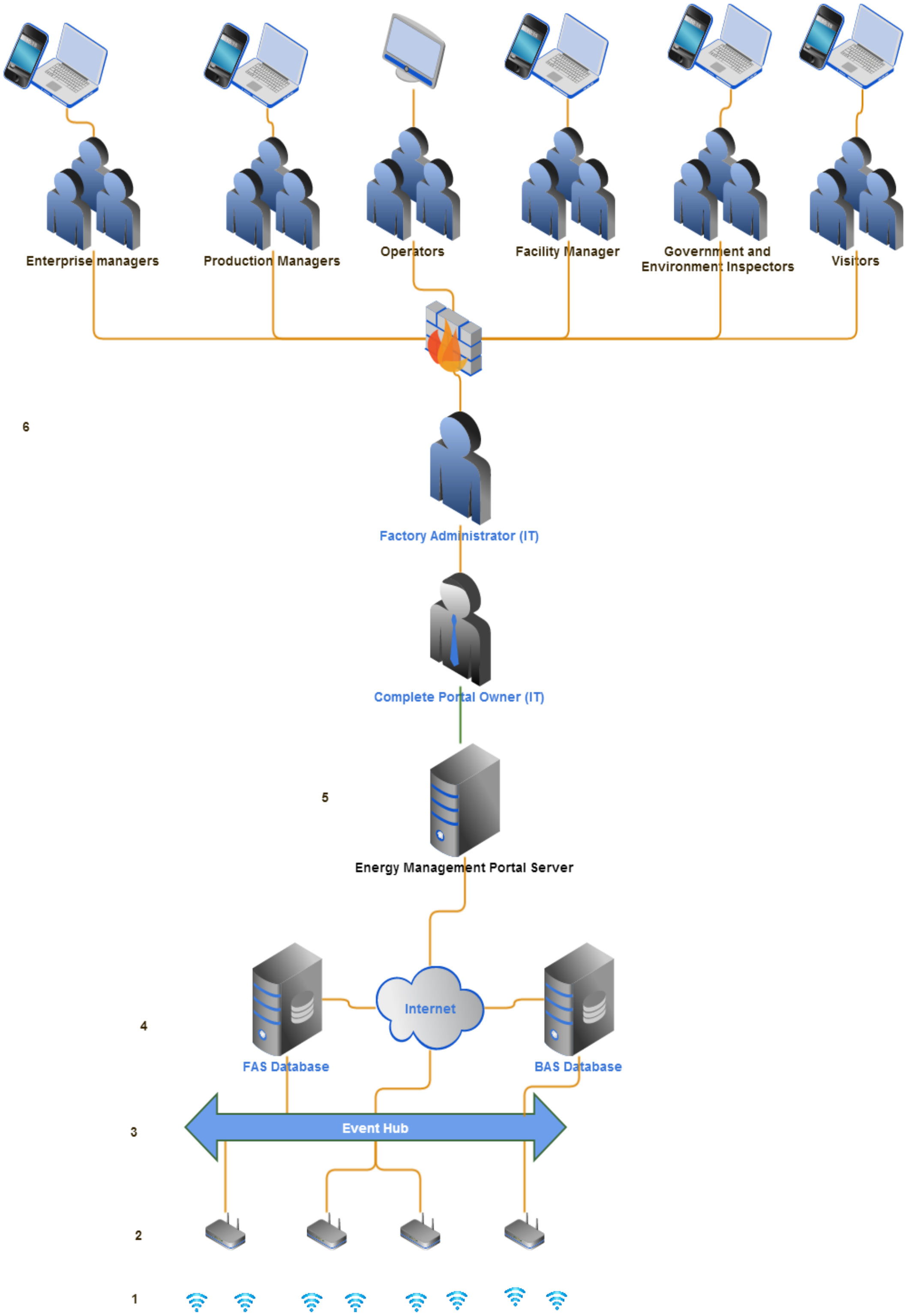

Figure 30: Generic EMS Architecture

4.2 Implementation Test-bed (Fastory)

Fastory is a setup of production line of 12 cells located at Factory Automation Systems and Technologies Laboratory (FAST-Lab.) in Tampere University of Technology.

Fastory simulates production of mobile phones by performing different assembling recipes like frame drawing, keyboard drawing and screen drawing. Layout of Fastory is presented in Figure 31.

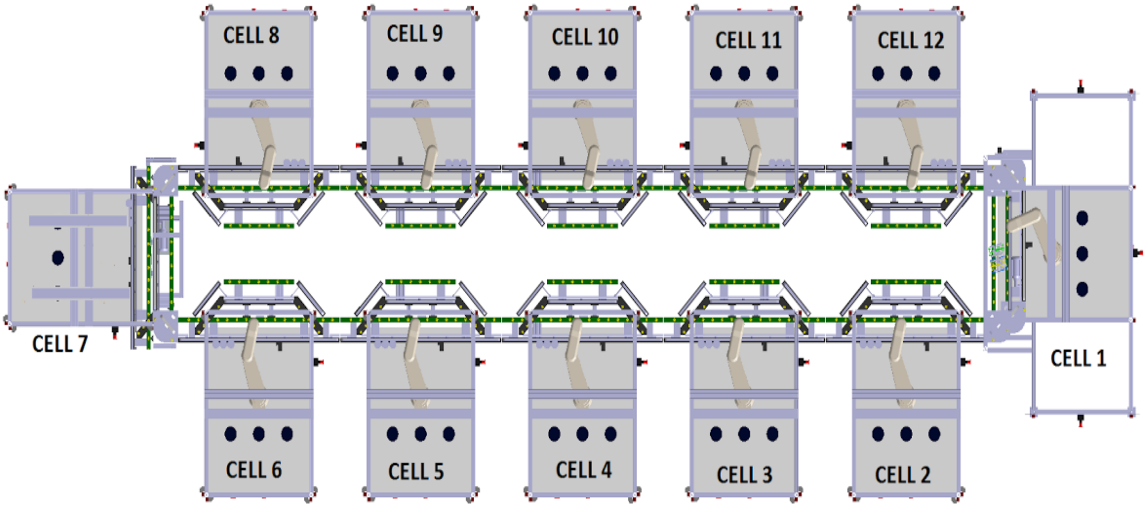

Figure 31: Fastory Layout Illustration

Each cell consists of a robot, controller and conveyors except cell no1 and 7. Cell 1 has additional machine vision system to inspect quality of product and status of product as well. If product is completed it forwards the product to completed products tray. Cell 7 works as a buffer without robots and controller. Rests of cells contribute in processes. When a cell is busy, it bypasses the product to next cell. Using such algorithm device level energy optimization was done. In order to monitor energy activity all the cells are retrofitted with S-1000 controllers and its E-10 extension.

Figure 32: S100 controllers' and E-10 module

S-1000 controller is a smart Remote Terminal Unit (RTU); able to control shop floor real-time and seamlessly publish data using WS to integrate with Enterprise applications. It also has SOAP/XML interface to communicate with industrial processes and DPWS enabled devices [66]. E-10 extension of S-1000 allows it to monitor energy consumption of robots, controllers and conveyors at real-time and publishes energy data using WS.

Fastory is a SOA enabled manufacturing site. Its FAS and BAS already have capabilities of publishing and consuming WS. A smart lighting application in FAST is considered as BAS of Fastory. Event hub is also designed to support FAS and BAS as single subscription interface to aggregate all data from S-1000 controllers. It is also integrated with EMS in this work. FAS and BAS have CEP engines that support generating designed KPIs.

4.3 Energy KPIs identification process

Existing CEP engines integrated with FAS and BAS are utilized to accumulate energy KPIs. But a set of energy KPIs needed to identify for implementation. KPIs are defined based on previous implementations and studies mentioned in theoretical background. Most of defined KPIs are generic and possible to implement in similar production setups. As this work is implemented in a simulated work setup, all the defined KPIs were not feasible to execute. Although a significant part of listed KPIs are implemented. In Appendix- C: KPIs List, all the KPIs are listed and documented according to ISO22400 KPI documentation specification.

4.4 Liferay Portal Implementation

Liferay Portal is a flexible web application from application server environments point of view. It can be installed on most popular server environments today from Tomcat to multi-node clustered commercial application servers. In this implementation Liferay is installed on Tomcat 7.

However, Liferay installation is not as straight forward as WAR file deployment. Several changes and customization in tomcat server needed to be done to make it compatible for Liferay. Liferay provides WAR file as liferay-portal-6.1.x-<date>.war and dependency jar files as liferay-portal-dependencies-6.1.x-<date>.zip in its download directory. An installation guide for Liferay on Tomcat is provided in [67].

After following the instructions, Liferay is implemented on tomcat folder with mysql 5.2 database server. Database saves all the portal information from content data to users' information and dashboard management credentials.

4.5 Implementation of Fastory as Organization

Liferay overrides home page of tomcat server. Thus, homepage of liferay portal itself become http://localhost:8080 Figure 33. An Organization is created naming Fastory from control panel of Liferay Portal. Fastory is implemented with public pages for visitors and private pages for user groups within manufacturing enterprise.

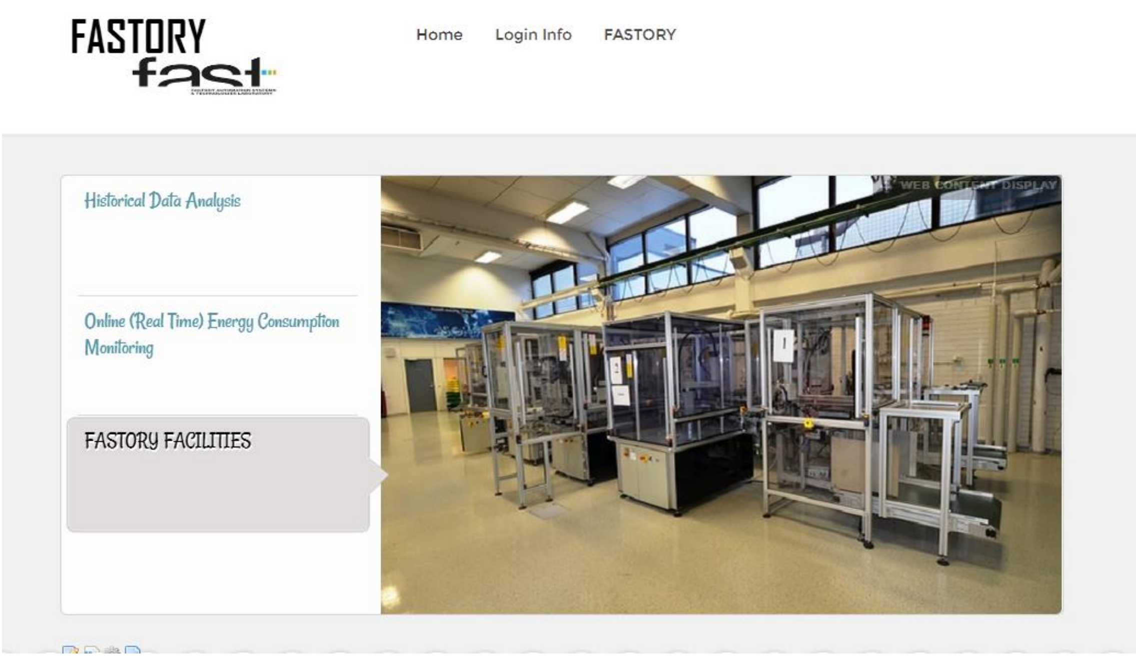

Figure 33: EMS home page

In this work, three internal user groups are eventually implemented for Fastory organization:

1. Production Manager
2. Facility Manager
3. Operators.

Homepage of private site is implemented as dashboard for these all user groups. But different set of portlets are developed targeting each user group. Therefore, dashboard presents different KPIs and energy information based on users' particular user group.

4.6 Implementation of Portlet Applications

Several porlets applications are developed in this thesis work. A generic configuration model of portlet development is already discussed in methodology chapter. In this section conceptual model, classes and architecture is presented by each implemented application.

4.6.1 Online Energy Monitoring Portlet

An online presentation of energy consumption in production line is achieved by establishing communication between event hub and portlet application. Camel framework is used to consume online energy information in an endpoint. Then DWR API streams the achieved value to client side and Highcharts API presents the information in linegraph to visualize overall trend Figure 34.

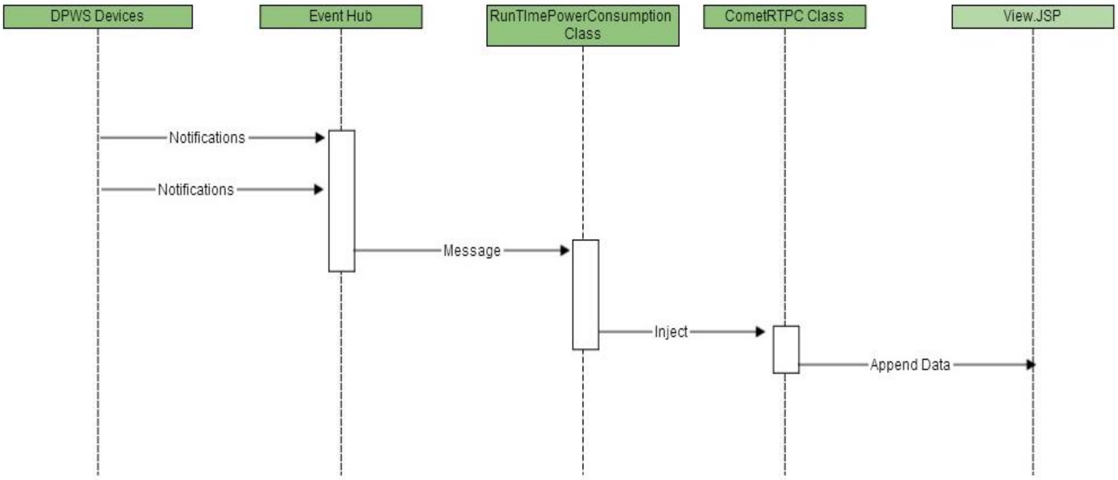

Figure 34: Online Energy Information Sequences

Event Hub receives all the energy data from devices and offers a single interface to subscribe certain information. It uses DPWS client to communicate with devices. `RunTimePowerConsumption` class extends `MVCPortlet` class. It utilizes camel framework and open an endpoint where event hub send messages in Json format. `RunTimePowerConsumption` class checks the message and forward it to `CometRTPC` class. The method sendjson(String) of `CometRTPC` class sends the json message to browser using reverse ajax technique. UML of the portlet application is presented in Figure 35.

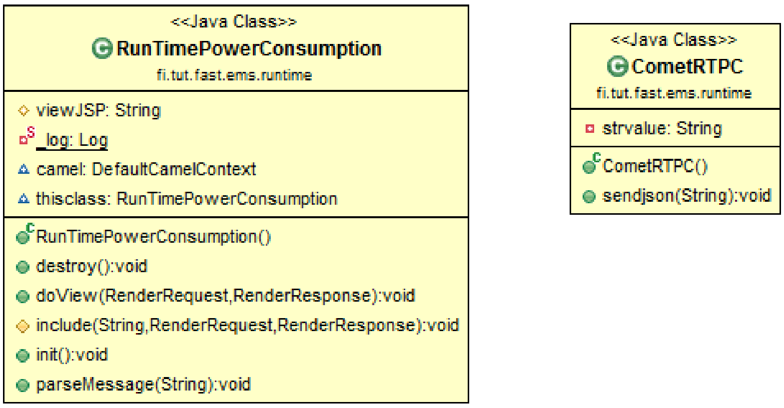

Figure 35: RunTime Power Consumption Portlet Server Side classes

In client side JSP, several JavaScript methods are implemented to parse the message, check duplicate message, append values in Highcharts graph API method and so on. In addition, a comparison window is also implemented using java script method that compares power consumption in watt (W) for two cells online. Furthermore, each cell can be monitored individually too Figure 36.

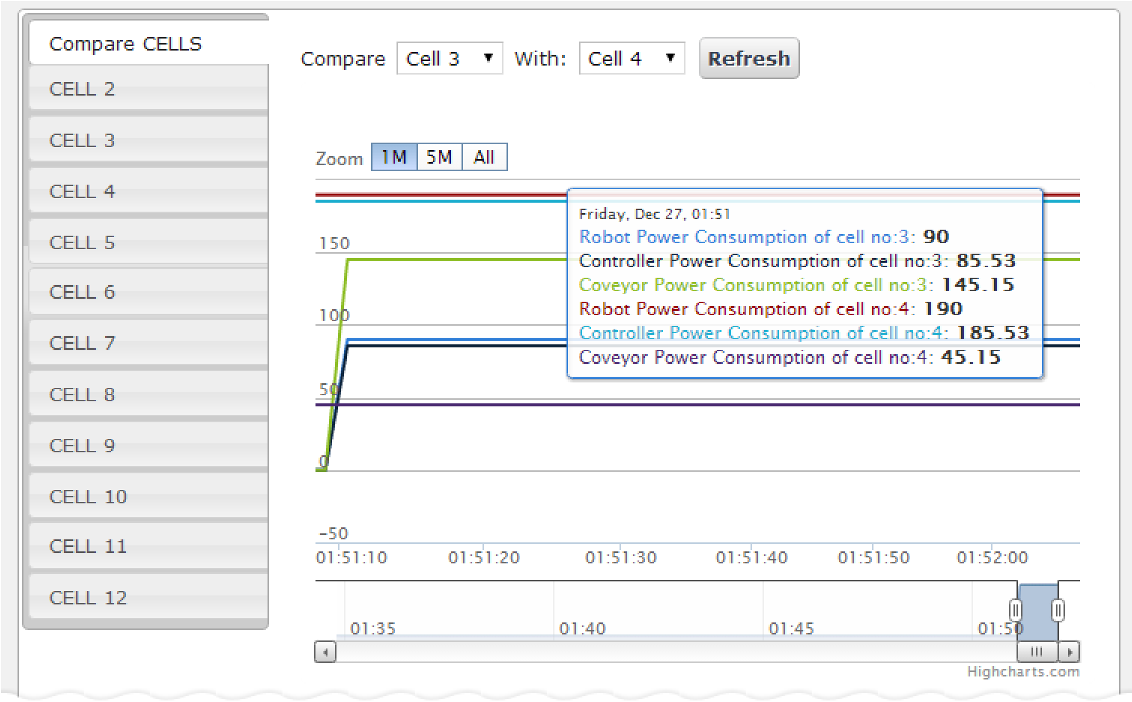

Figure 36: Online Energy Information Presentation

4.6.2 Overall Performance indicators (Activity Portlet)

Three activity portlets are developed targeting production manager, facility manager and operators. These portlets invoke web services from Data Acquisition Services (DAS) of KPIs info-store and present related information to each user group. serveResource method of portlets receive ajax request from client side and send Boolean request to DAS using instance of `SingleKpiRequest` class and get response.

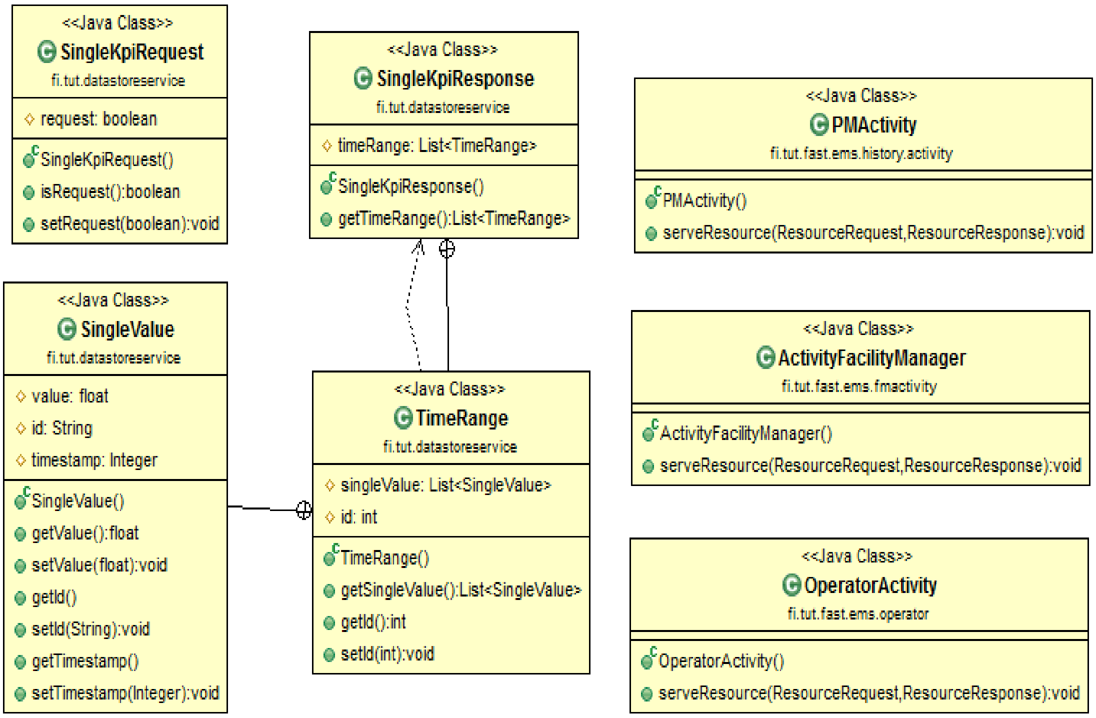

Figure 37: Activity portlets UML

Then based on user type, portlet sends particular response to client side. For example, "production manager activity portlet" responses with energy information within production line scope. In contrast, "facility manager activity portlet" responses energy information as summation of production line energy data and lighting system energy data. UML of activity portlets presented in Figure 37.

In client side JSP, energy information of activity portlets are presented as numerical values, presented in Figure 38.

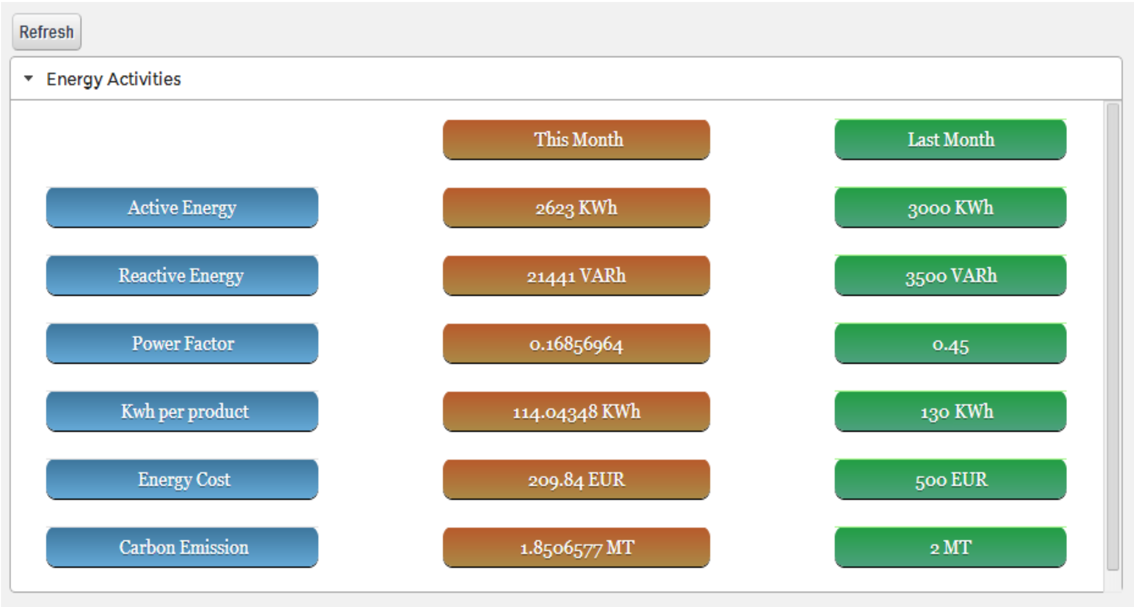

Figure 38: Activity portlet for Production Manager

4.6.3 Implementation of production line historical data acquisition

This portlet shows historical energy information of production line to production manager. It sends Ajax request from client side with time range, cell no and specific parameter like: KWH, VAR, Irms and so on. Then in server side, portlet communicates with historical KPIs service enabled from FAS and get response. Figure 39 presents UML of request parameters for historical energy information. `HistoricalDataRequest` class consists of request parameters. `CellsList` class contains number of cells in list of integer. `TimeRange` class contains start time and end time of query.

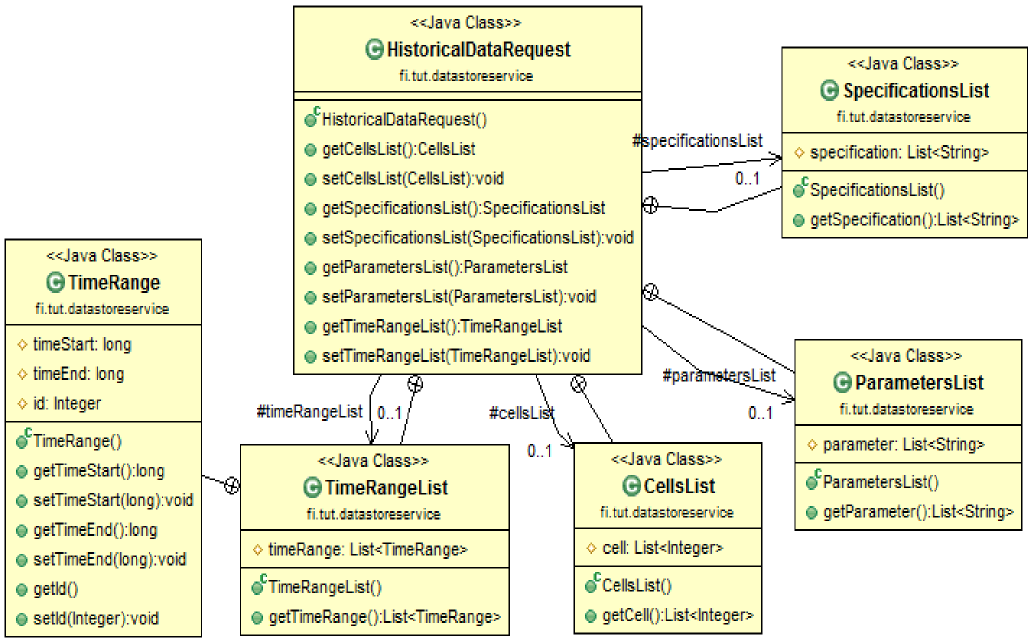

Figure 39: Service request classes of Historical energy KPIs

Figure 40 presents classes for handling response from historical energy data request. `Cell` class contains id of cell in integer and `ValueList`, where `ValueList` contains pair of value of specific parameter and timestamp.

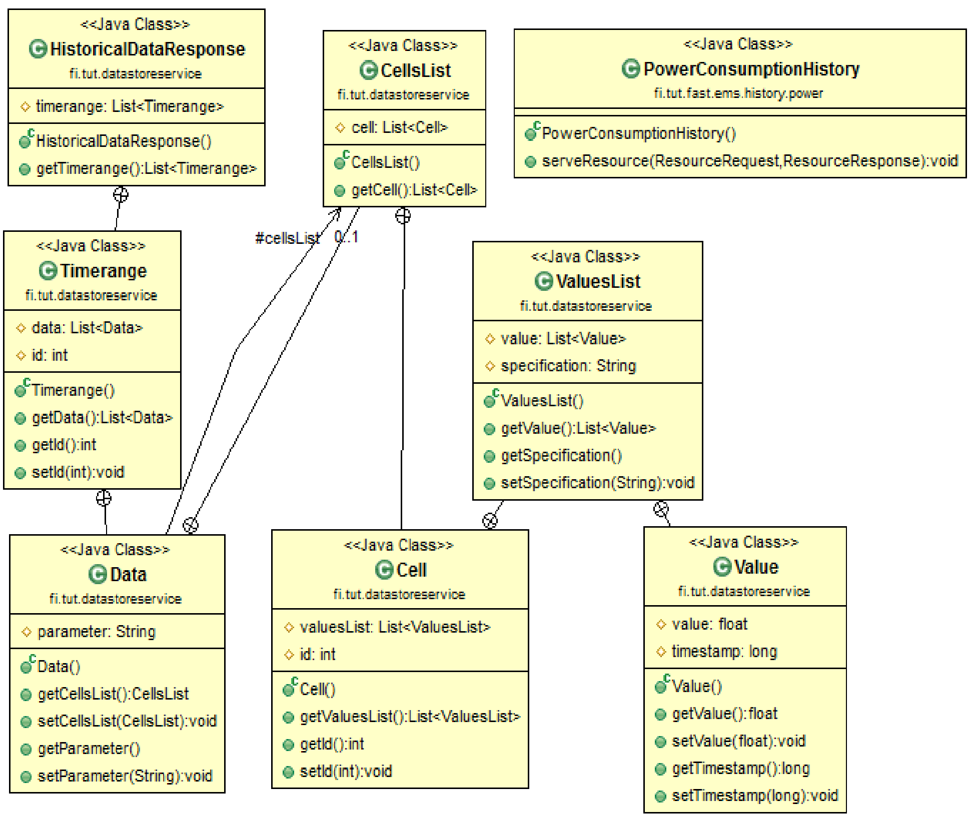

Figure 40: Service Response classes of Historical energy KPIs

In client side view.JSP, timestamp input boxes implements JqueryUI date picker library for click and input option. User can select cell number and monitoring parameters like Active Energy, Active Power, Reactive Energy, Reactive Power, Apparent Energy, Apparent Power, RMS Current and RMS Voltage. Figure 41 presents RMS voltage of cell no 3 for selected time range.

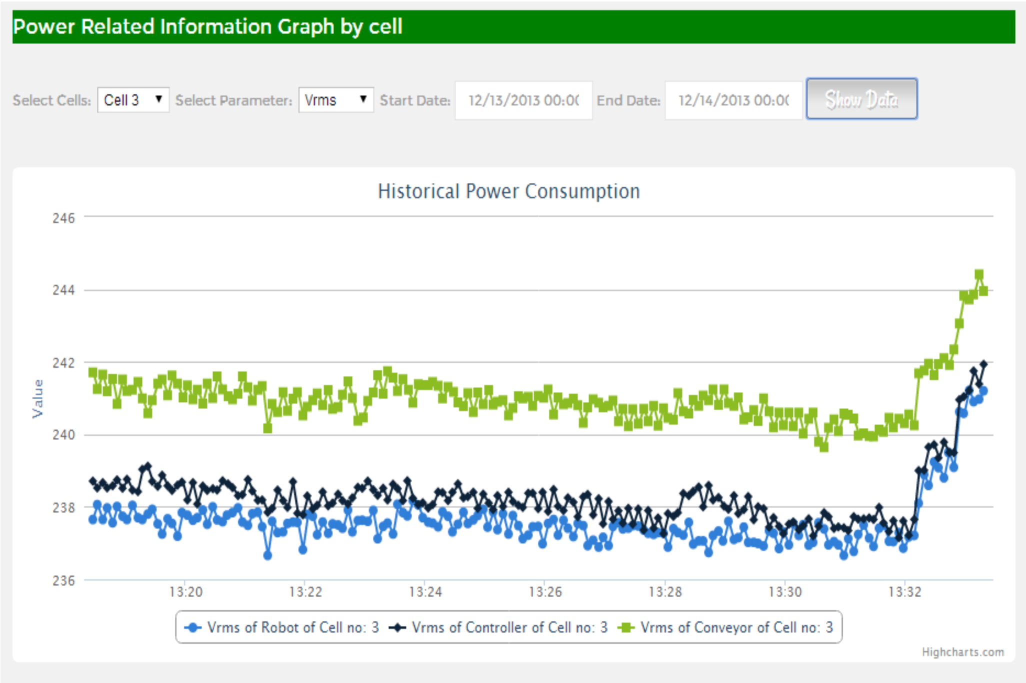

Figure 41: Historical energy trend monitoring Portlet

4.6.4 Implementation of Comparison Portlets

Two comparison portlets are developed targeting production manager and facility manager. These portlets use a manual composition of services and get response from historical KPIs mentioned in previous section and Production KPIs using JAX-WS API. Then it shows trends of energy information with production information.

Figure 42 presents classes responsible for sending request and getting response about production KPIs. `ProdustionKpiRequest` and `ProductionKpiResponse` classes are consequently responsible for sending request and receiving response. `ProdustionKpiValue` class contains values of specific KPI and its cell no. Besides `InterRelateKPI` class extends `MVCportlet` class and handles requests and responses from client side JSP.

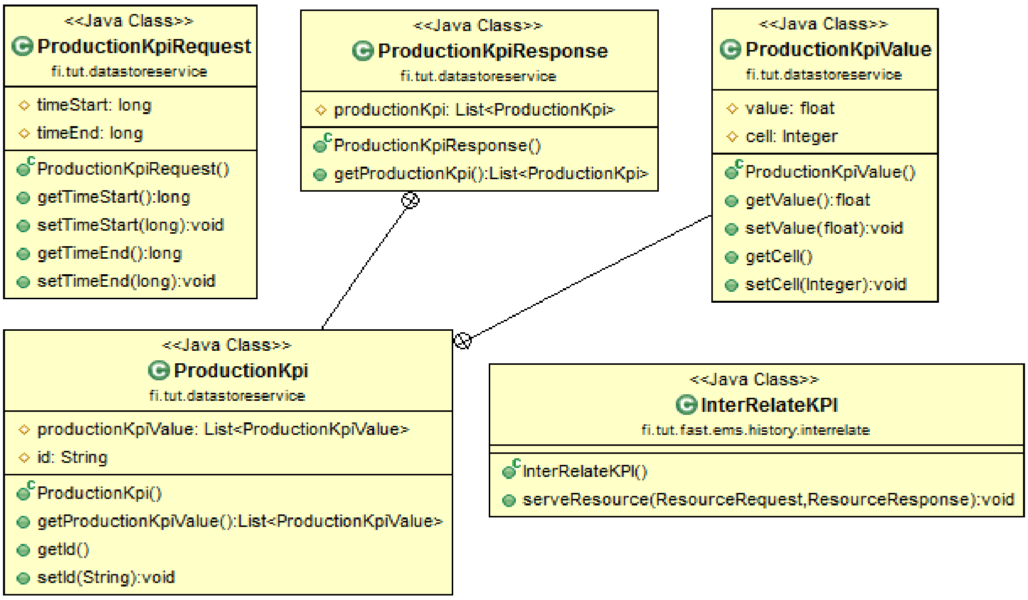

Figure 42: UML class diagram of Comparison Portlets

Figure 43 shows presentation window of comparison portlet. User can select time range and update graph by clicking refresh data.

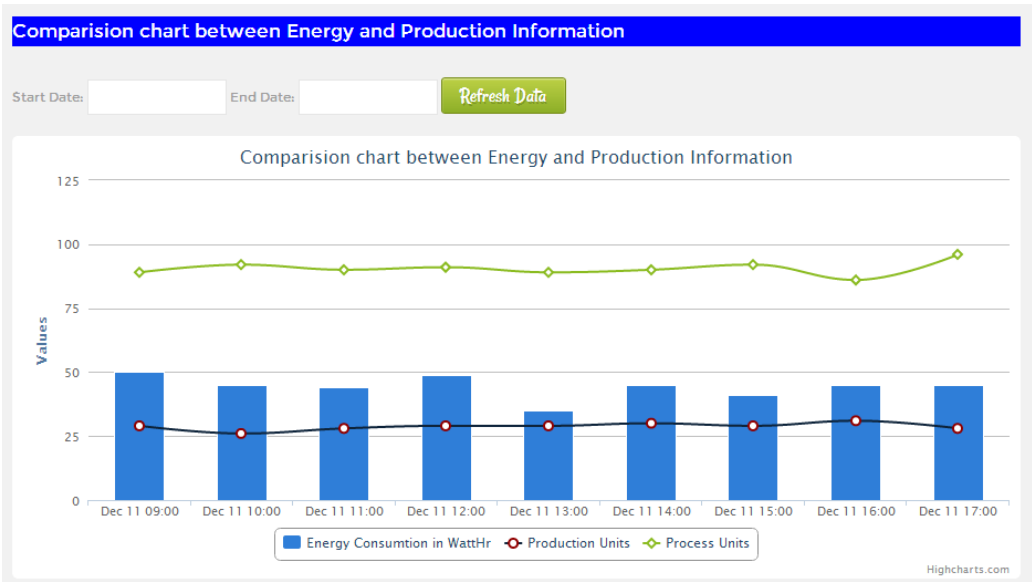

Figure 43: Comparision Portlet window for Production Manager

On the other hand, portlet for facilities manager presents only energy consumption information Figure 44. But this porlet combines energy consumption of BAS and FAS and displays overall consumption data.

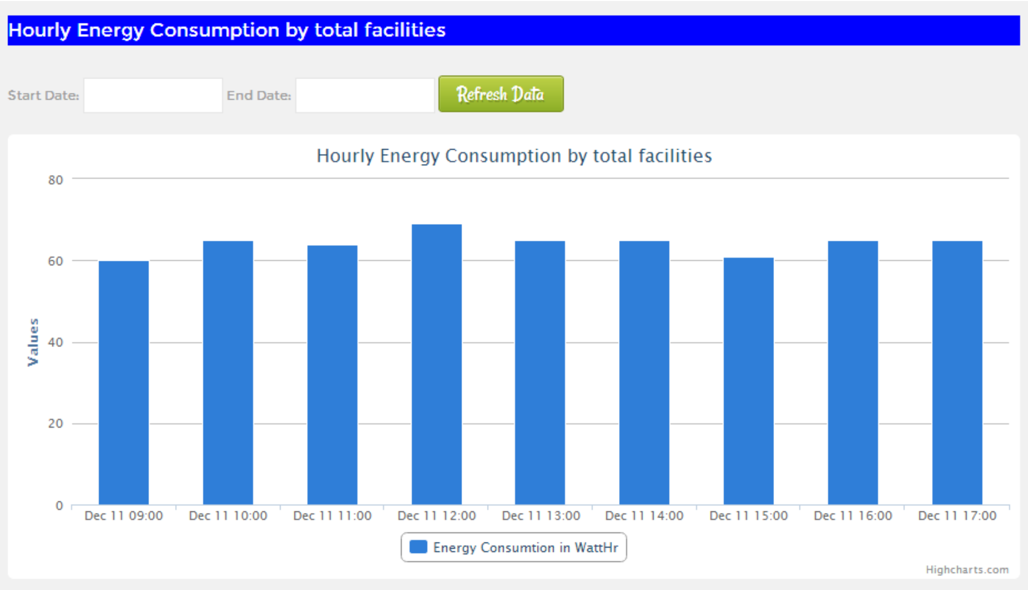

Figure 44: Portlet Window of Facilities Manager

4.6.5 Energy performance comparison portlet

This portlet compares historical power consumption of two cells and present it to production manager. It uses similar class diagram of section 4.6.3 but with an extra method in client side to accumulate data of two cells. This comparison supports production manager to identify maintenance requirement of particular cell or modification in production scheduling. Figure 45 compares power consumption between cell no 4 and 5 for selected time range. Power consumption of each cell contains power consumption information of its robot, conveyor and controller.

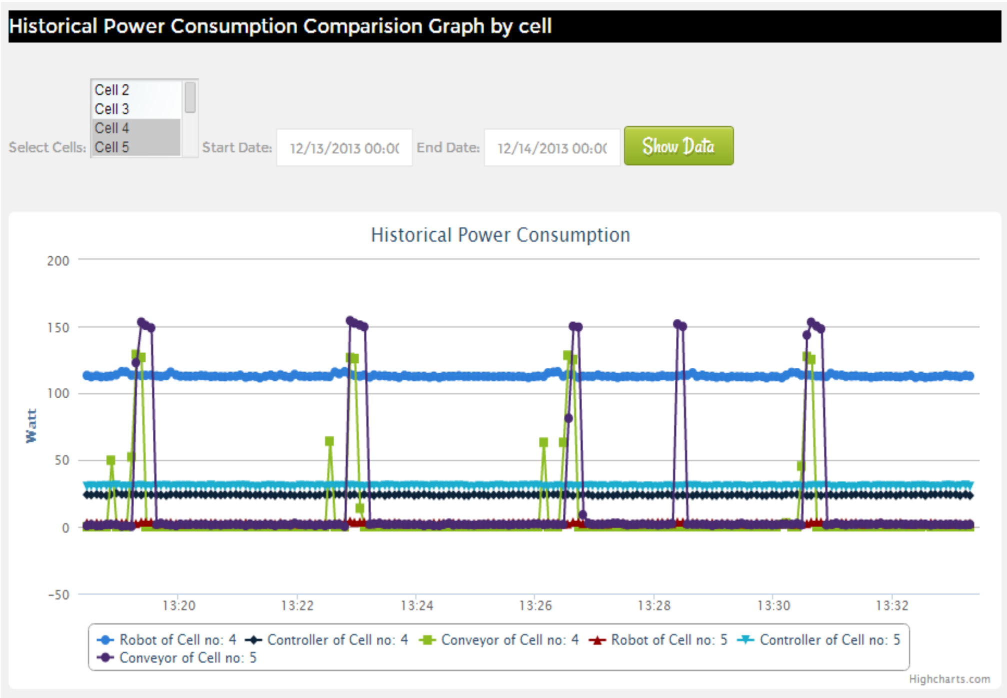

Figure 45: Power Consumption comparison portlet dispaly window

4.6.6 CO2 monitoring portlets

Two CO2 monitoring portlets are developed for production manager and facility manager. These portlets also invoke DAS to get daily energy consumption and then calculates CO2 emission. Request parameters of these portlets are start time and end time then response information are presented in volume graph. Figure 46 presents daily CO2 emission in Kg within production line scope.

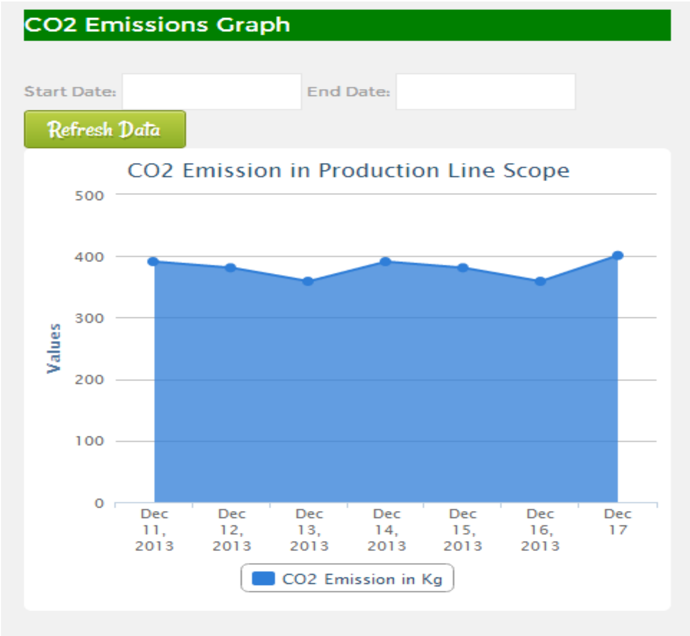

Figure 46: CO2 emission monitoring portlet for Production Manager

4.7 Implementation of Dashboard

Private site homepage of created Fastory organization is designed as dashboard of EMS. Portlets are placed in the page by simple drag and drop method from portlets add menu. Figure 47 presents all the developed portlet in add menu.

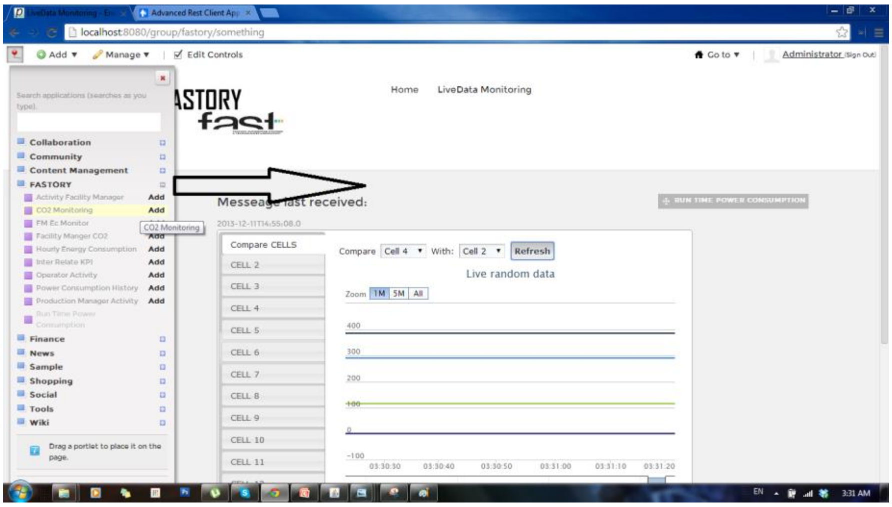

Figure 47: Add Portlet menu

After the portlet is placed on the page, when admin hovers on the portlet, an "Options" icon appears on the top-right corner of the portlet Figure 48.

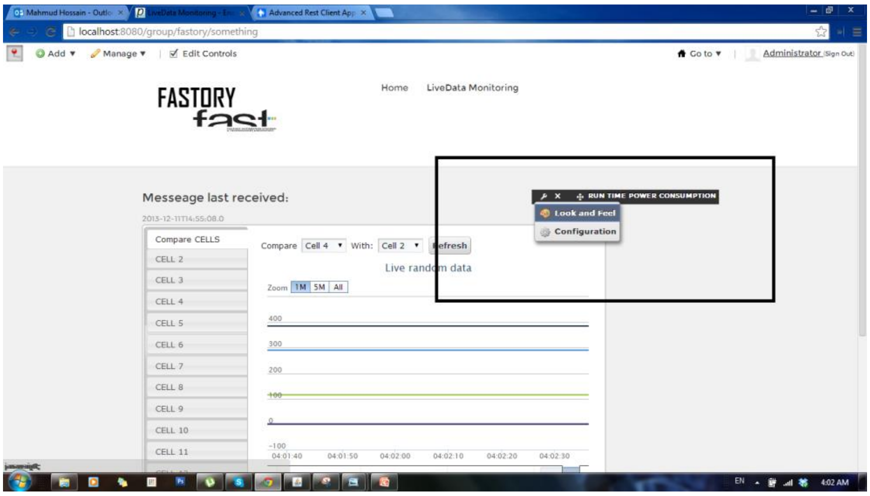

Figure 48: Portlet Configuration Option

In configuration option of the portlet, admin can set user credentials and rights for the particular portlet Figure 49.

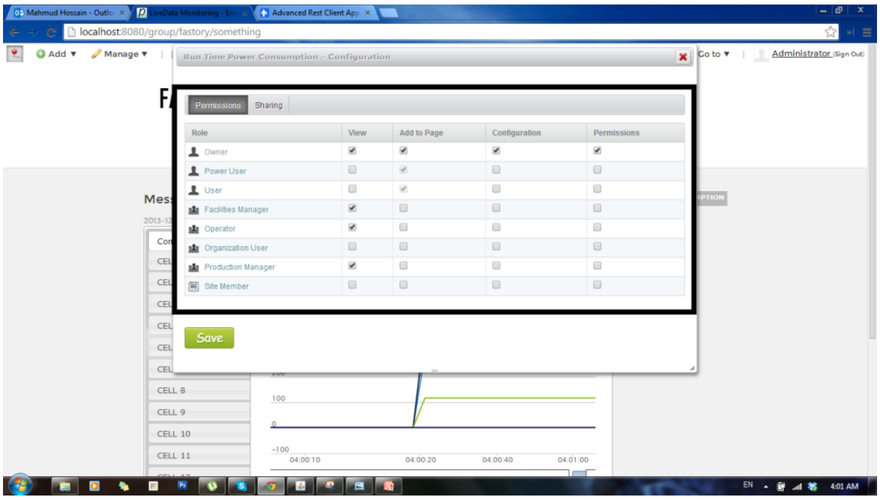

Figure 49: Permission Configuration of Portlets

Apart from that, there is "Plugins Configurations" option in the "Control Panel" can also be used to centrally enable and disable certain portlet for specific user group.

Once the dashboard is implemented and configured, administrator of Fastory organization assigns registered users in different user role. Then user of particular role sees corresponding energy management dashboard for his role.

5. RESULTS

This chapter presents results of implementation in test bed described in section 4.2 (Fastory). Fastory simulates cell phone production using 12 cells. Therefore, depicted results are conceptual within this study frame work, and a proof of concept for any manufacturing site. Implementation of portal architecture enabled following outcomes in user interface layer of EMS:

1. Dashboard management through developed portlet applications
2. User role assignment and management
3. Organization management.

The test is conducted on three user groups; Production Managers, Facilities Managers and Operators.

5.1 Dashboard for Production Manager

Considering the responsibilities of production manager following table shows portlets deployed in production managers dashboard of EMS.

Table 5: Portlets in Production Manager Dashboard

No	Name	Description
1	Activity Portlet	Presents numerical value of selected KPIs to show overall activity within production line scope.
2	Historical Power Comparison portlet of cells	Compare power load of cells to understand performance of different cells.
3	Interrelate Portlet	Interrelate energy KPIs with production KPIs accumulated from MES to depict energy performance and its relation with production performance.
4	Power related information by cells	This portlet present entire investigation of single cell by presenting trend of its all energy related information such as Active Energy, Active Power, Reactive Energy, Reactive Power, Apparent Energy, Apparent Power, RMS Current and RMS Voltage.
5	CO2 emission monitoring portlet	This portlet shows daily CO2 emission trend in production line scope.
6	Online energy consumption by cell	Monitor and compare near real-time energy consumption activities by cells.

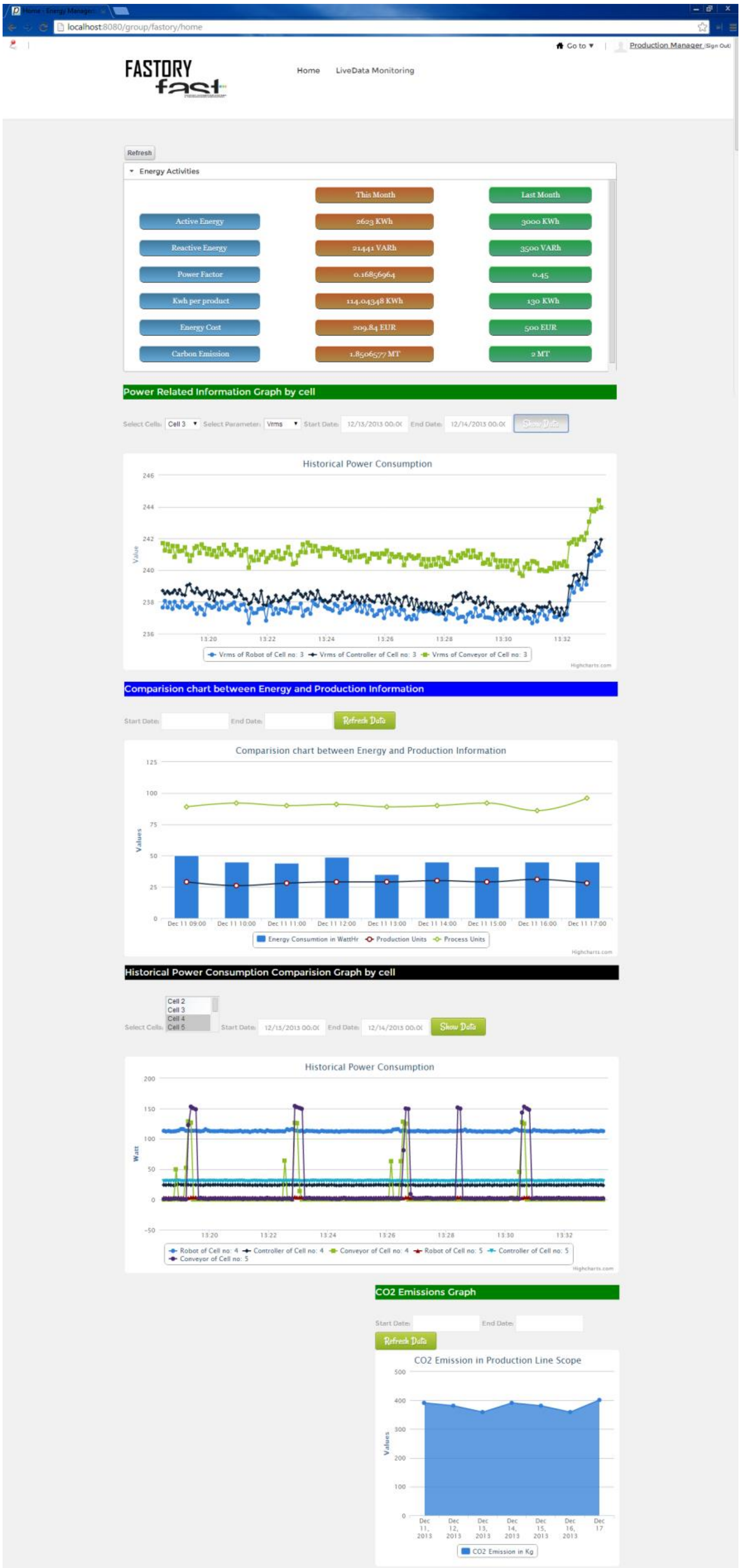

Figure 50: Dashboard Production Manager

Figure 50 presents complete dashboard page designed for production manager of Fastory. First portlet in the dashboard is activities portlet presenting numerical values of selected energy KPIs within production line scope. Historical Power Consumption portlet describing power related parameters of specific cell. Third portlet is interrelate portlet presenting interrelation between power KPIs and production KPIs. Fourth portlet compares power consumption of different cells. And daily CO2 emission is presented at the bottom of the dashboard.

5.2 Dashboard for Operators

Operator is the person who stays near production line or factory and change values or act immediately, when production line needs external support from human.

Tasks of an operator in FASTORY:

- Remove Pallet when it is stuck/block.
- Turn off /request maintenance of a cell (Robot, Conveyor, Controller) if it is not performing accordingly or consuming more than set energy.

Considering role of operators following table lists portlets deployed in EMS dashboard of operator.

Table 6: Portlets in EMS Dashboard for Operator

No	Name	Description
1	Activity Portlet for oprator	Presents numerical value of selected KPIs to operator describing overall activity within production line scope.
2	Online energy consumption by cell	Monitor and compare near real-time energy consumption activities by cells.

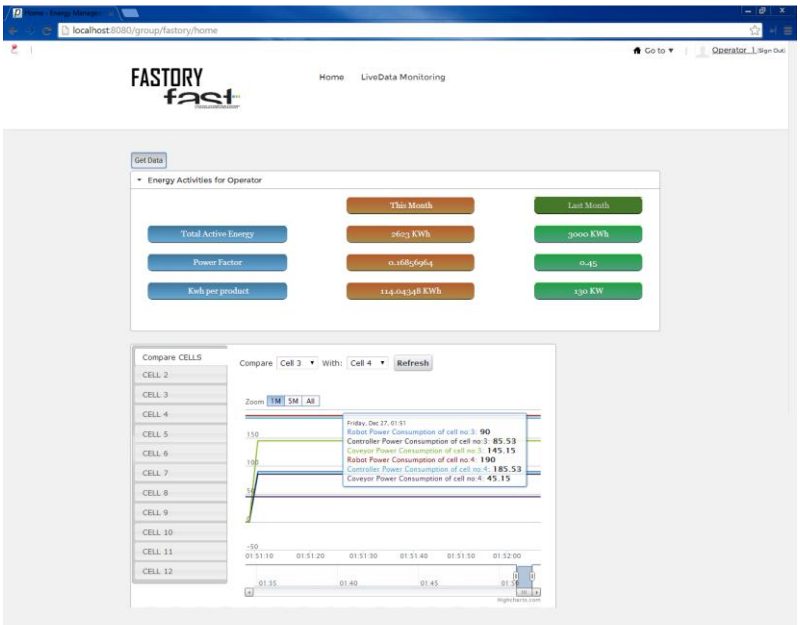

Figure 51: EMS Dashboard for Operator

Figure 51 presents dashboard page developed for operators of Fastory. Dashboard contains one activity portlet and online energy monitoring portlet described in Table 6.

5.3 Dashboard for Facilities Manager

Facilities manager is responsible for building management, maintenance, contracts and operational costs. Thus, energy data aggregation for facilities manager dashboard includes production line, HVAC and lighting system as well.

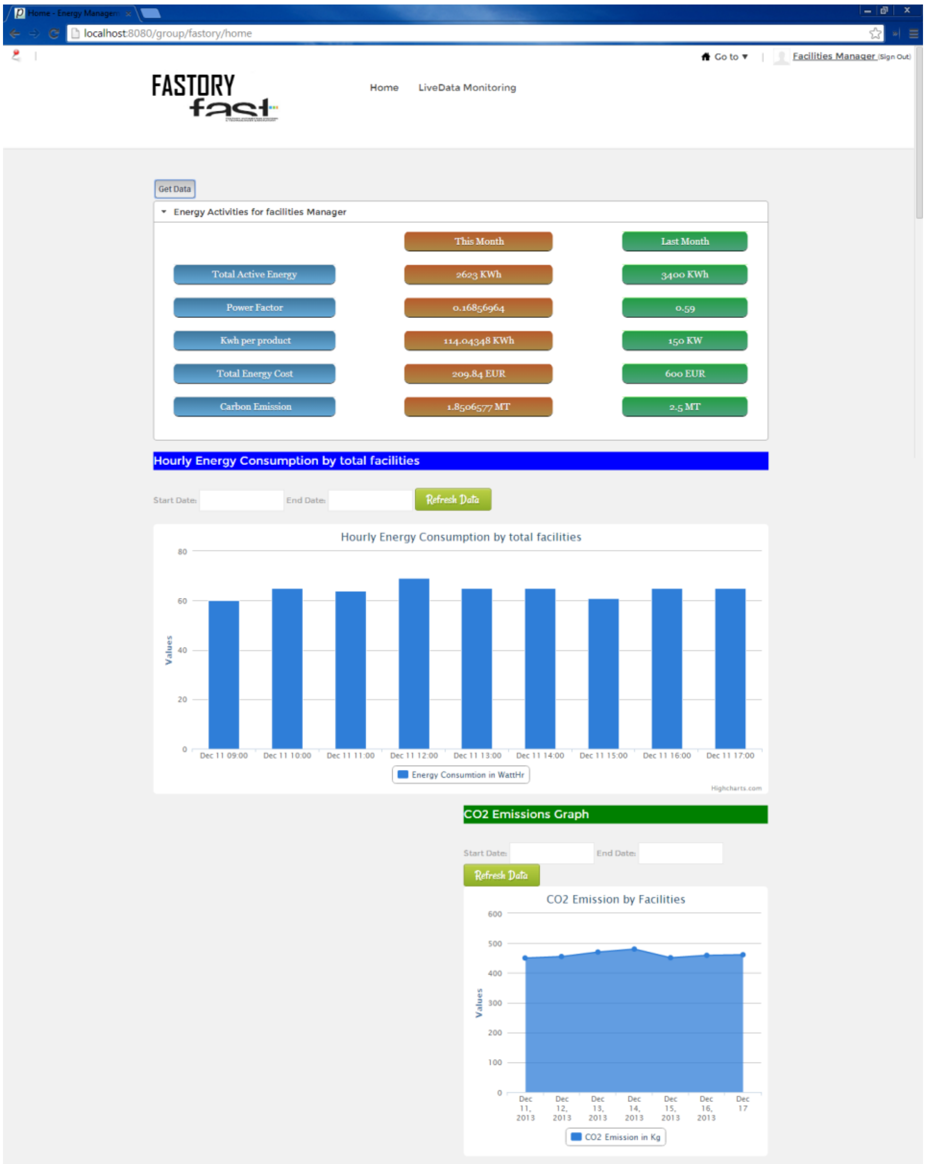

Figure 52: Dashboard for facilities manager

Figure 52 presents dashboard of facilities manager. List of portlets in dashboard of Facilities manager is presented in following table.

Table 7: Portlets in EMS Dashboard for Facilities Manager

No	Name	Description
1	Activity Portlet for facilities manager	Presents numerical value of selected KPIs to show overall activity within all facilities.
2	Hourly/Daily Energy Consumption by overall facilities	Presents hourly or daily energy consumption by all the equipments in facility including production line, lighting, utilities, and HVAC system.
3	CO2 emission monitoring portlet	Shows daily CO2 emission trend in overall facilities scope.

Each of this presentation unit is implemented by portlets developed in implementation phase. Customization of dashboards is done at flexible and configurable UI level like; right to view particular portlet, displaying portlet in dashboard, removing portlet from dashboard and user management.

6. CONCLUSION AND FUTURE WORK

This chapter concludes the thesis work conferring importance of achieved results and introduced approaches for the implementation energy management system. In addition, three different directions of further works are also presented to continue research on energy management solutions.

6.1 Conclusion

The main technical challenges of energy management solutions are: integrating energy data from heterogeneous sources and visualizing energy information based on near real-time and historical energy KPIs to different user groups according to their role in manufacturing facilities.

In this thesis, EMS is implemented using Liferay portal infrastructure. Organization management feature of Liferay is utilized to enable hierarchical user management and dashboards management for corresponding user groups.

The introduced approach is based on SOA paradigm which allowed EMS to communicate with FAS and BAS through web services. Seamless communication between these services realized a holistic portlet-based presentation layer for EMS. Successful Interoperability between these systems provided a true monitoring tool for energy information in manufacturing facility.

The objective of online energy data presentation is achieved by implementing middleware between EMS and heterogeneous energy concern components in manufacturing site. Utilization of middleware reduced the heterogeneity by unveiling on single subscription point. Middleware also ensured reliability of data flow between applications and devices. Online energy information is presented for each process cell with its major components like robot, controller and conveyor. Energy performances of different cells are also compared online to aware concerning user parties within the manufacturing facilities.

Energy KPIs are baseline of visualizing energy information of EMS. A set of KPIs is identified and documented concerning energy efficiency and performance within manufacturing facility. Designed KPIs are provided to KPI info store. Then communication between KPI info store and EMS is established through web services. KPI values are presented through charting library to understand the trend at a glance. Some crucial KPIs are also shown in numerical values to depict overall energy activity of manufacturing site. Production KPIs from MES and energy KPIs from energy KPI info store are interrelated to comprehend the relation between these activities and understand complete energy performance of production line.

In order to adapt with continuous modification and upgrade in modern manufacturing facilities, EMS needs to be flexible and configurable too. Successful achievement of this work is enabling flexible configurability of EMS in user interface level. Reconfigurability of dashboard or any security settings are enabled in UI level through Liferay architecture. A responsive theme of Liferay is adapted to keep the dashboards platform independent and accessible from any device like mobile, tablet, laptops or any device with an internet browser supporting HTML5.

From the achieved results and hierarchical dashboards, it can be conclude that the main objectives of the thesis work are fulfilled with ISO:50001 standardization. The value of this work can be summarized as a novel architecture for energy management system. The implemented data aggregation techniques and presentation will improve energy awareness among user groups in factory to drive it towards energy efficient smart manufacturing factory.

6.2 Future Work

The thesis work attempts to design and implement a EMS in portlet based portal infrastructure. Due to the confines in test bed few KPIs are visualized and implemented, so rest of KPIs need to be implemented and see the results as well. Online information flow and KPI generation results should be tested for longer period to realize the actual change in manufacturing decisions. Furthermore, manual controls can be introduced in EMS for controlling devices like lights, HVAC and other utilities. Implemented EMS only focuses on electric energy in manufacturing. However, other energy sources like; gas, crude oil and heat can be monitored in similar approach. Besides service compositions are done manually in this work. Thus in future work, a standard portlet can be developed with Business Process Model and Notation (BPMN) service composition capability.

Although work has been implemented in a manufacturing facility, the successful results of implementation revealed a huge diversified potential of the developed architecture.

EMS for manufacturing enterprise in multiple locations

One of the possibilities to implement the architecture is in manufacturing enterprise with multiple locations in different countries. As it is possible to implement multiple organizations with child and parent organizations, information flow between different manufacturing sites can be integrated and defined within defined portal infrastructure.

EMS for commercial Building

A similar attempt can be taken for monitoring energy activities in large commercial buildings, where each company is configured as organization within EMS architecture.

EMS for Smart Cities

The most significant potential of this architecture is to implement it for an entire city. With help of service oriented architecture, historical and onsite data of these individual energy management systems can be centralized in one portal to monitor total energy consumption of a city or area. Introducing such portal would help energy providers to rearrange energy distribution, analyze energy consumption from macro level to micro or vice versa. On the other hands, Individuals and Factories would also aware about source of their energy costs, monitor and take decisions. They would also receive offers and advice from the energy providers to reduce energy cost.

REFERENCES

[1] World Energy Outlook 2012 (Released on 12 November 2012) (Page 2) [Accessed Online on 10th January, 2014] Available: http://www.worldenergyoutlook.org/publications/weo-2012/

[2] Energy-smart Europe on the horizon - 09/03/2011 [Accessed Online on 26th December, 2013] Available: http://ec.europa.eu/news/energy/110309_1_en.htm

[3] Consumption of energy [Accessed Online on 26th December, 2013] Available: http://epp.eurostat.ec.europa.eu/statistics_explained/index.php/Consumption_of_energy

[4] Use of Energy in USA [Accessed Online on 26th December, 2013] Available: http://www.eia.gov/energyexplained/index.cfm?page=us_energy_use

[5] ISO 50001:2011(en) Energy management systems — Requirements with guidance for use [Accessed Online on 10th January, 2014] Available: https://www.iso.org/obp/ui/#iso:std:iso:50001:ed-1:v1:en

[6] A. Cannata, M. Gerosa, M. Taisch "SOCRADES: a Framework for Developing Intelligent Systems in Manufacturing" Industrial Engineering and Engineering Management, 2008. IEEM 2008. IEEE International Conference on 8-11 Dec 2008, page(s):1904 – 1908.

[7] Anna Florea, Jorge A. Garcia Izaguirre Montemayor, Corina Postelnicu, Jose L. Martinez Lastra "A cross-layer approach to energy management in manufacturing" Industrial Informatics (INDIN), 2012 10th IEEE International Conference on Date of Conference: 25-27 July 2012 Page(s): 304 – 308

[8] Pramudianto, F.;Simon, J. ; Eisenhauer, M. ; Khaleel, H. "Prototyping the Internet of Things for the future factory using a SOA-based middleware and reliable WSNs". Emerging Technologies & Factory Automation (ETFA), 2013 IEEE 18th Conference on Date of Conference: 10-13 Sept. 2013 Page(s): 1 – 4

[9] Endo, M.; Nakajima, H. ; Hata, Y."Simplified Factory Energy Management System based on operational condition estimation by sensor data". Automation Science and Engineering (CASE), 2012 IEEE International Conference on 20-24 Aug. 2012 Page(s): 14 – 19

[10] Curry, E.; Hasan, S.; O'Riain, S., "Enterprise energy management using a linked dataspace for Energy Intelligence," Sustainable Internet and ICT for Sustainability (SustainIT), 2012 , vol., no., pp.1,6, 4-5 Oct. 2012

[11] Bin Zhang, Corina Postelnicu, Jose L.Martinez Lastra "Key Performance Indicators for energy efficient asset management in a factory automation testbed" Industrial Informatics (INDIN), 2012 10th IEEE International Conference on 25-27 July 2012 Page(s): 391 – 396

[12] Dancheng Li,Wei Zhang, Portal-based Design for SaaS System Presentation Layer Configurability, Computer Science & Education (ICCSE), 2011 6th International Conference on 3-5 Aug. 2011 PP: 1327 - 1330

[13] Oliver Gmelch, Gunther Pernul "A Generic Architecture for User-Centric Portlet Integration" Commerce and Enterprise Computing (CEC), 2012 IEEE 14th International Conference on 9-11 Sept. 2012, Page(s): 70 – 77

[14] Yao, P., Wen, H.-Y. , Dai, "Study on presentation layer structure of multi-tenant e-business systems" 2011 IEEE 18th International Conference on Industrial Engineering and Engineering Management, IE and EM 2011; Changchun; China; 3 September 2011 through 5 September 2011, Issue PART 1, 2011, Article number 6035174, Pages 352-354

[15] Heilala J., Klobut K., Salonen T., Siltanen P., Ruusu R., Armijo A., Sorli, M., Urosevic L., Reimer P., Fatur T., Gantar Z., Jung A., "Ambient Intelligence based monitoring and energy efficiency optimisation system," 2011 IEEE International Symposium on Assembly and Manufacturing (ISAM) , vol., no., pp.1-6, 25-27 May 2011

[16] McKane A., Perry W., Aixian L., Tienan L., Williams R., "Creating a Standards Framework for Sustainable Industrial Energy Efficiency", Energy Efficiency in Motor Driven Systems EEMODS 2005, September 2005, Heidelberg, Germany.

[17] Technical Guideline EN 16001:2009 , Sustainable Energy Ireland [Accessed Online on 30th March, 2014] Available: http://www.seai.ie/Your_Business/Energy_Agreements/IS393_Energy_Management_System/EN16001_Technical_Guideline.pdf

[18] ISO 9000 - Quality management [Accessed Online on 30th March, 2014] Available: http://www.iso.org/iso/home/standards/management-standards/iso_9000.htm

[19] ISO 14000 - Environmental management [Accessed Online on 30th March, 2014] Available: http://www.iso.org/iso/home/standards/management-standards/iso14000.htm

[20] Image source PDCA cycle [Accessed Online on 15th January, 2014] Available: http://upload.wikimedia.org/wikipedia/commons/a/a8/PDCA_Process.png

[21] David Parmenter, Key Performance Indicators (KPI): Developing, Implementing, and Using Wining KPIs, 2nd ed.: John Wiley & Sons, Inc, 2010.

[22] ISO 22400-2:2014(en) Automation systems and integration — Key performance indicators (KPIs) for manufacturing operations management — Part 2: Definitions and descriptions [Accessed Online on 10th February, 2014] Available: https://www.iso.org/obp/ui/#iso:std:iso:22400:-2:ed-1:v1:en

[23] ISO/DIS 22400-1 Automation systems and integration -- Key performance indicators (KPIs) for manufacturing operations management Part 1: Overview, concepts and terminology

[24] Red Lion, "Seven Common KPIs for Production Monitoring, Using Virtual Man-agement to Drive Productivity"; White paper, 7 pages.

[25] Zhang, Bin, "Design and Implementation of an InfoStore for Key Performance Indicators" Publication type: Diplomityö Issue date: 2012-11-07 University: Tampere University of Technology

[26] J.Heilala, K.Klobut, T.Salonen, P.Järvinen, J.Shemeikka "Energy Use Parameters for Energy Efficiency Enhancement in Discrete Manufacturing Process" Intelligent Computation in Manufacturing Engineering - 7 Accessed Online on 15th January, 2014] Available: http://www.amimoses.eu/fileadmin/templates/amimoses/files/Copy%20of%20Heilala_et_al.pdf

[27] Amrina, E, Yusof, S.M. "Key performance indicators for sustainable manufacturing evaluation in automotive companies". Published in: Industrial Engineering and Engineering Management (IEEM), 2011 IEEE International Conference on 6-9 Dec. 2011

[28] Hussain, S., Gabbar, H.A. ; Musharavati, F. ; Pokharel, S., "Key performance indicators (KPIs) for evaluation of energy conservation in buildings, Published in: Smart Energy Grid Engineering (SEGE), 2013 IEEE International Conference on 28-30 Aug. 2013

[29] General specifications and key performance indicators 2012. [Accessed Online on 20th January 2014]. Available: http://www.itu.int/dms_pub/itu-t/oth/4B/04/T4B0400000B0009PDFE.pdf

[30] Service Oriented Architecture [Accessed Online on 14th February, 2014] Available: http://www.w3.org/TR/ws-arch/#service_oriented_architecture

[31] White, L. ; Wilde, N. ; Reichherzer, T. ; El-Sheikh, E. "Understanding Interoperable Systems: Challenges for the Maintenance of SOA Applications" System Science (HICSS), 2012 45th Hawaii International Conference on Date of Conference: 4-7 Jan. 2012 Page(s): 2199 – 2206

[32] Chehili, H.a , Seinturier, L.b , Boufaida, M. "FASOAD: A framework for agile service-oriented architectures development" 24th International Workshop on Database and Expert Systems Applications, DEXA 2013; Prague; Czech Republic; 26 -29 August 2013, Article number 6621375, Pages 222-226

[33] Baoan Li, Research and Application of SOA Standards in the Integration on Web Services Published in: Education Technology and Computer Science (ETCS), 2010 Second International Workshop on 6-7 March 2010

[34] Web Services Architecture [Accessed Online on 17th January, 2014] Available: http://www.w3.org/TR/2003/WD-ws-arch-20030514/

[35] Zeeb, E., Bobek, A. ; Bohn, H. ; Golatowski, F. "Service-Oriented Architectures for Embedded Systems Using Devices Profile for Web Services". Published in: Advanced Information Networking and Applications Workshops,

2007, AINAW '07. 21st International Conference on (Volume:1)Date of Conference: 21-23 May 2007

[36] H. Bohn, A. Bobek, F. Golatowski. “SIRENA - Service Infrastructure for Real-time Embedded Networked Devices: A service oriented framework for different domains”. IEEE ICONS’06. Pp. 43, April 2006

[37] S. de Deugd, R. Carroll, K. E. Kelly, B. Millett, J. Ricker. ”SODA: Service-Oriented Device Architecture“. IEEE Pervasive omputing. Pp. 94-96, 2006.

[38] M. Taisch, A. W. Colombo, S. Karnouskos, and A. Cannata. “SOCRADES Roadmap: The Future of SOA-based Factory Automation”. SOCRADES Project Publication. 2009

[39] Karnouskos S., Colombo A.W., Lastra J.L.M., Popescu C., "Towards the energy efficient future factory," 7th IEEE International Conference on Industrial Informatics, 2009. INDIN 2009, vol., no., pp.367-371, 23- 26 June 2009

[40] Cannata A., Karnouskos S., Taisch M., "Energy efficiency driven process analysis and optimization in discrete manufacturing," Industrial Electronics, 2009. IECON '09. 35th Annual Conference of IEEE , vol., no., pp.4449-4454, 3-5 Nov. 2009

[41] Anna Florea, Corina Postelnicu, Jose L. Martinez Lastra "A Holistic View on Energy Efficiency In Manufacturing" 2012 International Congress on Informatics, Environment, Energy and Applications-IEEA 2012 IPCSIT vol.38 (2012) © (2012)IACSIT Press, Singapore.

[42] Service-oriented architecture approach for industrial “System of Systems”: State-of-the-Art for Energy Management. Published in: Industrial Informatics (INDIN), 2012 10th IEEE International Conference on Date of Conference: 25-27 July 2012

[43] Curry, E., Hasan, S. ; O'Riain, S. "Enterprise energy management using a linked dataspace for Energy Intelligence", Sustainable Internet and ICT for Sustainability (SustainIT), 2012 on 4-5 Oct. 2012 pages: 1 – 6

[44] K. Tanaka, H. Watanabe, A. Enduo, “Enerize E3 Factory Energy Management System”, Yokogawa Technical Report English EditionVol. 53, No. 1, 2010, pp. 23-26

[45] Tetsuro Tanaka, Shozo Kono,"Using IT to Eliminate Energy Waste in Production Lines - A Case of Yokogawa’s Kofu Factory. Yokogawa Technical Report English Edition Vol.53 No.1 (2010) pp. 39-42

[46] H. Makita, Y. Shida, N. Nozue, “Factory Energy Management System Using Production Information”, Technical Report, Mitsubishi Electric Advance, Dec. 2012, pp. 7-11

[47] Ashish Sarin, Portlets in Action, ISBN: 9781935182542, Manning Publications, 2012

[48] Introduction: Portlet applications [Accessed Online on 17th February, 2014] Available: http://pic.dhe.ibm.com/infocenter/wasinfo/v6r1/index.jsp?topic=%2Fcom.ibm.websphere.express.doc%2Finfo%2Fexp%2Fae%2Fcport_container.html

[49] Jonas X. Yuan, Liferay Portal 6 Enterprise Intranets, ISBN 978-1-849510-38-7 Published by Packt Publishing Ltd 2010

[50] JSR-000154 JAVA™ 2.4 Specification Final Release [Accessed Online on 19th February,2014] Available: https://jcp.org/aboutJava/communityprocess/final/jsr154/

[51] JSR-000286 Portlet Specification 2.0 2.0 Final Release [Accessed Online on 19th February, 2014] Available: https://jcp.org/aboutJava/communityprocess/final/jsr286/

[52] Documentation Liferay [Accessed Online on 21st February, 2014] Available: http://www.liferay.com/documentation/liferay-portal/6.1/user-guide

[53] eXo Platform Overview [Accessed Online on 22nd February, 2014] Available: http://www.exoplatform.com/company/en/products/overview

[54] Alfresco Documentation [Accessed Online on 23rd February, 2014] Available: http://www.alfresco.com/resources/documentation

[55] Richard L. Sezov, Jr. and Stephen Kostas, "Portal Administrator's Guide" Liferay, Inc. [Accessed Online on 28th February, 2014] Available: http://cdn.docs.liferay.com/portal/6.0/official/liferay-administrator-guide-6.0.pdf

[56] Permissioning Liferay, [Accessed Online on 28th February, 2014] Available: https://www.liferay.com/community/wiki/-/wiki/Main/Permissioning+Explained

[57] Liferay Portal 6.1 - Developers Guide [Accessed Online on 1st March, 2014] Available: http://www.liferay.com/documentation/liferay-portal/6.1/development

[58] Enterprise Integration Patterns [Accessed Online on 1st March, 2014] Available: http://www.eaipatterns.com/

[59] Claus Ibsen, Jonathan Anstey, "Camel in Action" ISBN: 9781935182368 , Manning Publications Co. 2011 pp 4-13

[60] Apache Camel, user guide version 2.5.0, Apache Software Foundation, 2010. [Accessed Online on 2nd March, 2014] Available: http://camel.apache.org/manual/camel-manual-2.5.0.pdf

[61] Direct Web Remoting [Accessed Online on 2nd March, 2014] Available: http://directwebremoting.org/dwr/index.html

[62] Java™ API for XML Web Services (JAX-WS) [Accessed Online on 3rd March, 2014] Available: http://docs.oracle.com/javase/7/docs/technotes/guides/xml/jax-ws/

[63] JAX-WS Release Documentation [Accessed Online on 2nd March, 2014] Available: https://jax-ws.java.net/2.2.6/docs/release-documentation.pdf

[64] Highcharts and Highstock documentation [Accessed Online on 2nd March, 2014] Available: http://www.highcharts.com/docs

[65] Dufort, Y. C. ; "ISA-95-Based Operations and KPI Metrics assessment and Analysis"; White paper 24, A Mesa International, ISA and Ivensys Wonderware co-branded white paper , Nov, 28 2006

[66] S 1000 Overview [Accessed Online on 5th March, 2014] Available: http://www.inicotech.com/s1000_overview.html

[67] Liferay installation on Tomacat. [Accessed Online on 5th March, 2014] Available: http://www.liferay.com/documentation/liferay-portal/6.1/user-guide/-/ai/lp-6-1-ugen11-installing-liferay-on-tomcat-7-0

APPENDIX A- KPI SERVICES

```
<wsdl:definitions targetNamespace="http://tut.fi/DatastoreService"
xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"
xmlns:sch="http://tut.fi/DatastoreService"
xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
xmlns:tns="http://tut.fi/DatastoreService">
   <wsdl:types>
      <xs:schema elementFormDefault="qualified" target-
Namespace="http://tut.fi/DatastoreService"
xmlns:xs="http://www.w3.org/2001/XMLSchema">
         <xs:element name="ProductionRateRequest">
            <xs:complexType>
               <xs:sequence>
                  <xs:element name="cellId" type="xs:int"/>
               </xs:sequence>
            </xs:complexType>
         </xs:element>
         <xs:element name="ProductionRateResponse">
            <xs:complexType>
               <xs:sequence>
                  <xs:element name="cellId" type="xs:int"/>
                  <xs:element name="productionRate" type="xs:long"/>
                  <xs:element name="timestamp" type="xs:long"/>
               </xs:sequence>
            </xs:complexType>
         </xs:element>
         <xs:element name="TotalProductsRequest">
            <xs:complexType>
               <xs:sequence>
                  <xs:element name="cellId" type="xs:int"/>
               </xs:sequence>
            </xs:complexType>
         </xs:element>
         <xs:element name="TotalProductsResponse">
            <xs:complexType>
               <xs:sequence>
                  <xs:element name="cellId" type="xs:int"/>
                  <xs:element name="totalProducts" type="xs:long"/>
                  <xs:element name="timestamp" type="xs:long"/>
               </xs:sequence>
            </xs:complexType>
         </xs:element>
         <xs:element name="PalletProductionTimeRequest">
            <xs:complexType>
               <xs:sequence>
                  <xs:element name="cellId" type="xs:int"/>
               </xs:sequence>
            </xs:complexType>
         </xs:element>
         <xs:element name="PalletProductionTimeResponse">
            <xs:complexType>
               <xs:sequence>
                  <xs:element name="cellId" type="xs:int"/>
                  <xs:element name="palletId" type="xs:int"/>
                  <xs:element name="productionTime" type="xs:long"/>
                  <xs:element name="timestamp" type="xs:long"/>
               </xs:sequence>
            </xs:complexType>
         </xs:element>
         <xs:element name="CamxStatesOverviewRequest">
            <xs:complexType>
               <xs:sequence>
                  <xs:element name="cellId" type="xs:int"/>
               </xs:sequence>
```

```
            </xs:complexType>
         </xs:element>
         <xs:element name="CamxStatesOverviewResponse">
            <xs:complexType>
               <xs:sequence>
                  <xs:element name="cellId" type="xs:int"/>
                  <xs:element maxOccurs="unbounded" minOccurs="0"
name="stateOccurences">
                     <xs:complexType>
                        <xs:simpleContent>
                           <xs:extension base="xs:long">
                              <xs:attribute name="state" type="xs:string"/>
                           </xs:extension>
                        </xs:simpleContent>
                     </xs:complexType>
                  </xs:element>
                  <xs:element name="timestamp" type="xs:long"/>
               </xs:sequence>
            </xs:complexType>
         </xs:element>
         <xs:element name="HistoricalDataRequest">
            <xs:complexType>
               <xs:sequence>
                  <xs:element maxOccurs="1" minOccurs="1" name="cellsList">
                     <xs:complexType>
                        <xs:sequence>
                           <xs:element maxOccurs="unbounded" minOccurs="1"
name="cell" type="xs:int"/>
                        </xs:sequence>
                     </xs:complexType>
                  </xs:element>
                  <xs:element maxOccurs="1" minOccurs="1" name="specificationsList">
                     <xs:complexType>
                        <xs:sequence>
                           <xs:element maxOccurs="4" minOccurs="1"
name="specification" type="xs:string"/>
                        </xs:sequence>
                     </xs:complexType>
                  </xs:element>
                  <xs:element maxOccurs="1" minOccurs="1" name="parametersList">
                     <xs:complexType>
                        <xs:sequence>
                           <xs:element maxOccurs="unbounded" minOccurs="1"
name="parameter" type="xs:string"/>
                        </xs:sequence>
                     </xs:complexType>
                  </xs:element>
                  <xs:element maxOccurs="1" minOccurs="1" name="timeRangeList">
                     <xs:complexType>
                        <xs:sequence>
                           <xs:element maxOccurs="unbounded" minOccurs="1"
name="timeRange">
                              <xs:complexType>
                                 <xs:sequence>
                                    <xs:element maxOccurs="1" minOccurs="1"
name="timeStart" type="xs:long"/>
                                    <xs:element maxOccurs="1" minOccurs="1"
name="timeEnd" type="xs:long"/>
                                 </xs:sequence>
                                 <xs:attribute name="id" type="xs:int"/>
                              </xs:complexType>
                           </xs:element>
                        </xs:sequence>
                     </xs:complexType>
                  </xs:element>
               </xs:sequence>
```

```
            </xs:complexType>
        </xs:element>
        <xs:element name="HistoricalDataResponse">
            <xs:complexType>
                <xs:sequence>
                    <xs:element maxOccurs="2" minOccurs="1" name="Timerange">
                        <xs:complexType>
                            <xs:sequence>
                                <xs:element maxOccurs="unbounded" minOccurs="0"
name="Data">
                                    <xs:complexType>
                                        <xs:sequence>
                                            <xs:element name="cellsList">
                                                <xs:complexType>
                                                    <xs:sequence>
                                                        <xs:element maxOccurs="unbounded" minOc-
curs="0" name="cell">
                                                            <xs:complexType>
                                                                <xs:sequence>
                                                                    <xs:element max-
Occurs="unbounded" minOccurs="0" name="valuesList">
                                                                        <xs:complexType>
                                                                            <xs:sequence>
                                                                                <xs:element max-
Occurs="unbounded" minOccurs="0" name="value">
                                                                                    <xs:complexType>

<xs:simpleContent>
                                                                                            <xs:extension
base="xs:float">

<xs:attribute name="timestamp" type="xs:long" use="required"/>

</xs:extension>

</xs:simpleContent>
                                                                                    </xs:complexType>
                                                                                </xs:element>
                                                                            </xs:sequence>
                                                                            <xs:attribute
name="specification" type="xs:string" use="required"/>
                                                                        </xs:complexType>
                                                                    </xs:element>
                                                                </xs:sequence>
                                                                <xs:attribute name="id"
type="xs:int" use="required"/>
                                                            </xs:complexType>
                                                        </xs:element>
                                                    </xs:sequence>
                                                </xs:complexType>
                                            </xs:element>
                                        </xs:sequence>
                                        <xs:attribute name="parameter" type="xs:string"
use="required"/>
                                    </xs:complexType>
                                </xs:element>
                            </xs:sequence>
                            <xs:attribute name="id" type="xs:int" use="required"/>
                        </xs:complexType>
                    </xs:element>
                </xs:sequence>
            </xs:complexType>
        </xs:element>
        <xs:element name="SingleKpiRequest">
            <xs:complexType>
                <xs:sequence>
```

```
                  <xs:element maxOccurs="1" minOccurs="1" name="request"
type="xs:boolean"/>
               </xs:sequence>
            </xs:complexType>
         </xs:element>
         <xs:element name="SingleKpiResponse">
            <xs:complexType>
               <xs:sequence>
                  <xs:element maxOccurs="2" minOccurs="1" name="timeRange">
                     <xs:complexType>
                        <xs:sequence>
                           <xs:element maxOccurs="unbounded" minOccurs="0"
name="singleValue">
                              <xs:complexType>
                                 <xs:simpleContent>
                                    <xs:extension base="xs:float">
                                       <xs:attribute name="id" type="xs:string"
use="required"/>
                                       <xs:attribute name="timestamp" type="xs:int"
use="optional"/>
                                    </xs:extension>
                                 </xs:simpleContent>
                              </xs:complexType>
                           </xs:element>
                        </xs:sequence>
                        <xs:attribute name="id" type="xs:int" use="required"/>
                     </xs:complexType>
                  </xs:element>
               </xs:sequence>
            </xs:complexType>
         </xs:element>
         <xs:element name="ProductionKpiRequest">
            <xs:complexType>
               <xs:sequence>
                  <xs:element name="timeStart" type="xs:long"/>
                  <xs:element name="timeEnd" type="xs:long"/>
               </xs:sequence>
            </xs:complexType>
         </xs:element>
         <xs:element name="ProductionKpiResponse">
            <xs:complexType>
               <xs:sequence>
                  <xs:element maxOccurs="3" minOccurs="1" name="ProductionKpi">
                     <xs:complexType>
                        <xs:sequence>
                           <xs:element maxOccurs="unbounded" minOccurs="1"
name="ProductionKpiValue">
                              <xs:complexType>
                                 <xs:simpleContent>
                                    <xs:extension base="xs:float">
                                       <xs:attribute name="cell" type="xs:int"
use="optional"/>
                                    </xs:extension>
                                 </xs:simpleContent>
                              </xs:complexType>
                           </xs:element>
                        </xs:sequence>
                        <xs:attribute name="id" type="xs:string" use="required"/>
                     </xs:complexType>
                  </xs:element>
               </xs:sequence>
            </xs:complexType>
         </xs:element>
      </xs:schema>
   </wsdl:types>
   <wsdl:message name="TotalProductsResponse">
```

```
      <wsdl:part element="tns:TotalProductsResponse" name="TotalProductsResponse"/>
   </wsdl:message>
   <wsdl:message name="CamxStatesOverviewRequest">
      <wsdl:part element="tns:CamxStatesOverviewRequest"
name="CamxStatesOverviewRequest"/>
   </wsdl:message>
   <wsdl:message name="SingleKpiRequest">
      <wsdl:part element="tns:SingleKpiRequest" name="SingleKpiRequest"/>
   </wsdl:message>
   <wsdl:message name="PalletProductionTimeResponse">
      <wsdl:part element="tns:PalletProductionTimeResponse"
name="PalletProductionTimeResponse"/>
   </wsdl:message>
   <wsdl:message name="ProductionRateRequest">
      <wsdl:part element="tns:ProductionRateRequest" name="ProductionRateRequest"/>
   </wsdl:message>
   <wsdl:message name="HistoricalDataRequest">
      <wsdl:part element="tns:HistoricalDataRequest" name="HistoricalDataRequest"/>
   </wsdl:message>
   <wsdl:message name="SingleKpiResponse">
      <wsdl:part element="tns:SingleKpiResponse" name="SingleKpiResponse"/>
   </wsdl:message>
   <wsdl:message name="ProductionRateResponse">
      <wsdl:part element="tns:ProductionRateResponse" name="ProductionRateResponse"/>
   </wsdl:message>
   <wsdl:message name="TotalProductsRequest">
      <wsdl:part element="tns:TotalProductsRequest" name="TotalProductsRequest"/>
   </wsdl:message>
   <wsdl:message name="CamxStatesOverviewResponse">
      <wsdl:part element="tns:CamxStatesOverviewResponse"
name="CamxStatesOverviewResponse"/>
   </wsdl:message>
   <wsdl:message name="PalletProductionTimeRequest">
      <wsdl:part element="tns:PalletProductionTimeRequest"
name="PalletProductionTimeRequest"/>
   </wsdl:message>
   <wsdl:message name="HistoricalDataResponse">
      <wsdl:part element="tns:HistoricalDataResponse" name="HistoricalDataResponse"/>
   </wsdl:message>
   <wsdl:message name="ProductionKpiRequest">
      <wsdl:part element="tns:ProductionKpiRequest" name="ProductionKpiRequest"/>
   </wsdl:message>
   <wsdl:message name="ProductionKpiResponse">
      <wsdl:part element="tns:ProductionKpiResponse" name="ProductionKpiResponse"/>
   </wsdl:message>
   <wsdl:portType name="Schema">
      <wsdl:operation name="TotalProducts">
         <wsdl:input message="tns:TotalProductsRequest" name="TotalProductsRequest"/>
         <wsdl:output message="tns:TotalProductsResponse"
name="TotalProductsResponse"/>
      </wsdl:operation>
      <wsdl:operation name="CamxStatesOverview">
         <wsdl:input message="tns:CamxStatesOverviewRequest"
name="CamxStatesOverviewRequest"/>
         <wsdl:output message="tns:CamxStatesOverviewResponse"
name="CamxStatesOverviewResponse"/>
      </wsdl:operation>
      <wsdl:operation name="SingleKpi">
         <wsdl:input message="tns:SingleKpiRequest" name="SingleKpiRequest"/>
         <wsdl:output message="tns:SingleKpiResponse" name="SingleKpiResponse"/>
      </wsdl:operation>
      <wsdl:operation name="PalletProductionTime">
         <wsdl:input message="tns:PalletProductionTimeRequest"
name="PalletProductionTimeRequest"/>
         <wsdl:output message="tns:PalletProductionTimeResponse"
name="PalletProductionTimeResponse"/>
```

```
      </wsdl:operation>
      <wsdl:operation name="ProductionRate">
         <wsdl:input message="tns:ProductionRateRequest"
name="ProductionRateRequest"/>
         <wsdl:output message="tns:ProductionRateResponse"
name="ProductionRateResponse"/>
      </wsdl:operation>
      <wsdl:operation name="HistoricalData">
         <wsdl:input message="tns:HistoricalDataRequest"
name="HistoricalDataRequest"/>
         <wsdl:output message="tns:HistoricalDataResponse"
name="HistoricalDataResponse"/>
      </wsdl:operation>
      <wsdl:operation name="ProductionKpi">
         <wsdl:input message="tns:ProductionKpiRequest" name="ProductionKpiRequest"/>
         <wsdl:output message="tns:ProductionKpiResponse"
name="ProductionKpiResponse"/>
      </wsdl:operation>
   </wsdl:portType>
   <wsdl:binding name="SchemaSoap11" type="tns:Schema">
      <soap:binding style="document"
transport="http://schemas.xmlsoap.org/soap/http"/>
      <wsdl:operation name="TotalProducts">
         <soap:operation soapAction=""/>
         <wsdl:input name="TotalProductsRequest">
            <soap:body use="literal"/>
         </wsdl:input>
         <wsdl:output name="TotalProductsResponse">
            <soap:body use="literal"/>
         </wsdl:output>
      </wsdl:operation>
      <wsdl:operation name="CamxStatesOverview">
         <soap:operation soapAction=""/>
         <wsdl:input name="CamxStatesOverviewRequest">
            <soap:body use="literal"/>
         </wsdl:input>
         <wsdl:output name="CamxStatesOverviewResponse">
            <soap:body use="literal"/>
         </wsdl:output>
      </wsdl:operation>
      <wsdl:operation name="SingleKpi">
         <soap:operation soapAction=""/>
         <wsdl:input name="SingleKpiRequest">
            <soap:body use="literal"/>
         </wsdl:input>
         <wsdl:output name="SingleKpiResponse">
            <soap:body use="literal"/>
         </wsdl:output>
      </wsdl:operation>
      <wsdl:operation name="PalletProductionTime">
         <soap:operation soapAction=""/>
         <wsdl:input name="PalletProductionTimeRequest">
            <soap:body use="literal"/>
         </wsdl:input>
         <wsdl:output name="PalletProductionTimeResponse">
            <soap:body use="literal"/>
         </wsdl:output>
      </wsdl:operation>
      <wsdl:operation name="ProductionRate">
         <soap:operation soapAction=""/>
         <wsdl:input name="ProductionRateRequest">
            <soap:body use="literal"/>
         </wsdl:input>
         <wsdl:output name="ProductionRateResponse">
            <soap:body use="literal"/>
         </wsdl:output>
```

```
    </wsdl:operation>
    <wsdl:operation name="HistoricalData">
      <soap:operation soapAction=""/>
      <wsdl:input name="HistoricalDataRequest">
        <soap:body use="literal"/>
      </wsdl:input>
      <wsdl:output name="HistoricalDataResponse">
        <soap:body use="literal"/>
      </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="ProductionKpi">
      <soap:operation soapAction=""/>
      <wsdl:input name="ProductionKpiRequest">
        <soap:body use="literal"/>
      </wsdl:input>
      <wsdl:output name="ProductionKpiResponse">
        <soap:body use="literal"/>
      </wsdl:output>
    </wsdl:operation>
  </wsdl:binding>
  <wsdl:service name="SchemaService">
    <wsdl:port binding="tns:SchemaSoap11" name="SchemaSoap11">
      <soap:address location="http://130.230.182.28:8080/FastoryDatastore/ws/"/>
    </wsdl:port>
  </wsdl:service>
</wsdl:definitions>
```

APPENDIX B- ENERGY MESSAGE

Online Energy Message from Middleware:


```
{ "EnergyMeter" : {
    "@AIRMS" : "0.44",
    "@AVA" : "107.62",
    "@AVAHR" : "49",
    "@AVAR" : "-91.80",
    "@AVARHR" : "-41",
    "@AVRMS" : "237.05",
    "@AWATT" : "290.00",
    "@AWATTHR" : "0",
    "@BIRMS" : "0.23",
    "@BVA" : "55.37",
    "@BVAHR" : "27",
    "@BVAR" : "-49.79",
    "@BVARHR" : "-24",
    "@BVRMS" : "236.97",
    "@BWATT" : "385.53",
    "@BWATTHR" : "7",
    "@CIRMS" : "1.45",
    "@CVA" : "351.78",
    "@CVAHR" : "156",
    "@CVAR" : "-65.27",
    "@CVARHR" : "-30",
    "@CVRMS" : "234.76",
    "@CWATT" : "115.15",
    "@CWATTHR" : "376",
    "@LINEFREQ" : "50.00",
    "@cellId" : "4",
    "@dateTime" : "2014-06-11T14:55:08.0",
    "@devType" : "energyMeter"
  } }
```

APPENDIX C- KPI LIST

ID EMS001	Title: Active Electrical Power by CELLs
Mathematical Expression	$P_{NY} = I \times V \times Power\ Factor$
Notations	Active Power: P_{NY} [WATT] where NY represents flowingly: N= number of cell Y= for cells, Y replaces by R to symbolize robot, CNR for conveyor, CTR for controller and empty for the complete cell. e.g. Power of Cell 1: P_{cl1} Robot Power of Cell 1: P_{cl1R} Electric Current: I [A] ;EMS004 Voltage: V [V]; EMS005
Description	Active Power is true power consumed by specific components of Production System.
Comments	This KPI defines the true demand of the system. It is measured by the Energy meter real time. A harmonic timeline view of this KPI represent, trend of demand in the system.
Input Data	Active Power of CELL(s) id and specification (R, CNR, CTR or total) Start and End Timestamp.

ID EMS002	Title: Apparent Electrical Power
Mathematical Expression	$R_{NY} = V \times I$
Notations	Apparent Electrical Power: R_{NY} [VAR] Where NY represents similarly EMS001. Electric Current: I [A] ;EMS004 Voltage: V [V]; EMS005
Description	Apparent Power is multiplication Voltage and Ampere in the System.
Comments	It is necessary to monitor Apparent Power also as it is an element of creating power factor.
Input Data	Apparent Power of CELL(s) id and specification (R, CNR, CTR or total) Start and End Timestamp.

ID EMS003	Title: Reactive Electrical Power
Mathematical Expression	$Q_{NY} = V \times I \times \sin\varphi$
Notations	Reactive Power: Q_{NY} [VAR] Where NY represents similarly EMS001. Electric Current: I [A] Voltage: V [V]
Description	Reactive Power is the angle between the current and voltage
Comments	It is necessary to monitor Reactive Power also as most often it causes extra charge in electric bill.
Input Data	Reactive Power of CELL(s) id and specification (R, CNR, CTR or total) Start and End Timestamp.

ID EMS004	Title: Root mean square Voltage
Mathematical Expression	$$V_{rms} = \frac{V_p}{\sqrt{2}}$$
Notations	Root mean square Voltage: V_{rms} [rmsV] Peak Voltage: V_p [V]
Description	RMS (root mean square) voltage is the equivalent DC voltage.
Comments	This KPI helps to monitor harmony of voltage supply. As most of renewable energy comes in form of DC voltage this value would also help Managers to consider the needed amount of renewable energy resource units.
Input Data	CELL(s) id Start and End Timestamp.

ID EMS005	Title: Root mean square Current
Mathematical Expression	$$I_{rms} = \frac{I_p}{\sqrt{2}}$$
Notations	Root mean square Current: I_{rms} [rmsA] Peak Current: I_p [A]
Description	RMS (root mean square) of Current is the equivalent to steady DC.
Comments	This Kpi helps to monitor harmony of Current supply. I_{rms} Value denotes heat (Thermal Energy) created by the component itself. It can warn the monitoring system to avoid damage in advance.
Input Data	CELL(s) id Start and End Timestamp.

ID EMS006	Title: Power Factor This Month by Cells
Mathematical Expression	$$cos\varphi = \frac{AcPCbyCELL[i]}{ApPCbyCELL[i]}$$
Notations	Power Factor: $cos\varphi$ Phase difference: φ $AcPCbyCELL$: Average Active Power Consumption This Month by cell $ApPCbyCELL$: Average Apparent Power Consumption This Month by cell $[i]$: Cell ID
Description	Power Factor is ratio of active power and apparent power in percentage.
Comments	Power Factor improvement is the significant fragment for energy savings. Since true power consumption depends upon this factor. A cost effective Power factor is greater than 0.95
Input Data	Average Active Power Consumption by Cell from data source Average Apparent Power Consumption by Cell from data source Optional Timestamp to compare value for larger or shorter period.

ID EMS007	Title: Power Factor This Month for complete System
Mathematical Expression	$$cos\varphi = \frac{\sum AcPCbyCELL[i]}{\sum ApPCbyCELL[i]}$$
Notations	Power Factor: $cos\varphi$ Phase difference: φ $AcPCbyCELL$: Average Active Power Consumption This Month by complete production System $ApPCbyCELL$: Average Apparent Power Consumption This Month by complete production System $[i]$: Cell ID
Description	Power Factor is ratio of active power and apparent power in percentage.
Comments	Power Factor improvement is the significant fragment for energy savings. This is KPI assists to figure, overall Power Factor of manufacturing system. Since true power consumption depends upon this factor. A cost effective Power factor is greater than 0.95
Input Data	Average Active Power Consumption by all Cells from data source Average Apparent Power Consumption by all Cells from data source Optional Timestamp to compare value for larger or shorter period.

ID EMS008	Title: Active Electrical Energy Consumption by Cell
Mathematical Expression	$$W_{XNY}^{"} = W_{XNY}^{t_2} - W_{XNY}^{t_1}$$
Notations	$W_{XNY}^{"}$ [KWh] :Active Energy Consumption by Cell $W_{XNY}^{t_2}$ [KWh]: Recorded Active Energy Consumption by Cell at start date time. $W_{XNY}^{t_1}$ [KWh]: Recorded Active Energy Consumption by Cell at end date time. Period of Measurement: t_1 to t_2
Description	Measured active electric consumption of each cell.
Comments	This KPI gives figure about active energy consumption of specific cell. Cost can be calculated directly with this KPI A graph relates this KPI with KPI cells production rate, cell process units, cell quality rate would be helpful tool to understand system performance.
Input Data	Cell number(i) Period of Measurement: from t_2 to t_1

ID EMS009	Title: Active Electrical Energy Consumption by Cell by process
Mathematical Expression	$$Energy\ Consumption\ BCP = \left[W_{XNY}^{"} \bigcap Process\ ID[j]\right]_{t_1}^{t_2}$$
Notations	BCP = Energy Consumption by Cell by process Period of Measurement: t_1 to t_2 Similar to EMS006
Description	Total Active Energy Consumption by Cell, Categorized by Processes.
Comments	This KPI extends monitoring capabilities of previous KPI EMS009, so that the energy consumption by cell for specific process can also be measured.
Input Data	CELLID[i] PorcessID[j] Period of Measurement: from t_2 to t_1

ID EMS0010	Title: Total Active Electrical Energy Consumption for complete Production Line
Mathematical Expression	$$W_{TAEC} = \left[\sum_{n=0}^{N} W_{TMAEC}\right]_{t_1}^{t_2};$$
Notations	W_{TAEC} [KWh]: Total Active Energy Consumption W_{TMAEC} [KWh]: Measured Energy Consumption value Period of Measurement: t_1 to t_2
Description	Total active energy consumption is summation of measured active electric consumption of components, actively involved in production system at FASTORY. More specifically, Cells and bypass Conveyors.
Comments	This KPI gives figure about overall active energy consumption of production system in processes. Cost can be calculated directly with this KPI.
Input Data	Period of Measurement: t_1 to t_2

ID EMS0011	Title: Total Reactive Electrical Energy Consumption for complete Production Line
Mathematical Expression	$$W_{TREC} = \left[\sum_{n=0}^{N} W_{TMREC}\right]_{t_1}^{t_2};$$
Notations	W_{TREC} [KWh]: Total Reactive Energy Consumption W_{TMREC} [KWh]: Measured Reactive Energy Consumption value from all the energy meters of production line. Period of Measurement: t_1 to t_2
Description	Total reactive energy consumption is summation of measured reactive electric consumption of components, actively involved in production system at FASTORY. More specifically, Cells and bypass Conveyors.
Comments	This KPI gives figure about overall reactive energy consumption of production system in processes.
Input Data	Period of Measurement: t_1 to t_2

ID EMS0012	Title: Peak Load
Mathematical Expression	$$P_p = MAX\left(\sum P_{XNY}\right)$$
Notations	Active Power: P_{XNY} [WATT]; refer to EMS001 Peak Load: P_p [WATT]
Description	Maximum load of the manufacturing system
Comments	Overloading of Manufacturing system can cause damage, It also has penalty in Energy Bill. So it is necessary to identify Peak Load events.
Input Data	Period of Measurement: t_1 to t_2

ID EMS0013	Title: Process Units by Cell
Mathematical Expression	$PsU[i] = \left[\sum CPs\right]_{t_1}^{t_2}$
Notations	PsU: Completed Process Units [i]: Cell ID CPs: Completed Processes Period of Measurement: t_1 to t_2
Description	Number of Processes done by Cell for defined time period
Comments	This Kpi is used to monitor Hourly processed units, Daily Processed units, Monthly processed Units or for any specific Time Period.
Input Data	Cell ID; ProcessID (Optional) Period of Measurement: t_1 to t_2

ID EMS0014	Title: Completed Product Units by Production System
Mathematical Expression	$PrU = \left[\sum CPr\right]_{t_1}^{t_2}$
Notations	PrU: Completed Product Units CPr: Number of Completed Product at specific time Period of Measurement: t_1 to t_2
Description	Number of Completed Product produced in specified time period
Comments	This Kpi is used to monitor Hourly produced units, Daily produced units, Monthly produced Units or for any specific Time Period.
Input Data	Period of Measurement: t_1 to t_2

ID EMS0015	Title: Cell Production Rate
Mathematical Expression	$PRT[i] = \left(\sum CP\right)/HRS$
Notations	PRT: Production rate CP: Completed Process HRS: Number of hours in selected Time Stamp.
Description	Hourly Production Rate by cells
Comments	This Kpi in used to compare with specific hours' production rate and average production rate. Or to compare with efficiency of other cells. It also used in many graphs to compare with Energy Consumptions.
Input Data	Cell ID Period of Measurement: t_1 to t_2

ID EMS0016	Title: Complete Production Rate
Mathematical Expression	$CPRT = \left(\sum LCP\right)/HRS$
Notations	CPRT: Production rate of completed Units LCP: Completed Products HRS: Number of hours in selected Time Stamp.
Description	Hourly Production Rate by Manufacturing System
Comments	This Kpi in used to compare with specific hours' production rate and average production rate. It also used in many graphs to compare with Energy Consumptions.
Input Data	Cell ID [i]; Period of Measurement: t_1 to t_2

ID EMS0017	Title: Energy Consumption Per Process by cell
Mathematical Expression	$ECP[i] = W^{"}_{XNY} / PsU$
Notations	ECP: Energy Consumption per Process $W^{"}_{XNY}$: EMS008 PsU : EMS0013
Description	Energy Consumption per Process by specific Cell
Comments	This KPI helps to track performance of Specific CELL
Input Data	Cell ID [i]; Period of Measurement: t_1 to t_2

ID EMS0018	Title: Energy Consumption per Specific Process by Cell
Mathematical Expression	$ECP"[i] = BCP / PsU$
Notations	ECP": Energy Consumption per specific Process by cell BCP: EMS009 PsU : EMS0013 with ProcessID
Description	Energy Consumption per specific Process by specific Cell
Comments	This KPI helps to track performance of Specific CELL for specific Process
Input Data	Cell ID [i]; ProcessID; Period of Measurement: t_1 to t_2

ID EMS0019	Title: Energy Consumption Per Complete Product
Mathematical Expression	$ECPr = W_{TAEC} / PrU$
Notations	ECPr: Energy Consumption Per Product W_{TAEC} : EMS0010 PrU: EMS0014
Description	Average Energy Consumption Per Complete Product
Comments	This KPI helps to find Average Energy Consumption per product that can be displayed to show overall energy cost per product in manufacturing system.
Input Data	Period of Measurement: t_1 to t_2

ID EMS0020	Title: Energy Consumption of Specific Pallet
Mathematical Expression	$ECSP = W_{TAEC} \cap PalletID$
Notations	ECSP = Energy Consumption of Specific Pallet W_{TAEC} : Total Energy Consumption PalletID: Pallet ID.
Description	Average Energy Consumption of Specific Completed Product
Comments	This KPI helps to find Energy Consumption of Specific Completed Product.
Input Data	PalletID

ID EMS0021	Title: Average Unit Production Time
Mathematical Expression	$UPrT[i] = TI / PrU$
Notations	UPrT: Average Production Time Per Unit TI : Time Interval PrU: EMS0014
Description	Average Unit Production Time
Comments	This Kpi helps to predict how long it would take, to complete order of the production.
Input Data	Period of Measurement: t_1 to t_2

ID EMS0022	Title: Average Unit Process Time
Mathematical Expression	$UPsT[i] = TI/PsU$
Notations	UPsT = Unit Process Time TI = Time Interval PsU : EMS0013
Description	Average Time to complete single Process in Manufacturing system
Comments	This KPI depicts performance of Cells.
Input Data	Cell ID Period of Measurement: t_1 to t_2

ID EMS0023	Title: Total Process Units
Mathematical Expression	$TPsU = \left[\sum TCPs\right]_{t_1}^{t_2}$
Notations	TPsU: Total Completed Process Units TCPs: Total Completed Processes by whole system Period of Measurement: t_1 to t_2
Description	Total Process Units produced by all cells.
Comments	This KPI helps to compare with produced complete product to monitor quality rate.
Input Data	Cell ID; ProcessID (Optional) Period of Measurement: t_1 to t_2

ID EMS0024	Title: Frame Quality By Cell
Mathematical Expression	$FQC[i] = (TFQC[i]/TFC[i]) \times 100$
Notations	FQC: Frame quality by cell [i]: Cell ID TFQC : Number of Frames passed quality control by this Cell TFC : Equivalents to EMS0013 but with ProcessID Frame drawing
Description	Percentage of frame that meets the quality criteria by selected Cell
Comments	This KPI helps to detect defected cells for Frame drawings.
Input Data	Cell ID Period of Measurement: t_1 to t_2

ID EMS0025	Title: Frame Quality by complete System
Mathematical Expression	$OFQ = (TFQC/TFC) \times 100$
Notations	FQ: Overall Frame quality [] TFQC : Number of Frames passed quality control TFC : Equivalents to EMS0023 TPsU but with ProcessID Frame drawing
Description	Percentage of frame that meets the quality criteria by complete system
Comments	This KPI helps to monitor Frame Drawing quality of manufacturing system
Input Data	Period of Measurement: t_1 to t_2

ID EMS0026	Title: Keyboard Quality By Cell
Mathematical Expression	$KQC[i] = (TKQC[i]/TKC[i]) \times 100$
Notations	KQC: Keyboard quality by cell [i]: Cell ID TKQC : Number of Keyboards passed quality control by this Cell TKC : Equivalents to EMS0013 but with ProcessID Frame drawing
Description	Percentage of Keyboards that meets the quality criteria by specific Cell
Comments	This KPI helps to detect defected cells for Keyboard drawings.
Input Data	Period of Measurement: t_1 to t_2

ID EMS0027	Title: Keyboard Quality by complete System
Mathematical Expression	$OKQ = (TKQC/TKC) \times 100$
Notations	FQ: Overall Keyboard quality produced by whole system TFQC : Number of Keyboards passed quality control TFC : Equivalents to EMS0022 TPsU but with ProcessID Keyboard drawing
Description	Percentage of keyboards that meets the quality criteria by complete system
Comments	This KPI helps to monitor Keyboard Drawing quality of manufacturing system
Input Data	Period of Measurement: t_1 to t_2

ID EMS0028	Title: Screen Quality By Cell
Mathematical Expression	$SQC[i] = (TSQC[i]/TSC[i]) \times 100$
Notations	FQC: Screen quality by cell [i]: Cell ID TFQC : Number of Screens passed quality control, produced by this Cell TFC : Equivalents to EMS0012 but with ProcessID Screen drawing
Description	Percentage of Screens that meets the quality criteria drawn in specific cell
Comments	This KPI helps to detect defected cells for Screen drawings.
Input Data	Cell ID Period of Measurement: t_1 to t_2

ID EMS0029	Title: Screen Quality by complete System
Mathematical Expression	$OSQ = (TSQC/TSC) \times 100$
Notations	FQ: Overall Screen quality produced by production line TSQC : Number of Screens passed quality control TSC : Equivalents to EMS0022 TPsU but with ProcessID Screen drawing
Description	Percentage of Screens that meets the quality criteria drawn in manufacturing system for specific time period
Comments	This KPI helps to monitor Screen Drawing quality of manufacturing system
Input Data	Period of Measurement: t_1 to t_2

ID EMS0030	Title: Final Product Quality rate by Production System
Mathematical Expression	$FPQ = (NCPq/PrU) \times 100$
Notations	FPQ: Final Product quality produced by production line NCPq : Number of Completed Product passed quality control PrU : EMS0013
Description	Percentage of Total Completed Product passed quality inspection
Comments	This KPI helps to monitor overall quality of manufacturing system
Input Data	Period of Measurement: t_1 to t_2

ID EMS0031	Title: Percentage of IPC2541 States by CELL
Mathematical Expression	$PIPC = (TSPIPC/TI) \times 100$
Notations	PIPC: Percentage of IPC2541 States TSPIPC : Time when cell remained in Specific State TI : Time Interval
Description	Percentage of each IPC2541 state in total Production Time
Comments	This Kpi Helps to see how many percentage Cells were in different states
Input Data	StateID; CellID; Period of Measurement: t_1 to t_2

ID EMS0032	Title: Energy Consumption at IPC2541 States by CELL
Mathematical Expression	$EnIPC = \sum(W_{XNY}^{t_2} - W_{XNY}^{t_1})$
Notations	EnIPC: Energy Consumption at IPC2541 States by CELL $W_{XNY}^{t_2}$ [KWh]: Recorded Active Energy Consumption by Cell at start date time. $W_{XNY}^{t_1}$ [KWh]: Recorded Active Energy Consumption by Cell at end date time. State Start time: t_1 State End Time: t_2
Description	Energy Consumption at IPC2541 States by CELL
Comments	This KPI figure out defected cells by monitoring cells energy consumption in their deferent states.
Input Data	StateID; CellID; Period of Measurement: T_1 to T_2

ID EMS0033	MTTR
Mathematical Expression	$MTTR = \left(\sum(Str - End)_{MTTR}\right) \div N_{MTTR}$
Notations	MTTR: Mean Time to Repair Each Cell $(Str - End)_{MTTR}$: Time needed to repair cell N_{MTTR}: Number of time Cell needed to Repair
Description	Mean Time to Repair Each Cell
Comments	Helps to depict Workers Efficiency
Input Data	CellID; Period of Measurement: T_1 to T_2

ID EMS0034	MTTF
Mathematical Expression	$MTTF = \left(\sum(Str - End)_{MTTF}\right) \div N_{MTTF}$
Notations	MTTR: Mean Time between Failure of Each Cell $(Str - End)_{MTTR}$: Time between Failure of Each Cell N_{MTTR}: Number of times Cell Failed to Run.
Description	Mean Time between Failure of Each Cell
Comments	Helps to understand reliability of system
Input Data	CellID; Period of Measurement: T_1 to T_2

ID EMS0035	Tile: Air Pressure at compressed air
Mathematical Expression	
Notations	Measured by Sensors in mm
Description	
Comments	It is a Measured KPI to compare with EMS0039
Input Data	Period of Measurement: T_1 to T_2

ID EMS0036	Title: Air Flow
Mathematical Expression	
Notations	AF: Air Flow to Production System
Description	Air flow to production system by compressed air supply system. Measured by sensor
Comments	It helps to track air flow to production system.
Input Data	Period of Measurement: T_1 to T_2

ID EMS0037	Title: Air Consumption per Product
Mathematical Expression	$ACP = \left(\sum Av\right) \div PrU$
Notations	ACP: Air Consumption per Product Av: Volume measured by sensors PrU: Number of Product Units
Description	Air Consumption per Product
Comments	It shows how much air is used by single product
Input Data	Period of Measurement: T_1 to T_2

ID EMS0038	Title: Humidity in Compressed Air
Mathematical Expression	
Notations	Measured by Sensor.
Description	Humidity in Compressed Air
Comments	This Kpi helps to avoid damage production because of over humidity
Input Data	Period of Measurement: T_1 to T_2

ID EMS0039	Title: Energy Consumption by Air Compressor
Mathematical Expression	$EnCAC = LstRd - FstRd$
Notations	$EnCAC$: Energy Consumption by Air Compressor $LstRd$: Energy Consumption recorded at start time FstRd: Energy Consumption recorded at end time.
Description	Energy Consumption by Air Compressor for specific time period
Comments	This Kpi helps to measure energy consumption by supplementary components of Production System.
Input Data	Period of Measurement: T_1 to T_2

ID EMS0040	Title: Energy Consumption by Air Compressor per Product
Mathematical Expression	$PEnCAC = EnCAC \div PrU$
Notations	$PEnCAC$: Energy Consumption by Air Compressor per Product $EnCAC$: EMS0039 PrU: EMS0014
Description	Energy Consumption by Air Compressor per Product
Comments	This KPI helps to monitor energy cost by air compressor for single product
Input Data	Period of Measurement: T_1 to T_2

ID EMS0041	Temperature At Fastory
Mathematical Expression	
Notations	Measured by Sensor
Description	
Comments	This KPI depicts Temperature surrounding production system
Input Data	Period of Measurement: T_1 to T_2

ID EMS0042	Energy Consumption By Lights
Mathematical Expression	$$EnCL = \sum EnL[i]_{t1} - \sum EnL[i]_{t2}$$
Notations	$EnCL$: Energy Consumption by Lights $\sum EnL[i]_{t1}$: Sum of recorded Kwh from all lights at start time $\sum EnL[i]_{t2}$: Sum of recorded Kwh from all lights at end time
Description	Energy consumption by all lights in Facilities
Comments	This Kpi depicts energy cost by lighting in facilities of production system.
Input Data	Period of Measurement: T_1 to T_2

ID EMS0043	Energy Consumption By HVAC
Mathematical Expression	$$EnCHVAC = \sum EnHVAC[i]_{t1} - \sum EnHVAC[i]_{t2}$$
Notations	$EnHVAC$: Energy Consumption by HVAC $\sum EnHVAC[i]_{t1}$: Sum of recorded Kwh from HVAC system at start time $\sum EnHVAC[i]_{t2}$: Sum of recorded Kwh from HVAC system at end time
Description	Energy Consumption By HVAC in Facilities of Production System.
Comments	This Kpi depicts HVAC Energy Consumption in facilities of production system.
Input Data	Period of Measurement: T_1 to T_2

ID EMS0044	Title: Total Energy Consumption at FASTORY
Mathematical Expression	$$TEnFastory = W_{TAEC} + EnCAC + EnCL + EnCHVAC$$
Notations	$TEnFastory$: : Total Energy Consumption at FASTORY W_{TAEC} : EMS0010 $EnCAC$: EMS0039 $EnCL$: EMS0042 $EnCHVAC$: EMS0043
Description	Summation of Energy Consumption by Production system, Supporting elements and energy consumption in facilities
Comments	This KPI the ultimate or true energy cost at Fastory for specific time period
Input Data	Period of Measurement: T_1 to T_2

ID EMS0045	Title: Ultimate Energy Consumption per Product at FASTORY
Mathematical Expression	$$UEC = TEnFastory \div PrU$$
Notations	UEC: Ultimate Energy Consumption per Product at FASTORY $TEnFastory$: EMS0044 PrU: EMS0014
Description	Considers true energy consumption based on EMS0044
Comments	This KPI reveals the true energy consumption per product at FASTORY
Input Data	Period of Measurement: T_1 to T_2

ID EMS0046	Title: Ultimate Energy Consumption per Process at FASTORY
Mathematical Expression	$ECpc = TEnFastory \div PsU$
Notations	$UECpr$: Energy Consumption per Process at FASTORY $TEnFastory$: EMS0044 PsU: EMS0013
Description	Considers true energy consumption based on EMS0044 in every process
Comments	This KPI reveals the true energy consumption per process at FASTORY
Input Data	Period of Measurement: T_1 to T_2

ID EMS0047	Title: Energy Cost per Process in Production System scope
Mathematical Expression	$CECpc = \left((W_{TAEC} + EnCAC) \div PsU\right) \times PKw$
Notations	$CECpc$: Energy Cost per Process in Production System scope W_{TAEC} : EMS0010 PsU: EMS0013 PKw: Price of 1Kwh energy
Description	Divide the total calculated Energy and number of processes for specific period and multiply with unit Kwh energy cost.
Comments	This KPI reveals energy consumption cost per process in Production System scope
Input Data	Period of Measurement: T_1 to T_2

ID EMS0048	Title: Energy Cost per Product in Production System scope
Mathematical Expression	$CECpr = \left((W_{TAEC} + EnCAC) \div PrU\right) \times PKw$
Notations	$CECpr$: Energy Cost per Product in Production System scope W_{TAEC} : EMS0010 PsU: EMS0014 PKw: Price of 1Kwh energy
Description	Divide the total calculated Energy and number of products for specific period and multiply with unit Kwh energy cost.
Comments	This KPI reveals energy consumption cost per product in Production System scope
Input Data	Period of Measurement: T_1 to T_2

ID EMS0049	Title: Total Energy Cost in Production System scope
Mathematical Expression	$TCEC = (W_{TAEC} + EnCAC) \times PKw$
Notations	$TCEC$: Energy Cost in Production System scope W_{TAEC} : EMS0010 PKw: Price of 1Kwh energy
Description	Get the total calculated Energy Consumption in production system scope and multiply with unit Kwh energy cost.
Comments	This KPI reveals energy consumption cost in Production System scope for defined period of time.
Input Data	Period of Measurement: T_1 to T_2

ID EMS0050	Title: Total Energy Cost at FASTORY
Mathematical Expression	$TCECfy = TEnFastory \times PKw$
Notations	$TCECfy$:Total Energy Cost at FASTORY $TEnFastory$: EMS0044 PKw: Price of 1Kwh energy
Description	Get the total calculated Energy Consumption at FASTORY and multiply with unit Kwh energy cost.
Comments	This KPI reveals energy consumption cost at FASTORY for defined period of time.
Input Data	Period of Measurement: T_1 to T_2

ID EMS0051	Title: Total Energy Cost per product at FASTORY
Mathematical Expression	$CECpr = TCEC \div PrU$
Notations	$CECpr$:Total Energy Cost per Product at FASTORY $TCEC$: EMS0050 PrU : EMS0014
Description	Get the total calculated Energy Consumption at FASTORY and divide by total production unit.
Comments	This KPI reveals energy consumption cost at FASTORY for defined period of time.
Input Data	Period of Measurement: T_1 to T_2

ID EMS0052	Title: Total CO2 Cost at FASTORY
Mathematical Expression	$TCO2 = TEnFastory \times KCO2$
Notations	$TCO2$: Total CO2 emission at FASTORY $TEnFastory$: EMS0044 KCO2: Coefficient CO2 emission
Description	Get the total calculated Energy Consumption at FASTORY and multiply with Coefficient CO2 emission
Comments	This KPI reveals CO2 emission at FASTORY for defined period of time.
Input Data	Period of Measurement: T_1 to T_2

ID EMS0053	Title: Total CO2 Emission per Product at FASTORY
Mathematical Expression	$TCO2pr = TCO2 \div PrU$
Notations	$TCO2pr$: Total CO2 Emission per Product at FASTORY $TCO2$: Total CO2 emission at FASTORY PrU: EMS0014
Description	Get the total calculated CO2 emission at FASTORY divide by number of produced unit by that time.
Comments	This KPI reveals CO2 emission per product at FASTORY for defined period of time.
Input Data	Period of Measurement: T_1 to T_2

ID EMS0054	Title: Energy Drain Coefficient
Mathematical Expression	$ED = (TIP \div TEnFastory) \times 100$
Notations	ED: Energy Drain Coefficient in percentage $TEnFastory$: Total Energy consumption at Fastory TIP: Total Energy consumption when all Production Line in stop IPC state
Description	It takes the Energy Consumption during IPC Stop status and Total Energy consumption to calculate the percentage of energy consumed in stop status.
Comments	This KPI reveals percentage useful energy consumption in production line
Input Data	Period of Measurement: T_1 to T_2