Design of Midway Energy (Middleware System) as part of Illuminator

Energy System Integration Development Tool Kit

Siva Kaviya Trichy Siva Raman

MSc Graduation Project

TU Delft

DESIGN OF MIDWAY ENERGY (MIDDLEWARE SYSTEM) AS PART OF ILLUMINATOR

ENERGY SYSTEM INTEGRATION DEVELOPMENT TOOL KIT

by

Siva Kaviya Trichy Siva Raman

in partial fulfillment of the requirements for the degree of

Master of Science
in Electrical Engineering

at the Delft University of Technology,
to be defended publicly on August 27, 2020 at 4:00 PM.

Student number:	4784804
Daily Supervisors :	Dr M.(Milos) Cvetkovic, IEPG, TU Delft
	Dr ir A.(Arjen) Van der meer, programme manage, Power web
Thesis committee :	Dr .ir . M. (Marjan) Popov, IEPG, TU Delft
	Dr M. (Milos) Cvetkovic, IEPG, TU Delft
	Dr J. (Jacky) Bourgeois, Internet of Things ,IO
	Dr ir A.(Arjen) Van der meer, programme manager, Power web

TU Delft | PowerWeb Institute | Stichting 3E
Stimulering onderwijs en onderzoek Elektrische Energietechniek

ACKNOWLEDGEMENTS

I am eternally grateful for getting the opportunity to pursue my graduation project in the state-of-the-art Illuminator - Energy System Integration Development Tool Kit project. I am grateful to Dr.ir.Arjen Van der meer for getting me involved with this project and also constantly guiding me. It would not be possible to complete this project without the guidance, continuous support, and invaluable timely feedback of Assist Prof Dr.Milos Cvetkovic and Prof Peter Palensky. I really appreciate the time and effort you dedicated to guiding me throughout the project and I would especially like to thank Professor Marjan Popov for guiding me through the final stages of the project and I would also like to thank Assist Prof. Jacky Bourgeois for agreeing to be part of my thesis committee and evaluating my work.

I am also grateful to the whole illuminator team and my colleagues Sai Medha Subramanian for helping me getting familiarized with the initial setup of Illuminator. Next, I would like to thank Prof. Laurens De Vries and Harsha for sharing some valuable information regarding the Electricity Market Simulation Game (EMSG). I would also like to thank Digvijay Gusian, Luca Barbierato, and Aihui Fu for helping me with some of the valuable information which helped complete this project. I am also grateful to the executive committee of power web institute and the Stitching 3e organization for funding my research. I take this opportunity to thank everyone who helped me with this thesis project. I would also like to extend my gratitude towards the power web secretary Everline and IEPG Secretary Carla for their timely administrative support.

Finally, I would like to thank god and my parents for providing me with the wonderful opportunity of pursuing my dream from the world-class university. A big shout out to all my friends and family who provided me constant support for making my survival in the Netherlands easy for the past two years. Last but not the least, I would like to thank my best friend and partner Shriram Santhanam for encouraging me and being my constant support system for the past two years. These two years of life at delft will be cherished forever and it helped me grow both personally and professionally. Thank you Delft University of Technology for providing me with an amazing study experience.

Siva Kaviya Trichy Siva Raman
Delft, August, 2020

CONTENTS

EXECUTIVE SUMMARY

Using the energy system is part of our daily routine but the complexity of power systems and its understanding have both been confined to the experts in the field. Furthermore, the advent of the Information and Communication Technology (ICT) and the Internet of Things (IoT), power system integration is becoming even more complex for the system planners and operators. Various clean energy technologies like solar power, massive offshore wind, EV as well as the increasing dependency of the Prosumers to move towards a future decentralized and transactive energy market. There is hence a need to educate the general public about the important problems encountered during the energy transition as well as to provide a demystifying version of the existing power system to exhibit its complexity and its benefits of intelligent multi-energy systems.

Energy System Integration Development (ESID) kit could solve this issue.

This project concerns the design and development of a Middleware software system based IOT application, which runs on Raspberry PIs (RasPi), contribute to the layered bottom-top approach of the Illuminator - Energy System Integration Development Tool Kit and has the main goal to help in assisting the application layer to simulate energy system integration scenarios and will be prototyping simple proofs-of-concept for validation purpose. It's realised as part of the master thesis curriculum of the author. On the completion of this project, there will be a generic middleware system that can be used by the end users/application layer to demonstrate Energy System Integration Scenarios, which is scalable, reconfigurable, inter portable.

Illuminator - Energy System Integration Development Tool Kit aimed to become an open-source platform depicting energy integration problems, an education tool kit, and research to prototype control algorithms for counteracting the energy transition challenges. The general public and system integrators would benefit as well as the stakeholders as it can help them analyze the energy integration challenges in a user-centric fashion.

GLOSSARY

Advanced Message Queuing Protocol *"It is an application layer protocol that lets interaction between the client applications and the server. However, it should not be considered a protocol for over the- wire communication; it defines both the network layer protocol and a high-level architecture for message brokers , but it is evolving, so the features are basic.".* 26, 63

Aggregators *"A company or firm that negotiates with producers of a utility service such as electricity on behalf of groups of consumers or prosumers.".* 15

Application layer *"it provides services for an application program to ensure that effective communication with another application program on a network is possible".* 4

Arduino *"single-board microcontrollers or microcontroller kits for constructing digital applications".* 3

Client *"A desktop computer or workstation that is capable of retrieving information,data and applications from a central server".* 3

Co-Simulation *"both the solver, the models are embedded and exported to the importing simulation environment when exchanging dynamic models".* 20, 25, 37, 39, 44, 49, 52–54

Consumers *"Someone who buys things for daily consumptions".* 15

Cyber Security *"It is the practice of defending computers, servers, mobile devices, electronic systems, networks, and data from malicious attacks".* 15, 46, 47, 52, 55

Day Ahead *"It takes into account the prediction for next day".* 35, 39, 40

Decentralized *"Distribution of administrative power and activities to a less concentrated area from central authority(centralized)".* 2

Energy Sim *"Co-Simulation based interpretation in python".*. 22, 25, 44, 48, 61, 63

Functional Mock UP Units *"The Functional Mock-up Interface (FMI) is a free standard that defines a container and an interface to exchange dynamic models using a combination of XML files, binaries and C code zipped into a single file".* 20, 25, 37, 39, 44, 45, 49, 52–54, 63

Hardware in the loop *"it is a technique, which is used in the development and testing of the real time complex embedded systems".* 3

Hive MQ *" is a MQTT protocol based messaging platform designed for the fast, efficient and reliable movement of data to and from connected IoT devices. It uses the MQTT protocol for instant, bi-directional push of data between your devices".* 48, 55

Illuminator - Energy System Integration Development Tool Kit *"Illuminator is a development kit for demonstrating the challenges of the energy transition and the benefits of intelligent multi-energy systems".* 1, 6, 13–16, 20, 27, 33, 44–46, 48, 49, 52, 54, 55, 63

Internet of Things *"this is the concept of basically connecting any device with an on and off switch to the Internet (and/or to each other). This includes everything from cellphones, coffee makers, washing machines, headphones, lamps, wearable devices and almost anything else you can think of".* 3

Interoperability *"It is the ability to exchange information between different application platforms".* 20

Intra Day *"It will take into account the prediction for next 1 hour or a certain time in the same day".* 35, 39, 40

JSON *"JSON (JavaScript Object Notation) is a lightweight data-exchange format. It is easy for humans & computers to read , write and interpret"*. 21, 28, 53

Key Performance Indicators *"refer to a set of quantifiable measurements used to gauge a company's overall long-term performance"*. vi, xiii, 44, 45, 47, 48, 54, 55

Message Queuing Telemetry Transport *"It is a inter process communication protocol based on MOM, it is mostly realised in embedded iot systems, where a physical broker is required. it has low latency, dynamic brokerage and resilient feature of exchanging messages. It can also be used as distributed by clustering message broker. it can be realised in all the coding languages."*. x, xv, 13, 26–28, 44, 45, 47, 48, 52, 54, 55, 61, 63

Middleware *"it is a software that interacts between an operating system and the application layer. Essentially, hidden as translation layer, it is mainly responsible for facilitating communication and data management in distributed application environments"*. vii, 1, 3, 14, 16, 19, 22, 27, 28, 33, 41, 44–48, 63

Open Modelica *"it is an open-source Modelica language-based modeling and simulation environment intended for industrial and academic usage"*. 25, 37, 39, 63

Paho MQTT *"It is open source python based () clients, which is used in exchanging information between different components"*. 22, 28, 44, 48, 61

Panda Power *"which is a open source power system modelling python package"*. 22, 25, 44, 48, 52, 61, 63

Peak shaving Mechanism *"It is a mechanism mostly commonly used in BESS for cutting down peak demand charges on the electricity/utility bill"*. 15

Power Factory *"It is a leading power system analysis software application for use in analysing generation, transmission, distribution and industrial systems"*. 25

Prosumers *"A energy consumer who can produce and consume energy , actively take part in energy markets"*. vii, 2, 15

Raspberry Pi *"it is a credit-card sized mini computer, which is low in cost, can be used through putty terminal or interfacing with monitor,wireless keyboard and mouse.It is designed for exploring the programming languages & increasing computing skills in languages likes Scratch and Python"*. 3

Server *"A computer or physical component or computer program which establish and manage entry to a centralized Service in a network"*. 3

Swimlane Diagram *"It is a type of flowchart that delineates the whole process. Using the metaphor of lanes in a pool, a swimlane diagram provides clarity and accountability by placing process steps within the horizontal or vertical “swimlanes” of a particular employee, work group or department"*. 22

Transactive Energy *"it refers to the common techniques of economic and control, that is used to manage the exchange of energy or flow inside an existing electric power system and regulation of the market based standard values of energy"*. 2

Zero MQ *"is an opinionated, light weight, blazing fast messaging asynchronous library used for distributed and concurrent applications.* 48, 55

Zero MQ Transport Protocol *"It is a web socket based protocol which uses peer-to-peer messaging architecture, it can be well suited for distributed systems but message routing is not possible, but it can also work with broker to enable the dynamic leverage and message routing features of the message broker. it also has low latency , high through output and available in all the coding languages"*. 26, 27, 63

ACRONYMS

API Application Programming interfaces. 5, 13, 21–27, 39

BESS Battery Energy Storage System. 33, 34, 46

CORBA The Common Object Request Broker Architecture. 25

CSV Comma Separated values. 23, 28, 29

DB Data Base. 20

DERs Distributed Energy Resources. 2, 35

ESI Energy System Integration. xv, 2, 6, 9, 15, 16, 20, 21, 25, 28, 34, 37, 41, 44–47, 49, 52–55, 63

ESID Energy System Integration Development Tool Kit. 2, 3, 5, 34

EV Electric Vehicles. 4, 15, 20, 46, 52

HIL Hardware in the loop. 3

ICT Information and Communication Technology. vii, 15, 20, 44, 48, 53

IEEE Institute of Electrical and Electronics Engineers. 13, 25, 29, 45, 48

IEPG Intelligent Electrical Power Grids. 7

IOT Internet of things. vii, 3, 13, 15, 20, 44, 53

IP Internet Protocol. 13, 27

LAN Local Area Network. 55, 61

LV Low Voltage. 33, 35, 40, 41, 44, 48, 63

MOM Message Oriented Middleware. 25–27, 47, 63

OMG Object Management Group. 25

OS Operating System. 3, 7, 21, 44, 54

PV Photo Voltaic Panel. 2, 4, 15, 20, 46, 52

QOS Quality of Service. 28, 54

RasPi Raspberry PI. xv, 13–17, 20, 21, 23, 27, 34, 37, 44, 46, 47, 49, 52, 54, 55, 63

RES Renewable Energy Sources. 2, 15, 35

REST Representational state transfer. 13

SD Secure Digital. 3

SQL Structured Query Language. 20

WIFI Wireless Fidelity. 3, 61

YAML YAML isn't a markup language. 28, 29, 37, 39, 44

List of Tables

List of Figures

1

INTRODUCTION

This Chapter encapsulates the energy transition and its related challenges, energy system integration, and its importance in the energy transition is the motivation behind this graduation project. It also introduces the Illuminator - Energy System Integration Development Tool Kit - Energy System Integration Development Tool Kit and its tabletop layered approach and defining the main scope of this graduation project that is designing and development of a Middleware system as part of Illuminator. Moreover, Thesis contributions, Methodology, Thesis outline, Stakeholders involved and Illuminator team is also described detail in this part.

1.1. Background and Motivation

1.1.1. Energy Transition

"The energy transition is a pathway toward the transformation of the global energy sector from fossil-based to zero-carbon by the second half of this century", it plays a crucial role in climate change as it aims to reduce the energy-related CO_2 emissions. The key drivers of the energy transition are the increasing penetration of RES into the traditional grid and improvements in energy storage technology.[1]

Some of the countries which are pioneer in energy transition are Germany, the United Kingdom, Sweden, Spain, Italy. Germany leads the world with 12.74% of renewable energy consumption in their total energy use. This European nation is on an energy revolution by 2050, aimed at replacing its fossil fuels with wind and solar technologies. [2]

Continuous improvements in energy efficiency, the development of a quality market for energy, a continuous increase in the use of renewable energy sources, improved energy management, continuous technological progress, and continuous improvements in citizens' and the economy's education and awareness of examples of good practices need to be achieved as part of the energy transition, which makes it extremely challenging.[3]

But, the main challenge is to make a shift towards carbon-neutral energy scenario's without compromising grid reliability, stability and dynamics .[4]

1.1.2. Energy System Integration

Energy system integration is a vital milestone in achieving successful clean energy transition. Which can be enabled by the integration of the state-of-art information technology, smart technology, policy frameworks and market instruments, distributed energy resources(Distributed Energy Resources) as well as engaging in decentralized transactive energy scenarios.

As, we move forward in Decentralized and Transactive Energy scenarios there is an increasing involvement of Prosumers who can take the role of both electricity producers and consumer by the use of their own distributed energy resources(DERs) such as photovoltaic panel(PV) or Wind farms. In this way, Local power micro-grids are formed with locally-controlled power management and market mechanisms. When such micro-grids are integrated into the existing power system it can help in improvising the stability and resilience of the same by minimising losses in the form of transportation and energy conversions.

Figure 1.1: Illustration of Power Engineer Explaining a ESI scenario to municipal people

Prosumers proactively engages in energy system integration and energy transitions, There is hence a need to educate the general public about the predominant problems encountered during the energy transition as well as to provide a demystifying version of the existing power system to exhibit its complexity and its benefits of intelligent multi-energy systems.[5].[6]

Energy System Integration Development Tool Kit(ESID) could be used to help educate policymakers, students as well as the general public. one such demonstrator is the Illuminator.

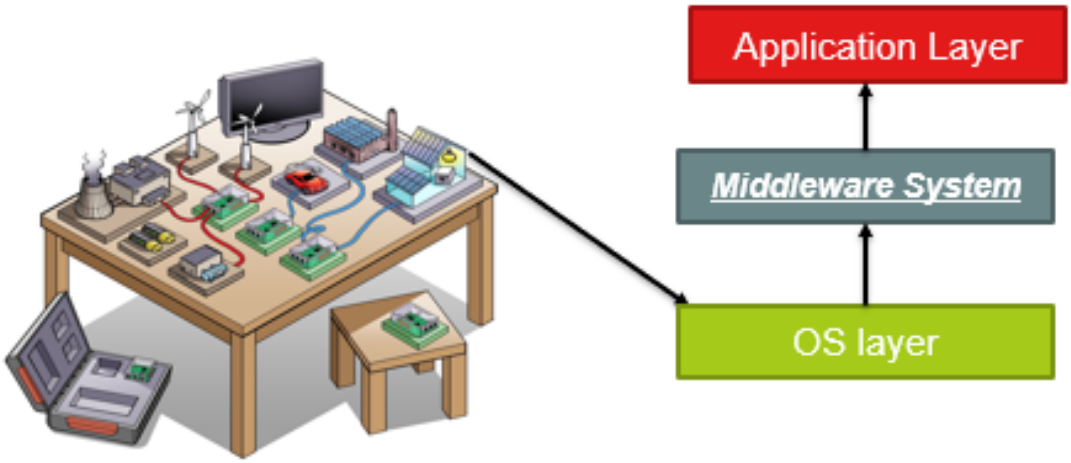


Figure 1.2: Depiction of Bottom Top Approach of Illuminator

1.2. Illuminator

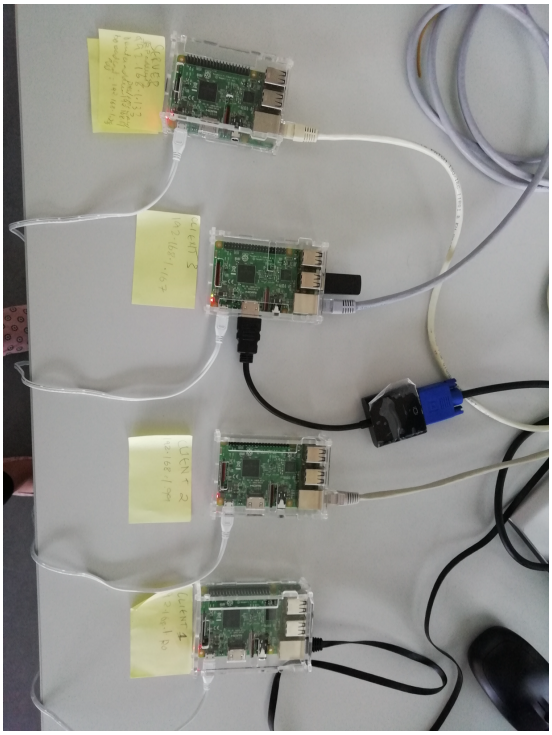
"Illuminator is a development kit for demonstrating the challenges of the energy transition and the benefits of intelligent multi-energy systems" [7] This tool kit is based on the Internet of Things(IOT), Hardware in the loop(HIL), inter process communication protocol and uses state of the art, information technology solutions & multi-disciplinary approach.[8][9]
Energy components which are used for generating, carrying, storing and consuming are modelled and visualized in tabletop level and intelligently programmed and controlled by mini-computer(Raspberry Pi), the tool kit's main goal is to simulate energy system integration scenarios and will be prototyping various proofs-of-concept for demonstration purposes.
Right now,4 raspberry pi's are used to represent the components of the distribution side of the grid and will provide an education tool kit for study purposes. [10]
As from the figure 1.2, a layered approach is followed in building and deployment of the Energy System Integration Development Tool Kit (ESID), starting with the development of software application with OS Layer, Middleware system and application layer, interfacing with hardware components and visualizing energy system integration scenarios.

1.2.1. Operating System(OS)

Raspberry Pi's are considered to be mini computers and has more computational capabilities than Arduino (microcontroller motherboard) [11]. As the Raspberry pi is a mini-computer, booting up of the raspberry pi is done through a local network booting or pi server [12] [13].
Where one raspberry pi acts as the pi server and has the SD card booting enabled whereas other 3 client raspberry pi's boot via the Server raspberry pi by local network booting, this is done to avoid SD card booting failure which was more frequent in the SD card booting method [14].
All the Raspberry pi has been loaded with Debian buster 10 & has python 3.7 installed by default and the Clients have a shared folder in the main pi server and they are connected to the WIFI router to enable local network booting.

1.2.2. Middleware System

Middleware is one of the key requirements for the design of the illuminator as it acts a software that bridges the gap between an operating system or database to applications over a network, also enabling the various components of a distributed system to communicate and manage data [15]. it might include interprocess communication protocols, power system modelling and will be helping the end-user in effectively building energy system integration applications.



(a) Illustration of booting of the raspberry pi's

(b) Raspberry PI settings

Figure 1.3: Illustration of current setup of Illuminator

1.2.3. Application Layer

Application layer [16] of the illuminator would include power and energy sector algorithms, models, visualization tools and user interactions to illustrate the impact of the energy transition and energy integration scenarios to a wider audience. some of the illustrations of illuminator might include

1. Integration of EVs in the distribution grid to demonstrate the impact of the adoption of electric transportation on the demand-side congestion
2. Depiction of electrolyzers for hydrogen production, where we could demonstrate their impact on balancing volatile renewable power
3. Demonstration of the impact of cyber attacks on the power system.[17]

1.2.4. Hardware Layer

It would consist of low power consuming energy components which can be emulated as power system components like Photovoltaics (PV panel), wind farms, Electric Vehicles(EV), Heat Pumps and also configured into a high-fidelity scaled-down version of the real-world grid.

1.3. Thesis Contributions

The main aim of this graduation project is :

- To build a middleware system for the Illuminator (Energy system integration development toolkit) for use by power system application engineers
- Prototyping a very simple proof of concepts for demonstration purpose and validating it

Based on the goal of this thesis, the key research questions which are intended to be answered at the end of this thesis are:

1.3.1. R01

What are the important design considerations to be considered while developing the middleware system as a distributed Network?

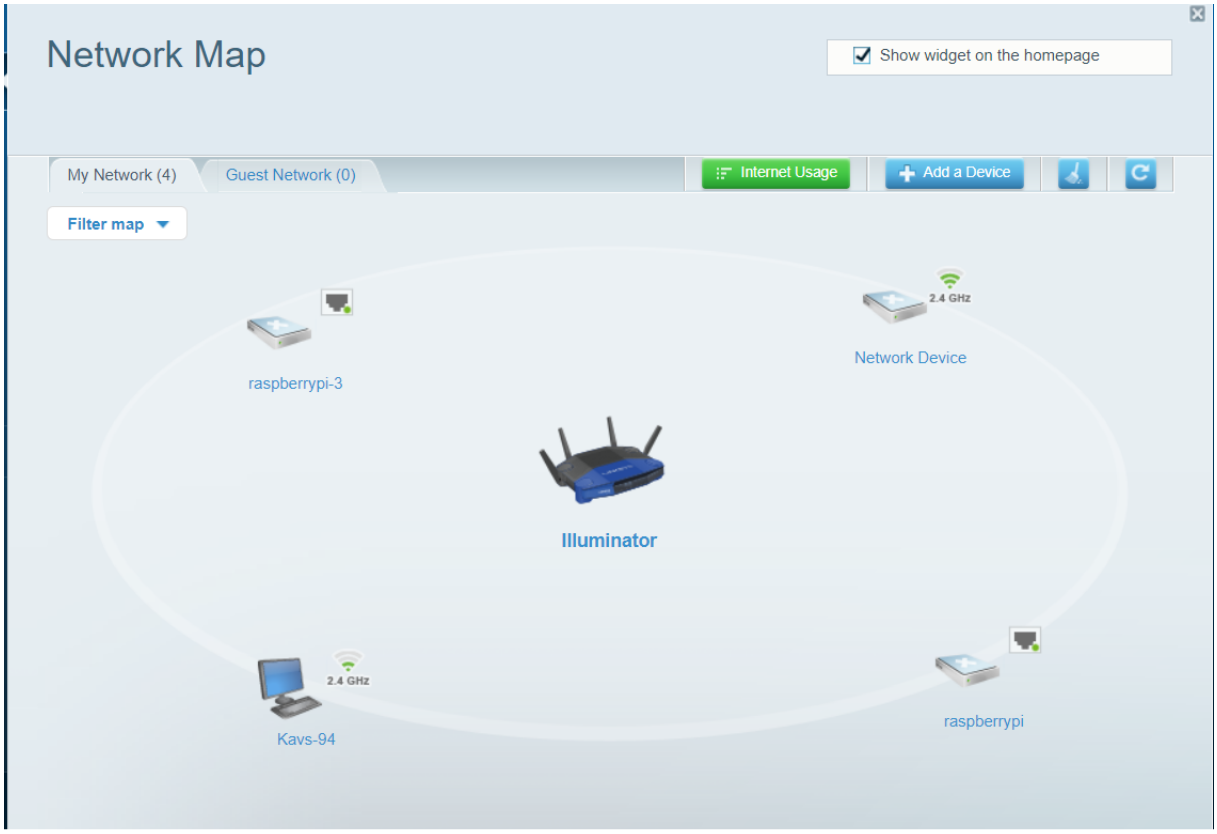


Figure 1.4: Illustration of Net booting configuration of the raspberry pi's with the router

1.3.2. R02

What are the different software architectures possible for middleware system?

1.3.3. R03

What are the measurable key performance indicators to be considered for the assessment of the middleware system?

1.4. Methodology

The research questions are answered with a scientific approach as follows:

1.4.1. Literature Survey

During this phase previous existing demonstrators are analysed for their advantages and disadvantages, learning of which would be used in implementing the ESID, Energy system integration scenario's requirements are analysed, design considerations and system-level requirements are defined

1. What are the existing demonstrators, what can be learnt from them and their drawbacks?
2. What should be considered for system-level requirements?
3. What are the design consideration to be taken into account and associated challenges?

1.4.2. Design Analysis

This phase involves defining the middleware functionalities and API Calls, it also takes us in-depth into the selection of different architecture for the specific middleware services.

4. What are the different middleware configurations?
5. Which is the best-suited message exchange format?

6. what are the available open-source software for building middleware?
7. Which are key functionalities of the middleware?
8. How to integrate steady-state/dynamic models of the power system into middleware?

1.4.3. Development, Implementation and validations

Very simple use cases are defined, implemented and validated from an end-user perspective.

9. is the results of the distributed system setup validated with the single panda power simulation?
10. is the middleware system reconfigurable? How?
11. is the Communication protocol resilience enough? How?
12. is there a necessity for security level in the communication layer? Why? Why not?

1.4.4. Results

Both the positive and negative test case scenarios of the middleware system have been validated.

1.5. Scope of the Graduation Thesis

This graduation thesis concerns the first level development of middleware system which can cater to most of the Energy System Integration (ESI) scenarios, especially in the grounds of power system modelling, communication infrastructure and a basic level of integration of steady-state/dynamic models of the power system models and prototyping a very simple case study of Energy System Integration (ESI) scenarios and validating it.

1.6. Thesis Outline

The graduation report consists of 6 chapters.

1. **Chapter 1** gives a brief overview of the important aspects related to this thesis such as the background, motivation related to this graduation thesis. Moreover Illuminator - Energy System Integration Development Tool Kit and its tabletop layered approach is introduced, it encapsulates the thesis contributions, scope, stakeholders and illuminator team.
2. **Chapter 2** reviews the literature about few existing demonstrators, their learning's and drawbacks, and also briefly review the need for illuminator and also important literature used in the selection of the design considerations and system-level requirements.
3. **Chapter 3** describes the design analysis involved in designing the middleware system, it describes the middleware functionalities, different possible architecture and which is best suited for this graduation project.
4. **Chapter 4** details on the case study of demand-side congestion management with peak shaving mechanism and its implementation & results from an end-user perspective, validation of same with a single python script.
5. **Chapter 5** deals with the evaluation and discussions about the middleware systems based on design considerations, key performance indicators and positive, negative test cases of the middleware system.
6. **Chapter 6** reflects upon the conclusions, gives a futuristic overview of the Illuminator and recommendation for future works.

1.7. Illuminator Team & External Contributions

- Energy System Integration development toolkit idea was proposed and maintained by Power web institute and Intelligent Electrical Power Grids (IEPG) research group under the guidance of Dr Milos Cvetkovic, Dr Ir. Arjen van der meer and Professor Dr Peter Palansky. [7] [10]
- A group of 6 B.sc students have worked on the dashboard and simulating energy components of Illuminator.
- My colleagues Sai Medha Subramanian, initially configured the illuminator OS layer during summer 2019 and also helped me in getting familiarized with the initial setup of the illuminator.

1.8. Stakeholders

Illuminator project is co-sponsored by Stichting 3e, Power web Institute and Intelligent Electrical Power Grids. [7] [10].

(a) Intelligent Electrical Power Grids

[18]

(b) TU Delft Power Web Institute

[19]

(c) Stitching 3E

[20]

Figure 1.5: List of Stakeholders involved in Illuminator

2

LITERATURE REVIEW

This chapter reviews the existing demonstrators addressing the energy transition scenarios. Existing demonstrators are further divided into the application of game theory to the power grids and real-time simulator-demonstrators. Furthermore, the important design considerations, Energy System Integration (ESI) scenarios, and system requirements are reviewed in-depth in this chapter. Thus, this chapter encapsulates all the information retrieved from the literature research done during this graduation project.

2.1. Existing Demonstrators

There are quite a few numbers of demonstrators already addressing the energy system integration/transition issues. Concerning this thesis few of the demonstrators are analysed and reviewed, some of the aspects of the demonstrators are also used for the implementation of this thesis. The Existing Demonstrators are classified as follows:

1. Application of Game Theory to Power Grids.
2. Real-Time Simulator-Demonstrators available.

2.1.1. Application of Game Theory to Power Grids

*" **Game Theory** is the process of modelling the strategic interaction between two or more players in a situation containing set rules and outcomes. While used in several disciplines, game theory is most notably used as a tool within the study of economics" [21].*

*"**Nash equilibrium** is a concept of game theory where the optimal outcome of a game is one where no player has an incentive to deviate from his chosen strategy after considering an opponent's choice"*[22]

Recent research shows that the game theory and Nash equilibrium can be effectively used for the applications in power grids. Game theory is exploited to be used in case of supply-demand scenarios, where there is an issue of non-renewable resources and how to accommodate renewable alternatives such as solar and wind. Nash Equations are used to attain the maximum benefit of the any given power grid solutions.[23]

These applications can be used to assess, analyse the critical solutions based on the end user's decision-making capabilities as well as demonstrate the adversity of a bad decision on the power grids. Few of its application are discussed further.

2.1.1.1. Electricity Market Simulation Game (EMSG)

which replicates the Dutch energy system generation market biding after the era of European energy liberation [24].It is a closed source web-based gaming simulation used in an electricity and gas market course of TU Delft and it also outsourced to other universities. It is used as an educational tool for encouraging students, market analysts, policymakers, researchers and companies in the power sector to explore biding algorithm and get an overview of short term and long term dynamics of electricity and carbon markets, analyse and understand the impacts of several policies in such place for energy systems. it is supported in the back end with a Java-based web application, which the students can interact with. [25]

Electricity Market Simulation Game was the pioneer in simulating and developing a market biding game and is maintained by TU Delft TPM Faculty and Indian based non-profit organisation Fields of view. Players assume different roles and try to understand the market complexity in terms of decision making, the human interaction with the power system which is difficult to experience, thereby leading to some trade-offs, irrationality and any other intangible, these are the main centre of attraction of this game. But, they are not at par with the current energy transition scenarios.

2.1.1.2. The Universal Smart Energy Framework(USEF) documentary game

Balance of Power is an interactive film-based on smart energy future and depicting the flexibility of demand-side management. It is also a decision- making and evokes the awareness of various roles and implications of different decisions on the energy system.

It is also fun to play and it has won the energy institute award, It takes the game based on the decision we as an end-user make, therefore every decision would be crucial and interesting. There are few characters associated with this game, the end-user as a journalist and its limelight is on the very grounding breaking real dilemmas about the future electricity systems. They are available as open-source and available for all. But, they are just web-based documentary game, there are not many technical details involved and it is easily accessible and understandable by the general public. .[26]

2.1.1.3. Energy Transition game

Energy Transition game is a web-based fact energy model, helping users to get their firsthand experience of energy transition scenarios for next few years and also helps them in understanding the need of the energy

Figure 2.1: Logo of the Balance of Power USEF Game

[27]

transition, the complexity involved. It is developed for a broader audience starting from school students to people in academia to policymakers, governmental organization and NGO's. The game is available in three different mode namely :

1. Mobile mode
   This mode of the game is specially created for testing the knowledge of the end-users.

2. Energy transition Model
   This model of the game is confined to the policymakers and experts in the field. It is based on energy scenarios of six different EU Countries.

3. Energy Game
   This aspect of the game is used for enlightening the general public and it is also fun to play with. [28]

Even though, the game is based on state-of-the-art technologies and can be used by the general public. it is not backed by a solid visualization.

2.1.2. Real Time Simulator- Demonstrators

These applications are used to model/simulate physical systems in a computerized environment, with accurate time execution similar to actual physical systems.[29]

These real-time simulators make use of both hardware, software architectures and give a visualization affects of the actual power systems, it helps in emulating the power system components in a big scale. Some of the existing real-time simulators are discussed in the upcoming sections

2.1.2.1. The Smart grid demonstrator

The "Smart-Grid-Demonstrator" is a flexible manifest, created to introduce the state-of-the-art topic of smart grids and the concept of linking the energy supply even to youngsters and amateurs. It is created by the Fraunhofer AST.

It not only offers the state-of-the-art grid topology for various voltage levels from traditional grids, but it also concentrates on the different storage technologies such as redox-flow battery and future centric developments on smart metering, demand-side management. They offer a commercial interactive software solution, the user can manage the entire energy infrastructure via a touch screen application & visualize it with a hardware demonstration, thereby getting their hands-on energy system complexities. [30]. Even though they are using

(a) Snapshot of current Dutch Energy Transition Model based game

(b) Snapshot of mobile game for energy transition model

(c) Snapshot of mobile game for energy game

Figure 2.2: Snapshot of the three different modes of the energy transition model game [28]

the state-of-the-art technology for demonstration, the exhibit is restricted to closed commercial solutions and also for the experts in the field of energy and only available for display in the exhibitions. [30]

2.1.2.2. The Heat simulation game

Heat is a Three dimension design tool, which is realistic, it is aimed for helping the stakeholders design and develop heat network, which is developed by Tygros and Alliander for residential areas. The Heat offers various actors of the heat network such as the producers, customers, governmental organizations and other stakeholders to manage their heat network from their virtual world and also base their decisions and experiments based on current data. [31]

Heat, thus offer an opportunity to install and maintain a heat network while considering all financial constraints and ambitions of all the stakeholder. It helps in the accelerating and improving complex decision-making processes and guide us towards sustainable future.[32] But, Heat is a closed commercial solution avail-

able only for the experts in the domain.

2.1.2.3. Embedded Simulation Toolkit for power distribution systems

Embedded simulation toolkit is realized for real-time simulations on a Raspberry PI (RasPi) 2 with support from the technological advancement associated with mobile devices. It is developed to become an efficiently small, flexible and reliable, easily affordable device. Because of the mobile-based platform, both software and hardware design are compatible to be installed in various locations of laboratories and field locations.

The tool kit was tested for real-time simulations on IEEE 37 bus system feeder emulator, some large scale applications like IEEE 8500 node test feeder and the results were quite promising in terms of accuracy and on-time performance. It is used for improvising the computational power for aiding in simulations, evaluating the real-time smart grid applications. One of the major limitations of this model is that it is operated on non-real-time operating systems.[33]

Even though the toolkit is realized for power systems, it is just to tackle the optimal power flow solution and improvise the computational capability.

2.1.2.4. Smart Metering Infrastructure for distribution grid

Smart Metering Infrastructure(Flexmeter) is designed to promote automation and simple management of distribution grids. It is established with a help of IOT device and microservices-based cloud architecture aimed to be the scalable, interoperable and flexible solution. It has utilised web services and API to access and integrate open-source data using standard web data formats. It is also used for consumer awareness program, demand response etc. The proposed system is dealt with three different layer namely as from figure 2.3

1 Device Integration Layer
It is used for retrieving information regarding measurements from several heterogeneous devices and allows the use of various communication protocols like IEEE 802.11, Zigbee.

2 Middleware layer
It is established to authorize bidirectional communications with the help of (Message Queuing Telemetry Transport), manage and store measurement values using REST web services and also help in controlling the commands sent to various devices.

3 Application Layer
This layer is specifically designed for offering tools and related API for designing and maintaining the distributed applications, for catering to the needs of the stakeholders involved in smart grid scenarios. [33]

It is a promising solution for the evolving smart grids , still it is targeted at a commercial scale and the architecture is a closed real time solution.

2.1.2.5. Smart Power Flow Monitoring and Controlling

This Demonstrator is realised on a Raspberry PI (RasPi) it is very similar to Illuminator - Energy System Integration Development Tool Kit, power monitoring and control system is a fully automated re configurable, end user friendly solution for a trust able energy management.

It is aimed at measuring and controlling the voltage, power and current levels of home applications when the parameters exceed a certain threshold. The control action involves tripping the relay and a notification would be sent to the controller. The actions and current status of the load parameters can be monitored with the IP address of the Raspberry PI (RasPi). [34]

It is developed to minimise power wastage by using sensors and detecting the presence of humans. Even though the demonstrator has various leverage over the above-mentioned demonstrators, it is used for a small scale and limited to household applications and serves a single purpose.

2.2. Learning's from Existing Demonstrators

Most of the demonstrator listed above is focused on the decision-making capabilities of the humans in the energy system integration and energy transition. Few of them are used for real-time simulation purposes, which

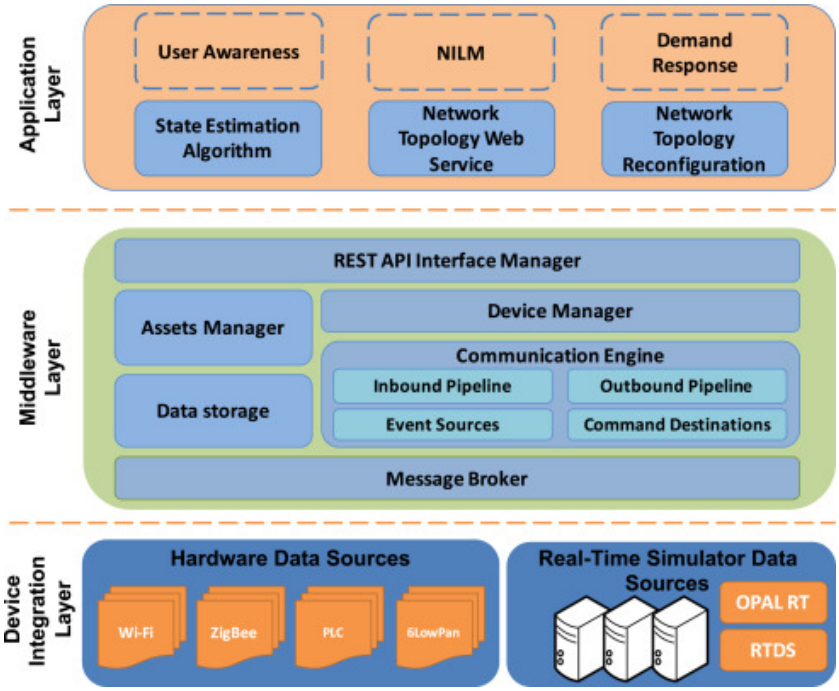


Figure 2.3: Demonstration of various layer involved in flexmeter

[33]

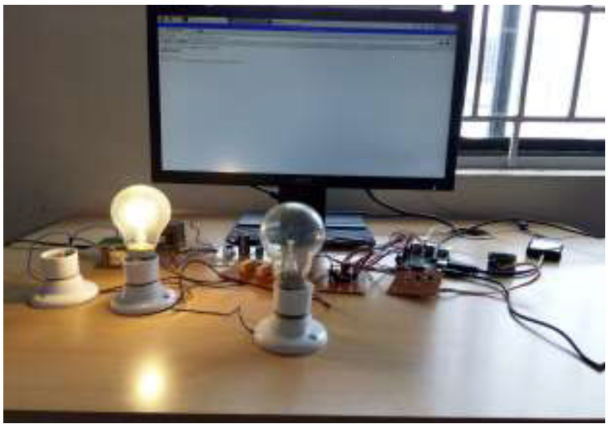


Figure 2.4: Snapshot of Power Monitoring and control system

[34]

helps in better visualization of the power system in the grounds of energy system integration and transition.

Two of demonstrators namely the smart power flow control & monitoring and Embedded toolkit uses Raspberry PI (RasPi)'s for their computation and illustrations. Flexmeter is also realised in the same layered manner as that of the illuminator and the message-oriented middleware approach could be used for implementing Middleware system of Illuminator - Energy System Integration Development Tool Kit.

2.3. Need for Illuminator

Most of the above-mentioned demonstrators are a commercial, closed solution, or web page-based simulation games. Moreover, users/players need to have a relatively high entry-level (i.e., knowing how a power system and its associated energy market works).

Therefore, there is a need for a portable, open-source, and easily configurable demonstrator which can be demystifying the power grid.

2.4. Energy System Integration Scenarios

Energy Systems Integration's (ESI) underlying principle lies in the coordination, integration of energy generation at local, regional and national levels. These aspects involve all the stages from energy production to conversion to delivery to the end-users. The base of such is exploiting the potential cost-effective way to achieve carbon-neutral energy and makes RES a more reliable, sustainable and resilient solution. It lays out a platform in which interconnected and integrated energy ecosystem opportunities can be made into reality. [35]

Most of the Energy System Integration (ESI) scenarios revolve around the integration of RES into the existing grids, challenges involves predicting their availability or how to overcome overproduction or consumption of energy, how to make use of the storage resources.

Energy System Integration (ESI) takes a whole new holistic view of the electricity, gas, and heat markets towards delivering a clean, reliable, and affordable energy system [36] Few of the energy system integration scenarios include :

1. Demand side congestion management of distribution grids by Peak shaving Mechanism mechanism.
2. Integration of PV or EV or wind farms into the existing traditional grids
3. Exploitation of flexibility from both Aggregators , Prosumers and Consumers
4. Evaluating and protecting of the grid from Cyber Security attacks.
5. Integration of ICT & IOT technologies into the power grids to make it more reliable and resilient.[37]

2.5. System Requirements of Middleware

Based on the present energy integration scenarios and need for integrating the ICT & IOT technologies and also keeping in my mind the need for real time simulators and the complexity, necessary services needed for a power system engineer to model or demonstrate the energy system integration challenges and benefits.
The following middleware system requirements have been decided based on the same.

Power System Modelling software or tool should be added to the middleware system to facilitate the emulation of power system components as it is going to be a ESI development tool kit, there are huge requirements for modelling the traditional power system components.

Various model of energy system integration components - battery, Solar panel, EV, Heat Pumps and generators should be incorporated to realise the real-time effects of those components and also they are the upcoming state-of-the-art models which are quite new to the power system modelling environment, it is needed to realise the renewable energy integration's and to demonstrate the challenges and benefits of the same.

Use state-of-the-art technology for communication facility across different components, thereby enhancing coordination and communication through ICT services. This particular technology would be useful in instrumenting communication between different Raspberry PI (RasPi)'s and would be the most important requirement for the middleware system.

Selection of the software, tools should be compatible to handle the anticipated increase of Raspberry PI's in the future (real-time) as shown in figure 2.5,2.6, 2.7 . This is a very important aspect during the scaling of the Illuminator - Energy System Integration Development Tool Kit and also in handling the related complexities.

Communication service should be resilient, reliable, inter portable, easy to identify the fault. This constraint is important as to the identification of faults in the communication services, as it is the backbone of the entire middleware system, therefore it needs to be stable and reliable in long term. Middleware system should be reconfigurable, expandable to accommodate power system activities and to be flexible towards energy system integration scenarios.

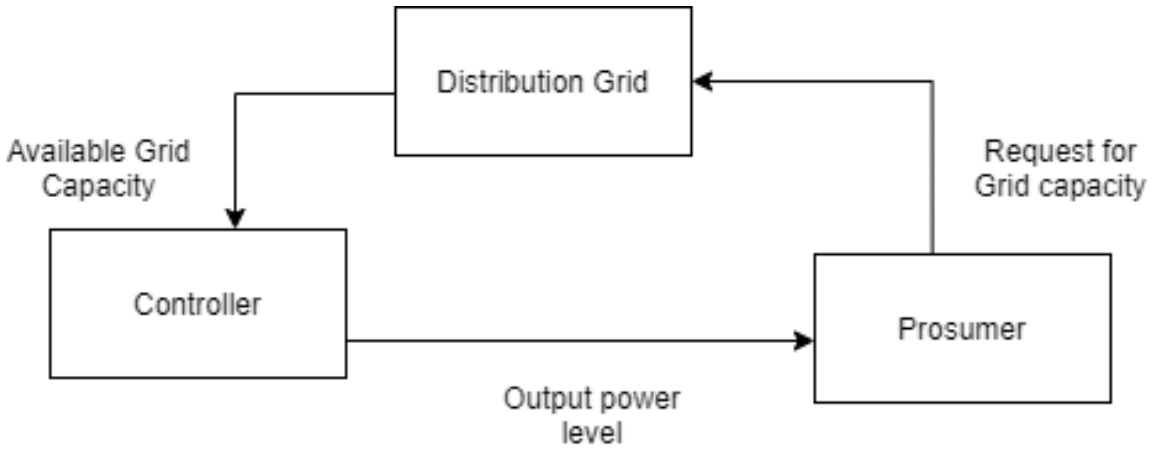


Figure 2.5: Illustration of simple distribution grid, where each rectangle represent a Raspberry PI

2.6. Middleware Design Criteria

Both the system requirements and design criteria should go hand in hand, and design criteria would be constant during different phases of the development of the Illuminator - Energy System Integration Development Tool Kit. Therefore defining the design criteria was taken with at most care and to cater to all the possible design requirements related to energy system integration scenarios.

Some of the design considerations are selected based on the nature and possibility of future expansion of the Illuminator - Energy System Integration Development Tool Kit project as described below.

All the software should be implemented in python and should be available as open-source for easy accessibility and integration. Python is chosen because by default Raspberry PI's has python software installed in them and python is the state-of-the-art nowadays, scripts can be automated in the future. An open-source version of the software is mandatory because Illuminator - Energy System Integration Development Tool Kit is going to be first of its kind and the illuminator team wants to make it available for the researcher to actively download and improve upon for energy transition activities and to reach the masses.

Illuminator - Energy System Integration Development Tool Kit in general would be a distributed setup so that it can be easily scalable, expandable in real-time mode and can still be portable. All the developed software and hardware should be compatible to work in Raspberry PI (RasPi), as the Illuminator - Energy System Integration Development Tool Kit is instrumented by a set of RasPi's. The developed model of middleware system should be easily scalable, reconfigurable for future perspective and also for adding some more ESI scenarios in the future. The developed version of the middleware system and also the Illuminator - Energy System Integration Development Tool Kit should be made flexible toward energy system integration scenarios as it is specifically developed for catering the same.

Middleware system should be user friendly and helps the end-user in easily realising Energy System Integration (ESI) scenarios. This criteria is very important because the middleware should be generic but makes the work of end-users simple, It is emphasised because all the application layer activities already involve some complex design and development of algorithms based on machine learning's or tackling the challenges related to energy system integrations.

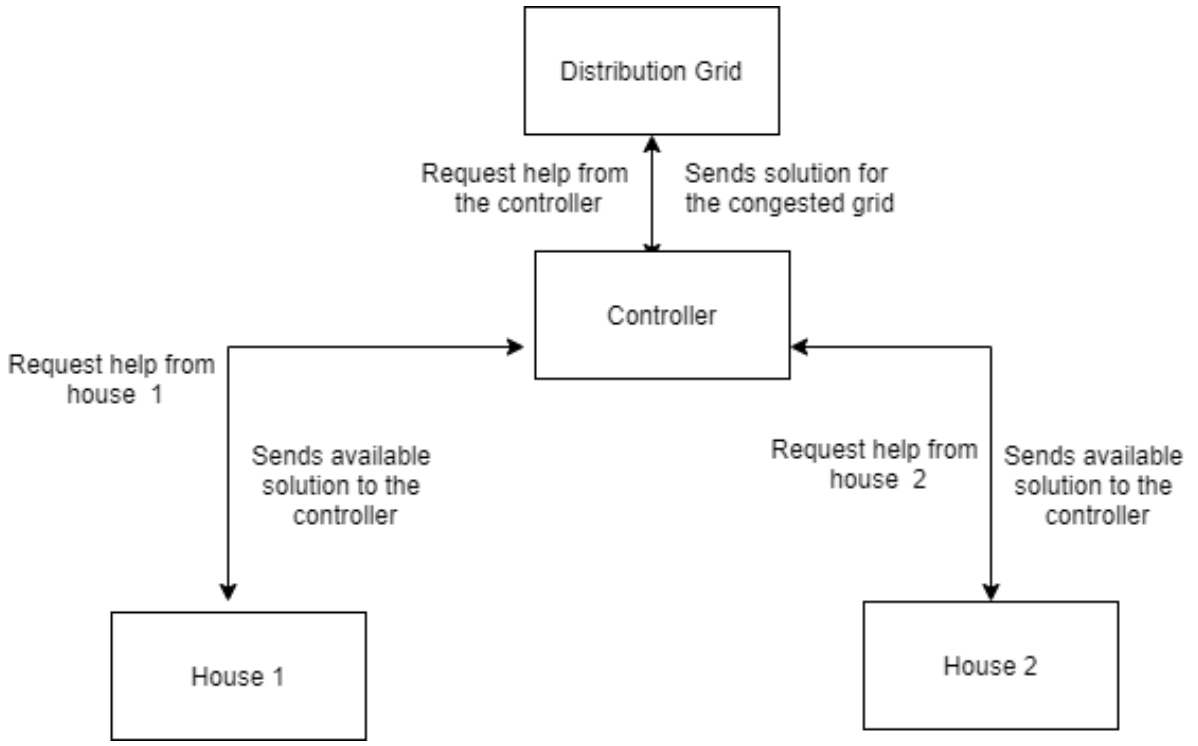


Figure 2.6: Illustration of a town with two houses, where each rectangle represent a Raspberry PI

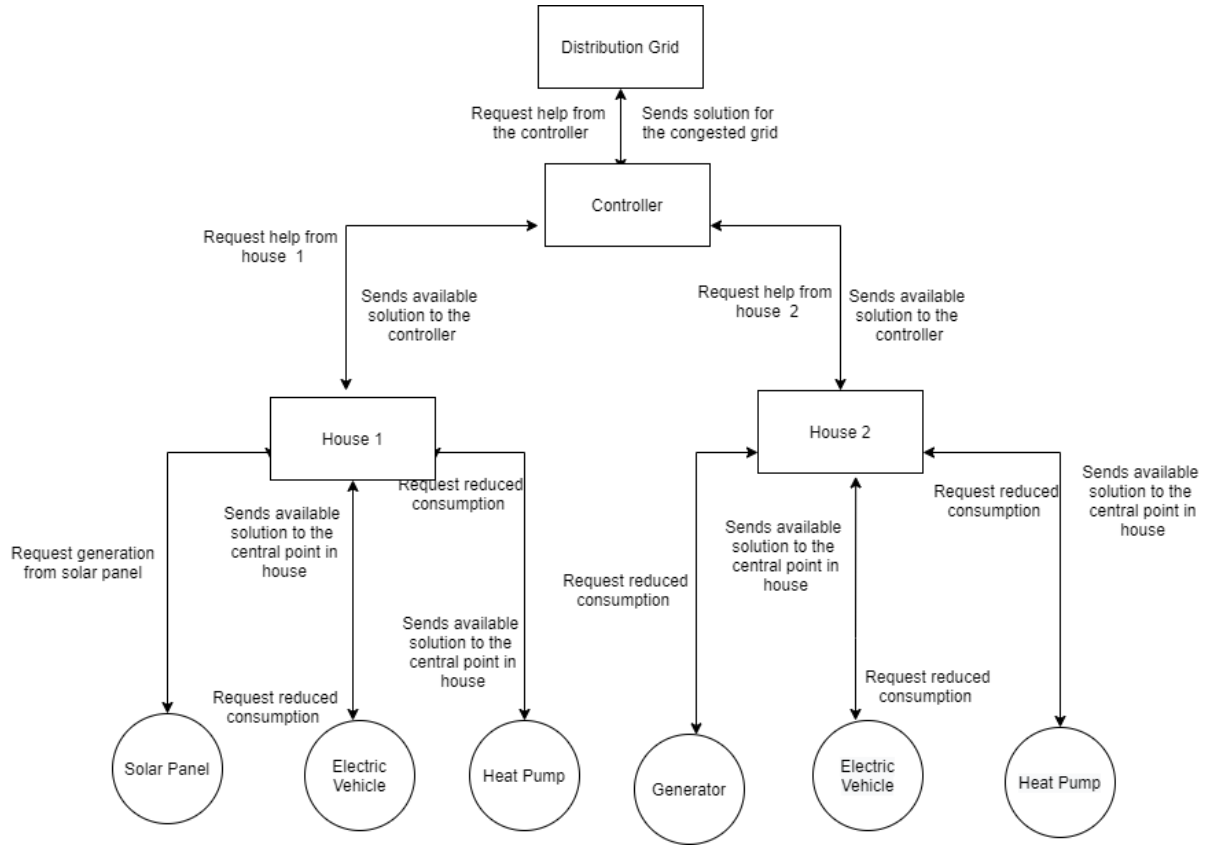


Figure 2.7: Illustration of town with 2 houses and its application, where each of rectangle and the circles represent a Raspberry PI

3

Design Analysis

This chapter is the most important part of this graduation project and details about the design analysis of the Middleware system. Moreover, this chapter gives broad limelight about different functionalities considered as part of the middleware system and what are their possible architecture configurations, which is most suited for this graduation project.

3.1. Middleware Functionalities

Based on the system requirements 2.5,design criteria 2.6 and energy system integration scenarios 2.4, some of the middleware functionalities are defined as follows

3.1.1. Discovery Services

This service is responsible for registering the Raspberry PI's or component from the application layer side and also retrieving neighbours from the same local network setup. It acts as an entry point into the middleware system from the end-user/application layer. It is important services as it keeps track of all the neighbour applications and also helps in defining the role of the application and its access limits.

3.1.2. Configuration Services

It is responsible for the configuration aspects of the middleware such as maintaining the back end configuration file, data storage, creation and maintenance of the topics which is important in the providing communication between different components or Raspberry PI's. It is most important in instrumenting the communication service as well as maintaining back end and data storage.

3.1.3. Power System Modelling Services

This service is responsible for the creation, maintenance of power system topologies, power flow computations, maintaining the status of the grid and also communicating the results and load variable changes to its neighbours, constantly updating the grid topologies. This feature of the middleware system adds the traditional power system modelling component to the middleware system and gives leverage to be used for power and ESI scenarios.

3.1.4. Communication Services

This particular service is the backbone of the entire middleware system, as it allows Interoperability , facilitate the communication between different components through inter-process communication protocol and responsible for state-of-the-art Information and Communication Technology (ICT) facilities, as well as demonstration of the IOT and ICT integration into the Energy System Integration (ESI) scenarios.

3.1.5. Energy Component Modelling Services

This service is responsible for the representation and emulation of the state of the art new hardware components like PV or EV, Converting the dynamic or steady-state models from a simulation environment into Co-Simulation based Functional Mock UP Units's. This service is of importance as it gives an edge of using the state-of-the-art renewable energy models which would be a game-changer in energy transition and Energy System Integration (ESI) scenarios.

3.1.6. Monitoring Services

This service would be used in realizing the dashboard feature of the middleware as well as logging and plotting the results. It could be a valuable feature for the end-users to plot, monitor their algorithms and data.

3.1.7. Data Analytical Services

This service could be used to predict or forecast the energy and power consumption's, help in decision making for congestion managements or for predicting the renewable energy sources during different seasons.

3.1.8. Data Conversion Services

This service could help in maintaining a standard data format for exchanging information between different components, thereby maintaining a standard data format.

3.1.9. Data Base Services

DataBase service involves maintaining a SQL or mongo DB for collecting and tracking data used during the demonstration.

After careful consideration, taking into account the fundamental requirements for Illuminator - Energy System Integration Development Tool Kit, we have decided to explore more and implement the discovery services, configuration services, power system modelling services and energy component modelling services as they

cater to the fundamental requirements of the middleware system. whereas the database service would make the middleware system complicated and we already have the internal storage feature of the data storage which is more than enough to handle the data storage related to the Energy System Integration (ESI) scenarios and by default, we enforce to use JSON format to enable a standard format thereby getting rid of the necessity to have data conversion service.

Data Analytical Services would be better suited for application layer and it cannot be generic for all the Energy System Integration (ESI) scenarios, it should be particular for different energy component, data models can be made only based on a data sets, it varies for different energy components over the season. Therefore the data analytical service would be difficult to implement from the middleware perspective and it is well suited for application layer during the definition, implementation of control logic and algorithms.

Even though monitoring service is an important feature, due to the limited time constraint and also for focusing on vital services of the middleware system. monitoring services is suggested for future implementations.

3.2. Middleware Architecture

Before diving into a distributed approach of the middleware system, all the services, its related internal python packages and tools have been tested in a specific application-centric model. it is done to ensure the compatibility of the software and packages with the Raspberry PI's Operating system. More about the specific application-centric model can be found in the appendix B.

For making the middleware system a distributed approach, as well as catering to all the energy system integration scenarios. It is developed as a python package, acts as an interface between the application layer and OS layer.
Middleware system is developed in python as it is open source and state-of-the-art, Raspberry PI's also have inbuilt python tool as said before.

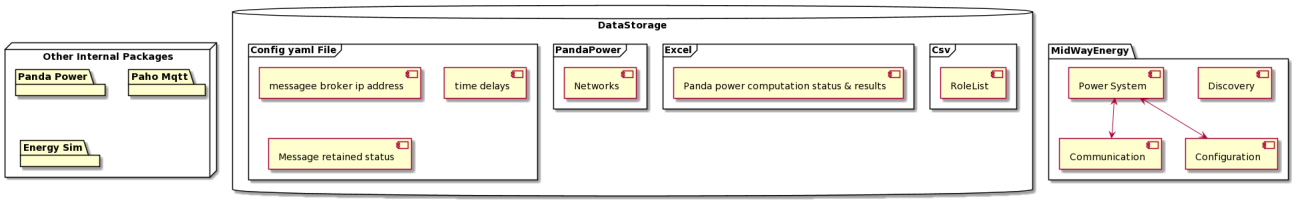


Figure 3.1: Component Level Illustration of Middleware system - Midway Energy

Python package architecture is followed to create a generic level of the middleware system libraries which can be used for most of the Energy System Integration scenarios.

All the python packages/libraries can be categorized into two types namely :

1. Regular Packages - This type of approach is followed in traditional package system in earlier python version like 3.2 or so, where __init__.py defined inside the directory and is first to implicitly executed and the objects defined in it are confined to names specified in the package's namespace.

2. Namespace Packages - This type of package is typically used in the case where the various portions(a set of files in the main directory compressed with .zip extensions) are made part of the package itself. it typically serves as a container for the sub-packages and they do not have __init__.py. [38]

Midway Energy python package/library is realised in the traditional regular packages manner as we define a class **dashboard** and objects to calls the methods and instances of other classes.
As stated in section 3.1, all the selected middleware services are encompassed in the middleware package as individual classes, and their functionalities are presented as a function call/API Calls.

The detailed list of functionalities and API calls for specific services, which are represented in the tables 3.1, 3.2, 3.4 & 3.3 and Figure 3.2 gives an indication of the interaction of different classes and their methods with the main class of the package **dashboard** and also reflects an overview of how other classes are interconnected.

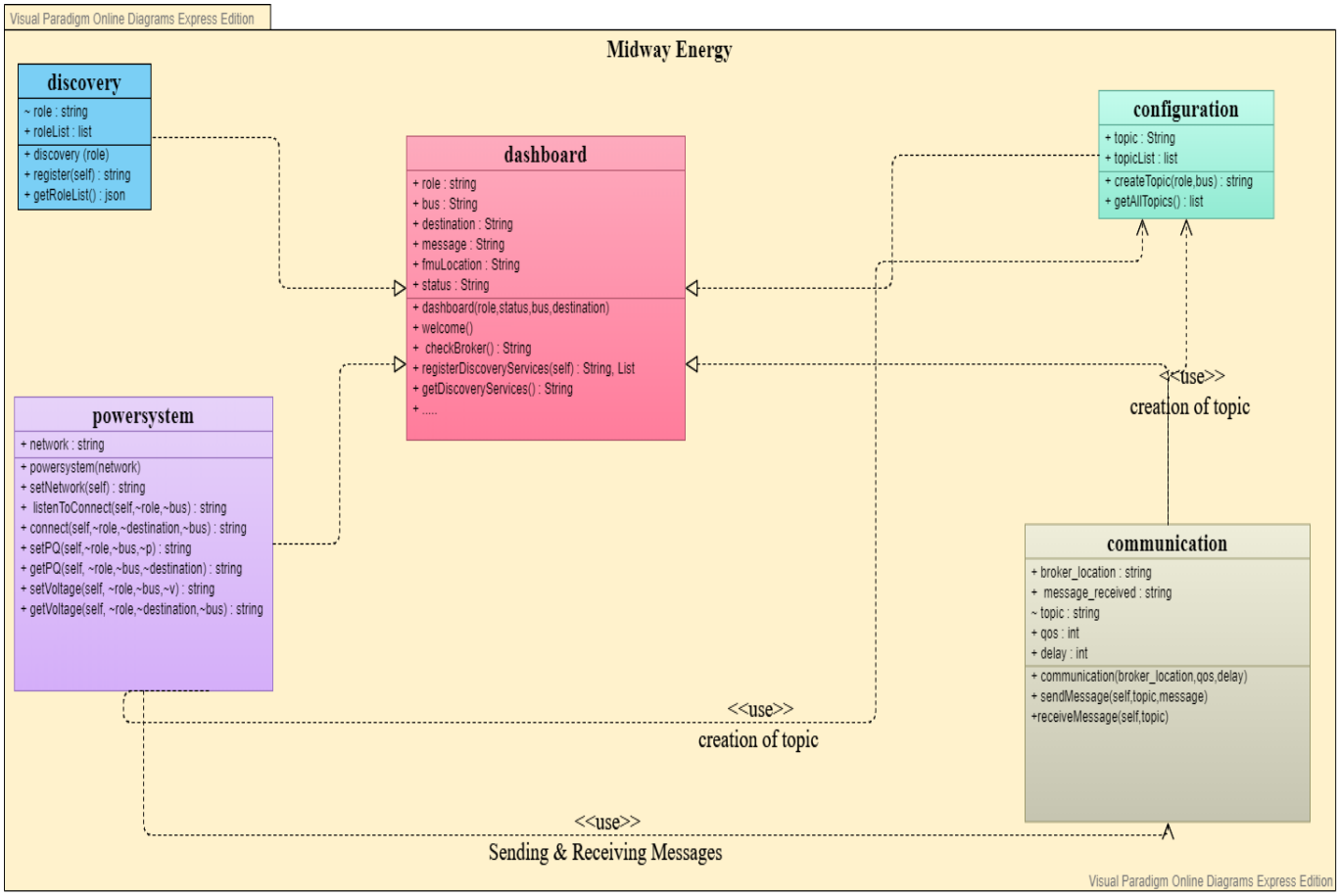


Figure 3.2: Interaction of Class Level Representation of the Midway energy Package

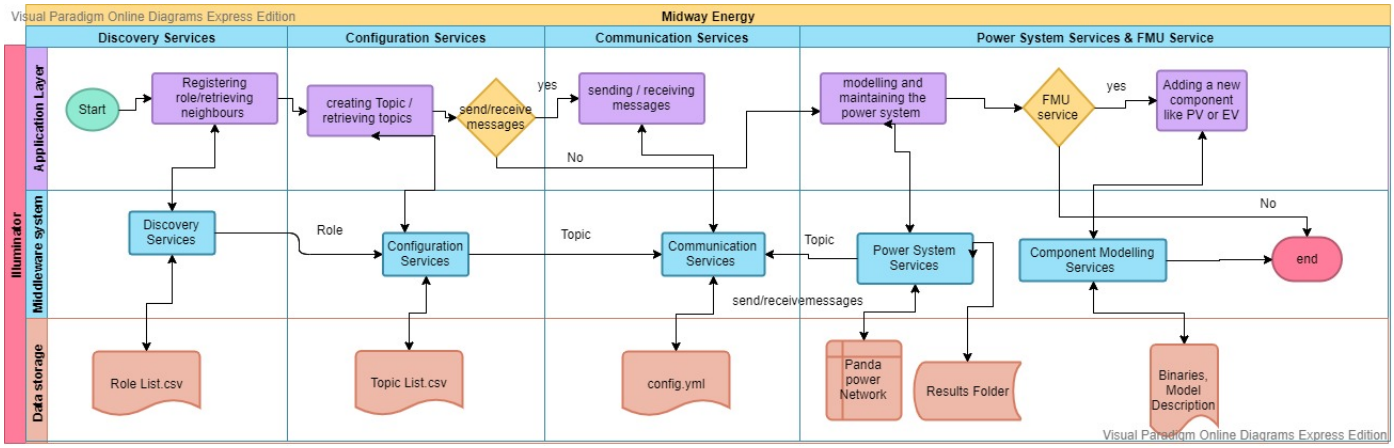


Figure 3.3: Illustration of System Level Context of Middleware & Application layer

whereas, Figure 3.1 shows the component view of the middleware python package named ***Midway Energy*** , associated internal packages Paho MQTT, Energy Sim , Panda Power and data storage variables.
overall Swimlane Diagram representation of the entire Middleware system and their interaction with internal data storage, configuration file and application layer is shown in the Figure 3.3 .
Finally, Figure 3.4 illustrate the interaction of the end users/application with the Middleware system with their simplified inputs and outputs.

More details of about the choices of each services API calls and methodology are dealt in upcoming sessions

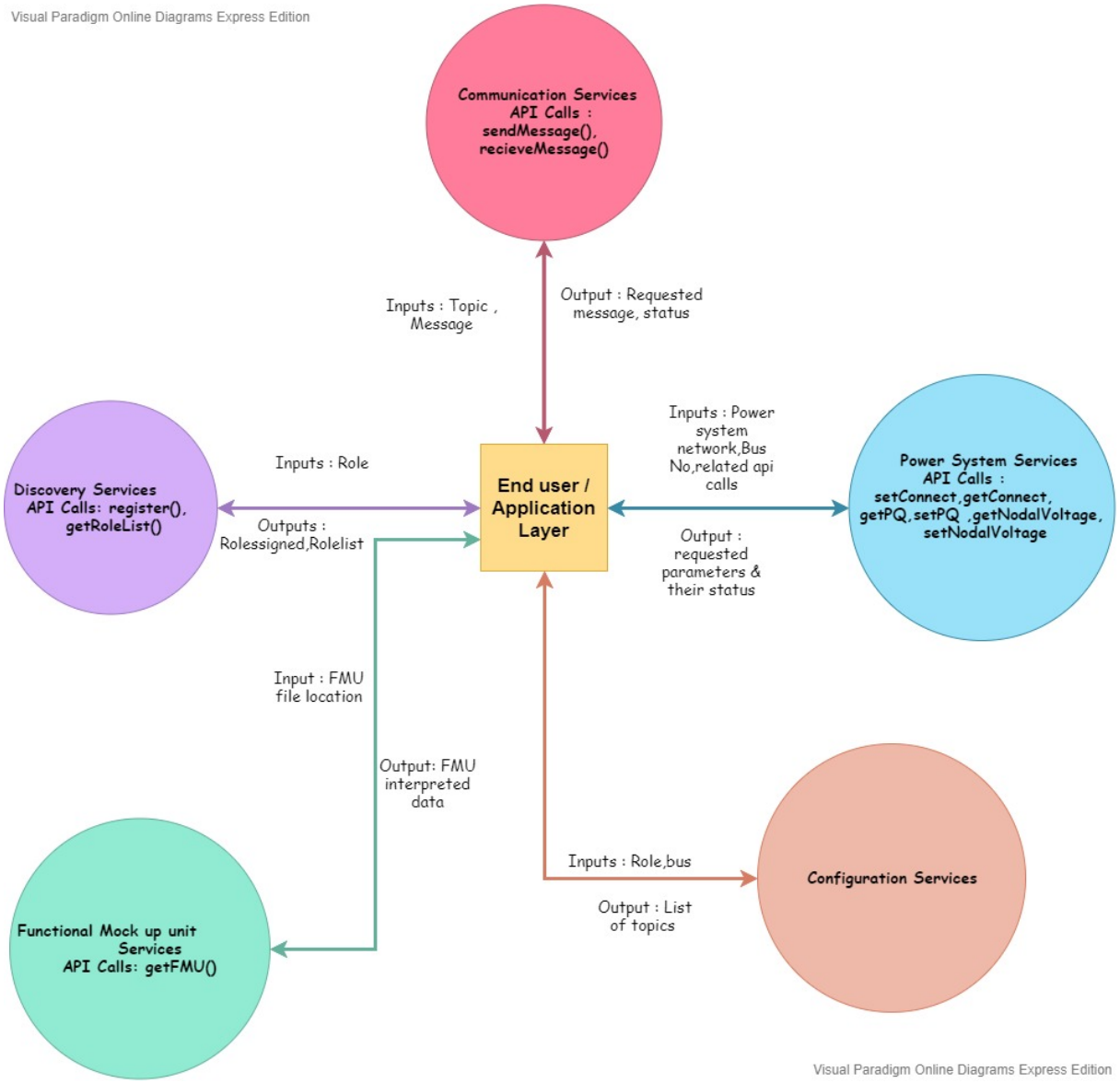


Figure 3.4: Interaction of end user/application layer with middleware system

3.2.1. Discovery Services

As mentioned earlier, this service is an entry-level point for the components or application layer or Raspberry PI (RasPi) into the middleware system. This service is mainly used to register and retrieve the roles of the application layer/Raspberry PI (RasPi)s. User-defined Application Programming interfaces (API) methods are implemented for the same taking into account the input and return values as shown in table 3.1.

The attribute role which assumes the literal meaning is defined dynamically to change it during the run time as well and give the application layer/RasPi leverage to change its purpose/action over time. The role which gets registered into the local illuminator network is saved in the data storage in the form of Comma Separated values (CSV) file and retrieved when requested through API calls.

Table 3.1: Discovery Services API Calls

Functional Description	API Calls	Return
Register the components/application in the middleware system	register(str role)	'role' : 'Distribution2'
Provide List of Connected Devices	getAllConnectedDevice()	"1", "distribution", "2", "prosumer", "3", "controller"

3.2.2. Configuration Services

This Service is important in instrumenting the communication services and serves as a second-level entry point into the middleware system. Topics are created and maintained, these are an important attribute to facilitate communication across different components and is defined in the user-defined API calls as given in table 3.2 with their following inputs and output. Bus number to which the components will be connected or involved in data transactions are also made part of the topic hierarchy and represented in 3.5. More details about the message transmission, scheduling and interprocess protocol are followed in upcoming sessions.

Another alternative by which the communication to the bus can be instrumented is by defining a vector variable, which can help in transactions of buses, by filling the associated row number with data needed to be exchanged. But, It would involve data storage for one-time use and complicate things further.
Whereas with topic tree/hierarchy approach it is easy to trace and we are re-using already existing feature of communication layer.

```
/illuminator
        /distribution #role
                /3 #bus number
        /battery #role
                /3 #bus number
        /load #role
                /3 #bus number
```

Figure 3.5: Example Illustration of topic hierarchy

Table 3.2: Configuration Services API Calls

Functional Description	API Calls	Return
Creation of Topic for that particular role	createTopic(roleassigned,bus ='13')	"topic "= 'illuminator/distribution1/13
List all the topics created – interface Call	getAllConnectedDevice()	Give the list of available topic and how to access it

3.2.3. Power System Modelling & Energy Component Modelling Services

This service would be the most crucial aspect of the middleware system as it involves modelling of power system topologies and would be the heart of visualizing Energy System Integration (ESI) scenarios, IEEE bus systems and making use of real-time power system simulations as in Power Factory, Open Modelica simulation environments.

It is also involves usage of internal packages such as Panda Power which is a open source power system modelling python package [39] and Energy Sim [40] (Co-Simulation based interpretation in python).
Even though, there are two kinds of Functional Mock UP Units which can help in exchanging dynamic models from one simulation environment to others namely:

1. Model exchange, where the solver is supplied by importing tool used in case computing the state derivatives, solver from the importing tool decides the time steps.
2. Co-simulation, both the solver and the model is exported together. The importing tool makes use of the exported solver's time steps.[41]

Co-Simulation Functional Mock UP Units's are used for integrating the dynamic/steady-state models of power system models as it is easy to interpret in python language.

The user-defined Application Programming interfaces (API) calls as stated in table 3.3 are oriented for enabling connection services, updating the state of the grid topology and to share the power system analysis results with the components of transmission or distribution grids.

As shown in table 3.3 , listentoConnect(roleassigned,bus) and connect(roleassigned,bus) is the API calls used for instrumenting connections between the transmission/distribution network and newly added components like battery storage.

setPQ(roleassigned,bus,p,q) & getPQ(roleassigned,bus) are the API calls used to update the state of the grid for the particular bus number. All the results after power flow analysis can be retrieved by using getVoltage() & setVoltage() API calls. For integrating the state-of-the-art renewable energy models getFMU() API call can be used.

3.2.4. Communication Services

It is considered to be the backbone of the middleware system facilitating message exchange through inter-process communication protocols. There are different types of inter-process communication protocols as discussed below.

- Object Request Broker or ORB-based middleware - The Common Object Request Broker Architecture (CORBA) is a standard tool developed and maintained by the Object Management Group (OMG) to provide interoperability among distributed objects.[42]
- Remote Procedure Call or RPC-based middleware - it is a protocol that one program can utilise to request a service from another program located in another computer on the same network without knowing much of the details of the network. It is commonly used in the local system. It is also known as a function or subroutine call.[43]
- Message Oriented Middleware or MOM based middleware - Message-oriented middleware is a kind of infrastructure that uses message exchange rather than function calls / shared memory. It's a design principle, and as a result, can be used anywhere. It's probably most useful in heterogeneous / high availability / high performance systems.[44]

Based on the system requirements 2.5 and also the features of the above-mentioned middleware system we choose to go ahead with Message Oriented Middleware as it scatters to different platforms, helps in exchanging messages rather based on function calls /shared memories.

3.2.4.1. Message Oriented Middleware or MOM based middleware

It is loosely coupled inter-process communication infrastructure which supports both hardware and software for both bi-directional sending and receiving of message between distributed systems (Cloud or Enterprise Service bus. it reduces the complexity of communication between the operating system and network interface as

Table 3.3: Power System and FMU related API Calls

Functional Description	API Calls	Return
Listen to connection request	listentoConnect(roleassigned,bus)	Yes connected
Connect to the bus in the network	connect(roleassigned,bus)	Yes connected
Set active & reactive power values	setPQ(roleassigned,p,q,bus)	{ 'P' : '0.02' , 'Q' : '0.045' }
Get active & reactive power values	getPQ(roleassigned,bus)	{ 'P' : '0.02' , 'Q' : '0.045' }
Set Voltage profile on the node :	setVoltage(roleassigned,V,bus)	{ 'V' : '50' }
Get voltage profile on the node :	getVoltage(roleassigned,bus)	{ 'V' : '50' }
Get all the nodal voltage	getAllNodelVoltage(roleassigned)	all the nodal voltage in case of distribution grid
Set all the nodal voltage	setAllNodelVoltage(roleassigned,voltages)	all the nodal voltage in case of distribution grid
Get the bus results after power flow	getBusResult(roleassigned,bus)	bus result , loading ,p ,q , all
Set the bus results after power flow	setBusResult(roleassigned,bus,busresult)	bus result , loading ,p ,q , all
Get the FMU Values	getFMU(fmulocation)	FMU interpreted value

it follows distributed communications.

Message Oriented Middleware can be utilized in API's which extend across multiple platforms. It supports asynchronous call between the client and the server applications. The application interface is the central point of communication for the application distributed on different network nodes. It also provides an administrative interface thereby making a virtual system of interconnected applications reliable and secure. [45] MOM can be further classified based on:

1. Peer-to-peer messaging - A unified middleware component in every peer coordinates discovery and interaction between peers.
2. broker-based messaging - The middleware a uses a physical broker to provide a messaging infrastructure between the heterogeneous peers.[46]

as a result different types of Message Oriented Middleware are represented below

1. **(Message Queuing Telemetry Transport)** It is an inter-process communication protocol based on MOM, it is mostly realised in embedded IoT systems, where a physical broker is required. it has low latency, dynamic brokerage and resilient feature of exchanging messages. It can also be used as distributed by the clustering message broker. it can be realised in all the coding languages. [47]
2. **(Advanced Message Queuing Protocol)** It is an application layer protocol that lets interaction between the client applications and the server. However, it should not be considered a protocol for over-the-wire communication; it defines both the network layer protocol and a high-level architecture for message brokers, but it is evolving, so the features are basic. [48]
3. **(Zero MQ Transport Protocol** It is a web socket-based protocol which uses peer-to-peer messaging architecture, it can be well suited for distributed systems but message routing is not possible, but it can also work with the broker to enable the dynamic leverage and message routing features of the message broker. it also has low latency, high through output and available in all the coding languages.[49] [46] [50]

Table 3.4: Communication Services and its related API Calls

Functional Description	**API Calls**	**Return**
Sending Messages to other devices	sendMessage(topic,message)	Message Sent
Receiving Messages from other devices	receiveMessage(topic)	{ Message in Json Format }

3.2.4.2. Comparison of Different MOM's

To select the best suited Message Oriented Middleware (MOM) based inter process communication protocol. it is further compared based on its availability, functionalities , suitability for illuminator,easily accessible forums and pros, cons in the tables 3.5, 3.6 & 3.7.

Table 3.5: Functionality comparison of ZMTP and MQTT

Types of Message oriented middle ware	Zero Message Transport Protocol (ZMTP)	Message Queing Telemary Transport (MQTT)
Message Topology	Wire point-point	Star topology
Need for physical server	not needed , it works on socket implementation	mostly for message broker , can be avoided if using the open broker of mqtt or eclipse
Message routing	maybe possible if broker is implemented	yes, with the help of broker
Type of messaging	Asynchronous	Synchronous
fault identification/ isolation	not ready	yes, easily accesible , message broker can be configured for cases when publisher is not available
Distributed Network	yes, with socket	yes with clustering of message broker , but reliability becomes less
Reliability / Dynamic	reliable but not dynamic	both reliable and dynamic
Open Source	is open source and available in all languages	is open source and available in all languages

It can be found that based on functionality as shown in table 3.5 between different MOM's type. Even though Zero MQ Transport Protocol is better suited for distribution setup and asynchronous. But, Message Queuing Telemetry Transport was still selected, emphasizing more on the message routing and resilient feature which it offers.

Based on the suitability , availability,pros & cons comparison from table 3.6 & 3.7 and also keeping in mind the design & system requirements. **Message Queuing Telemetry Transport** is found to be a good match for the implementing the communication services of the Middleware system as part of Illuminator - Energy System Integration Development Tool Kit.

Communication service is developed as a user-defined API calls realised on the internal python package Paho-MQTT which is an open-source software used for implementing based applications as shown in table 3.4. Also, Mosquito broker is activated as a physical server in Raspberry PI server, even though it can be used from open source IP addresses such as mqtt.org.eclipse or iot.org.eclipse. physical server IP address is used to trace the status of the IP address and makes use of an existing server in the local network.

Publish and subscribe feature which is used in the literal sense of the is used in the communication layer where the client can both publish or subscribe to a topic through message broker as shown in figure 3.6.
Topic plays an important role in letting the subscriber find the publisher and vice versa through the message broker.

To better understand the context of the publisher-subscriber let's take a simple example of TU delft energy club newsletter. let's say we are the subscriber who has subscribed to energy club on the topic of current energy trends. Here the publisher will be the energy club board and committee members, who publishes an article of current energy trends on a weekly or monthly basis. As we have subscribed to the energy club current energy trends even though whether we are active in the energy club forum we get a newsletter on the same as illustrated in Figure 3.6. Publish and subscribe feature also works on the same principle, but here we can set

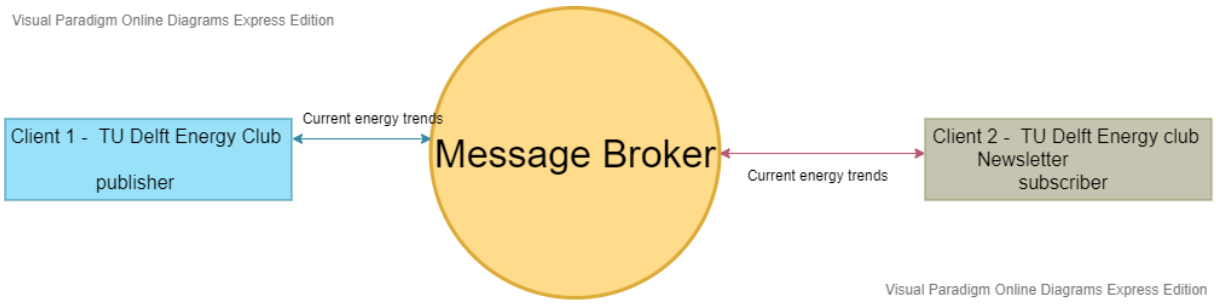


Figure 3.6: Illustration of based example

how to respond to incoming messages when the subscriber is not present or vice versa.

Communication layer can be made resilient by setting the QOS values to 1 or 2 in the back end configuration file, that is asking the broker to retain the message even when the subscriber is not active and set a will message that is to be sent to the subscriber when there is no active publisher.

Also by limiting the max message queue, the number of messages the client agrees to publish before it blocks and the maximum number of in-flight messages, that by using Paho MQTT to send/receive messages with Quality of Service (QOS) with 1/2 might result in error/exception like "Too many publishes in progress" in case many messages are sent in a short period. The straightforward fix is to use QOS with zero, but there are other possible ways to solve that problem.

The main root cause of this particular problem is that for every message sent with QOS > 0 the message is stored on the client-side, and sent to the broker and removed from client-side after the broker confirms receiving it (or resent if it doesn't – presumably, within a certain time frame). Such stored messages are called "in-flight" in Paho MQTT terminology, and they can be stored either in the file or in memory – depending on which MqttClientPersistence implementation is present [51]. All the characteristic feature of the middleware system can be modified in the back end configuration to adhere to the current specifications of the end-users.

Message exchange format used for exchanging messages is in the form of JSON. Even, when there are multiple messages sent to the subscriber all the messages can be retrieved by implementing and also specifying a maximum message queue size and limiting maximum-in-flight-messages.

3.2.5. Data Storage

Internal data storage is used in maintaining the transaction records related to the Energy System Integration (ESI) scenarios, which happened with the help of Middleware system. Also for tracking the records, by which application-layer get to know its neighbours.

Moreover, it contains back end configuration YAML isn't a markup language (YAML) file which can be helpful in reconfiguring the communication service to suit the Energy System Integration (ESI) case study.

Some of the important data and how it is stored internally via the middleware system is detailed below.

- Role and topic attributes would be stored internally in the form of CSV files and retrieved, Accessed by all

the components/raspberry pi.

- All the power system IEEE bus topologies are stored as panda power network files internally, accessed only by a role as distribution grid or transmission grid.
- Power flow results and computational variables are store in excel file and can be accessed by raspberry pi with the role as distribution grid or transmission grid.
- All the back end configuration of the communication services such as broker IP address, time delays, message retention capability, will message and maximum queue size is stored as attributes of configuration YAML file. So, the application layer can modify accordingly to their own needs.

Another alternative of creating and maintaining data storage and realizing a new service would be through the creation of database using MySQL or MongoDB, which further increases the complexity of maintaining and remembering the credentials of the database, occupying more spaces and all the data can be stored in an excel or CSV files for easy accessibility.

Table 3.6: Comparison of Different Types of MOM

Message Oriented Middleware	**Features**	**Suitable for Illuminator**	**Easy to Learn**	**Forum/Community**
MQTT (Message Queuing Telemetry transport)	1. it is broker-based architecture 2. it is a system consisting of a single or multiple brokers. 3. it is used for mainly for embedded system -IOT applications 4 . Hosting MQTT Broker is done by ->Use you Own Locally Installed Server ->Use a Cloud Based Sever or Virtual Server ->Use a Shared Server Application 6. works on eclipse foundation	Hive MQ - yes	Mosa (Node-red), Linkedin tutorial (mosquitto Mq)	Git hub, Hive MQ community
ZMTP (Zero Message Transport Protocol)	1. it is socket abstraction , also mom broker can be implemented 2. Socket API 3. Available in all the coding language 4. follows peer-peer messaging 5. The basic ZeroMQ patterns has: Request–reply Publish–subscribe 6.it offers several network transports, like TCP, UDP,IP multicast, and a number of sockets types for architectural patterns.	Yes	yes , Available in all the coding languages	Git Hub , Zero MQ community, Gitter ,
AMQP (Advanced Message Queuing Protocol)	1. Routing involves - it follows wire-level protocol(point to point), publish and subscribe, 2. Broker based messaging 3. it has selected the OASIS industry standards group 1	Not Really	Light weight e asy to deploy	Git Hub , Rabbit MQ community

Table 3.7: Continuation of Comparison of Different Types of MOM

Message Oriented Middleware	Open Source/License	Continuity in future	Pros	Cons	Available Software and features
MQTT	open source	Hive (better security and reliability)	1. it acts as a low-overhead, low-latency easy to implement, great way to send data, especially within embedded devices 2. it is specifically designed to act like transport based sensor for sending data and has emerged to the de-facto-standard in the field 3. The mosquitto broker has proven to be efficient & effective way with low delay	1. it is exposed to serious security setbacks and inability to make best use of resources or to support additional use cases 2. A crucial part missing in MQTT is the possibility to contact connected clients directly, as MQTT is a pure pub/sub protocol.	Hive MQ (Distri clusbuted broker by clustering) , Mosquito Mq (open source C message broker, window and linux), Active Mq,mosa(node.js)
ZMTP	Open Source Universal Messaging Library	emerging	1 It has low-latency / high throughput / data-intensive communication 2. it is proven to achieve very high throughput , while maintaining low delay, independent of the load 3. A decentralized,distributed system is possible 4. Flexible protocol,	1. No full-featured broker 2. implementation and less expressive than the other 3. protocols, as only prefix matching is supported. Therefore, crucial features may need to be and can be added for a deployment of ZeroMQ in IoT settings	Zero MQ , Nanomsg, however, is a reimagining of ZeroMQ
AMQP	Rabbit Mq is open source	it is still evolving	1. Different Software working on AMQP, can be deployed in client and server emphasizing interoperability 2. pragmatic approach of security and message reliability, has a future place as an ISO standard. 3. AMQP provides a very efficient set of messaging scenarios.	1. it does not have a standard API 2. no decentralized broker possible	Active MQ,Rabbit MQ

4

Case Study & Results

This chapter describes in detail the case study made for the Illuminator - Energy System Integration Development Tool Kit for demonstrating simple demand-side congestion management in a LV European cigre topology by peak shaving mechanism, which is achieved with a help of BESS and instrumented by the designed Middleware system. Moreover, the results obtained from the setup is validated and is presented in this chapter.

4.1. Case Study

The first level prototype of the Energy System Integration Development Tool Kit as shown in figure 4.1 will be the demonstration of a simple & decentralized energy scenario of demand side congestion management with

1. Raspberry PI A as distributed system operator
2. Raspberry PI B as prosumer household with load consumption
3. Raspberry PI as prosumer household with reserve battery
4. Raspberry PI D as a server/message broker

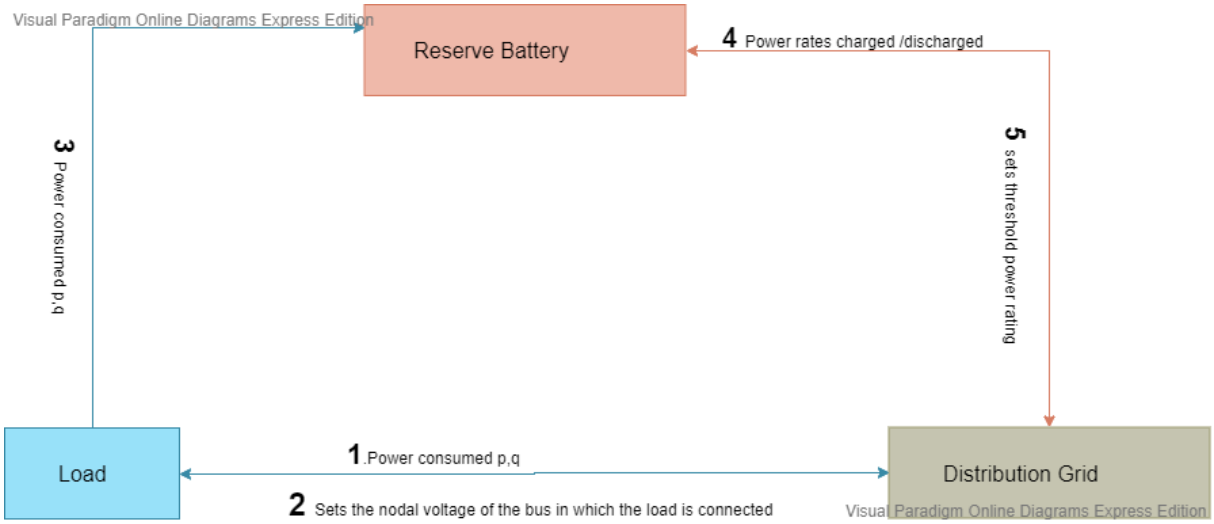


Figure 4.1: Illustration of Demand side congestion Management Case study

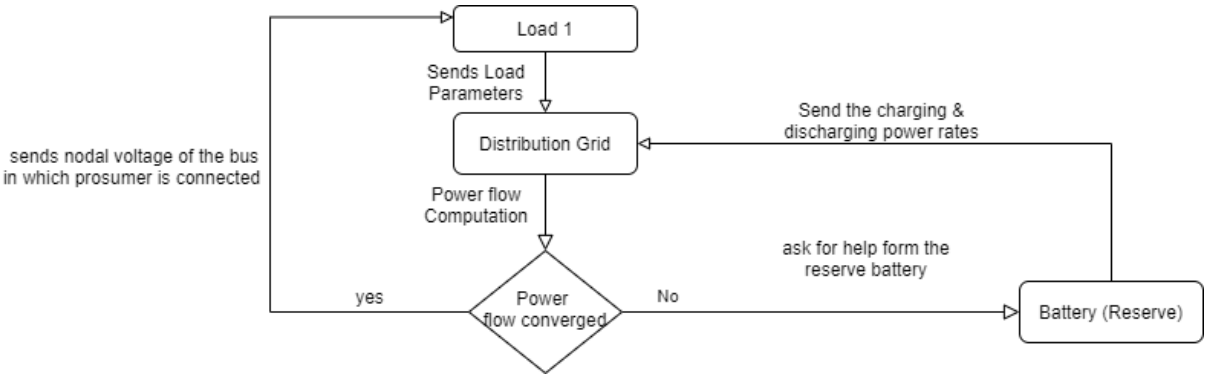


Figure 4.2: Flow chart representation of the demand side congestion management case study

This case study involves controlling the demand side congestion in the grid using the peak shaving mechanism. It is a mechanism mostly commonly used in BESS for cutting down peak demand charges on the electricity/utility bill. In this case study, based on the power threshold value set by the distribution grid applications the reserve battery is charged or discharged.[52]

All the different services developed as part of the middleware system would be utilized in realizing this particular case study and it is also validated with simple pandapower simulations.
As shown in Figure 3.4, users would be sending inputs to access the middleware layer to their convictions and for demonstrating Energy System Integration (ESI) scenarios.

4.1.1. Application Level Design

For this particular case study, the design choices of the application layers are as follows:

4.1.1.1. Distribution grid

For realising distribution grid topology, in this thesis, a benchmark CIGRE model developed by CIGRE task force C6.04.02 is used [53] [54]. These CIGRE model are designed to provide a framework approach for analysis and helps in validation of proposed methods for network integration of Renewable Energy Sources and Distributed Energy Resources (DERs), highlighting the significant challenges present owing to diverse nature of Distributed Energy Resources (DERs).

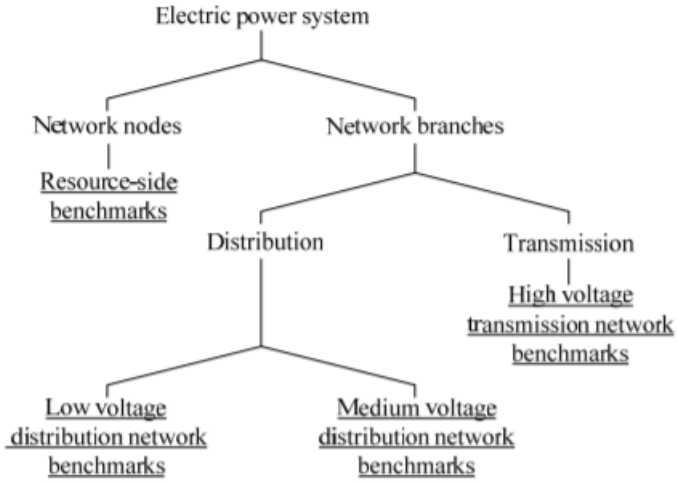


Figure 4.3: Identifying benchmarks through hierarchy structure

[53]

Even though there are four different hierarchical structure distinguishing transmission and distribution network as shown in figure 4.3 and further simplified for North America and European versions. For this test case, we would be using Low Voltage Distribution Network Benchmark as it is easy to implement and validate.

4.1.1.1.1 Low Voltage Distribution Network Bench mark

The real-world scenario of a LV the network is realised in this benchmark setup, while also promoting flexibility and user friendly for studies of Distributed Energy Resources (DERs) integration. This network setup comprises of three feeders namely:

1. Residential Feeder
2. Industrial Feeder
3. Commercial character Feeder

and the European representation. Some of the characteristics of European LV benchmark network is as shown in table 4.1.

For suiting the purpose of this project the default LV network, tweaked during various stages of using middleware services and only the residential feeder is going to be operated with the switch S1 connected and the other two switches S2 & S3 are disconnected as shown in figure 4.4. This is done to further simplify the operations and help in validating the outputs .More details regarding the LV distribution network is given in tables 4.2, 4.4 & 4.3 [53], [54]

4.1.1.2. Load Consumption

Load consumption variables which are affected by load impedance such the active(P-mw) and reactive(Q-mvar) power data values are modelled for Intra Day and Day Ahead scenarios with sample data set retrieved from real time LV building.

Table 4.1: Low Voltage European Network Benchmark Configuration

Structure	Physical low voltage network originate from medium/low voltage transformer, Configuration in the form of radial topology. Typical system frequency is 50 Hz. Low voltage network may include one/multiple lines Anywhere along the line,consumers can be connected
Symmetry	Single phase consumers makes this configuration inherently unbalanced, care is taken to reduce the unbalance
Line types	Urban areas are implemented using underground lines, whereas rural areas uses overhead lines. bare conductors made of aluminium is used for the construction of overhead line. cables are covered either in metallic or galvanized conduit
Grounding	Grounding depends on regional preferences, IEC 60364 classification is used , TN type or TT type of grounding is used for public LV networks, where the first T represents the neutral of the the transformer is connected to the ground, second letter N represents that frame of application being supplied is connected to the neutral The second letter T represents that the frame of application being supplied is connected to the ground.

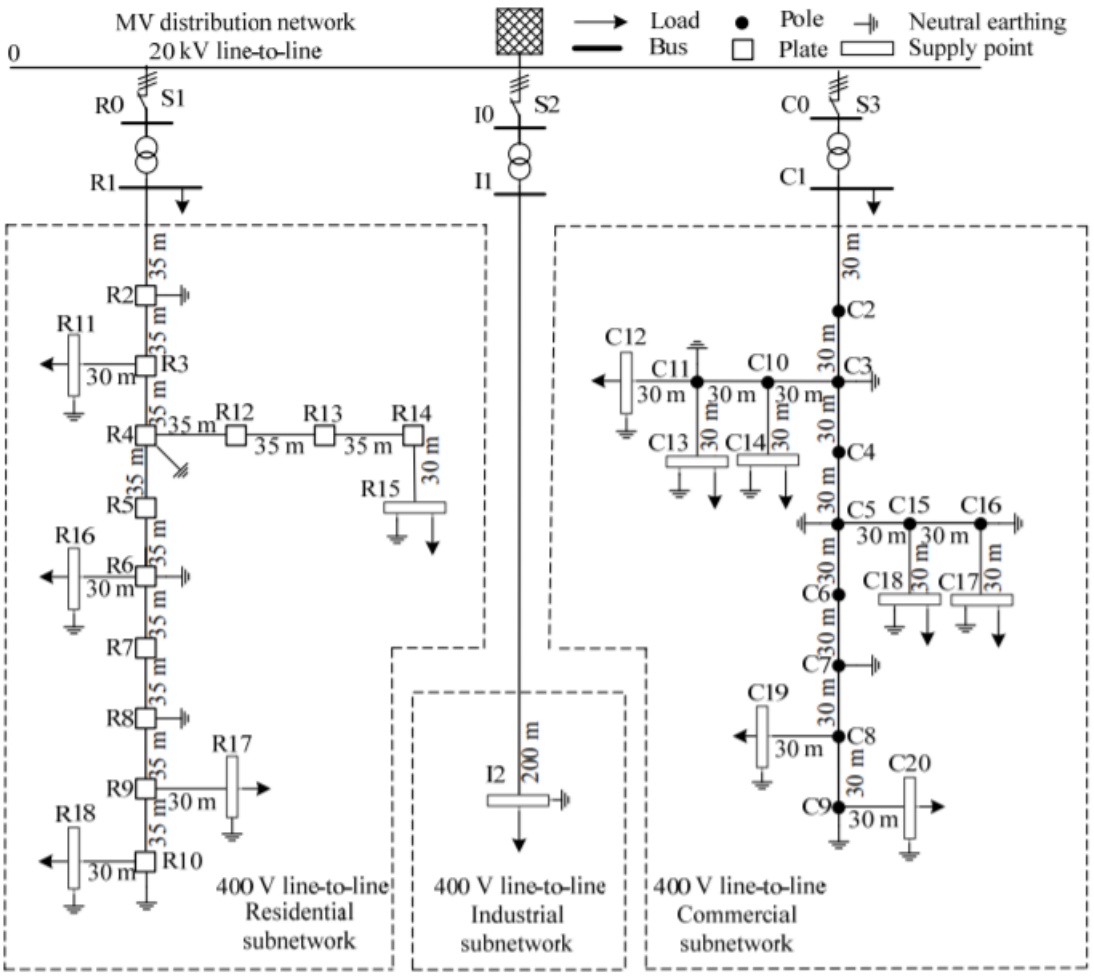


Figure 4.4: Topology of European LV distribution network benchmark 7.2.1
[53]

Table 4.2: Residential Bus Parameters of the European LV Network Benchmark

Index	Name	**vn_kv**
0	Bus 0	20
1	Bus R0	20
2	Bus R1	0.4
3	Bus R2	0.4
4	Bus R3	0.4
5	Bus R4	0.4
6	Bus R5	0.4
7	Bus R6	0.4
8	Bus R7	0.4
9	Bus R8	0.4
10	Bus R9	0.4
11	Bus R10	0.4
12	Bus R11	0.4
13	Bus R12	0.4
14	Bus R13	0.4
15	Bus R14	0.4
16	Bus R15	0.4
17	Bus R16	0.4
18	Bus R17	0.4
19	Bus R18	0.4

[53]

Table 4.3: Load Parameters of the LV European Network Benchamark

Index	**Name**	**bus**	**p_mw**	**q_mvar**
0	Load R1	2	0.19	0.06245
1	Load R11	12	0.01425	0.00468375
2	Load R15	16	0.0494	0.016237
3	Load R16	17	0.05225	0.0171737
4	Load R17	18	0.03325	0.0109287
5	Load R18	19	0.04465	0.0146757

[53]

4.1.1.3. Reserve Battery

To effectively realise the ESI scenarios a simple steady-state battery model developed with real-life commercial building data which is recorded load profile for 24 hours, the model is developed in Open Modelica and is exported as Co-Simulation based Functional Mock UP Units and interpreted with the help of energy component modelling service of the middleware system. [52]

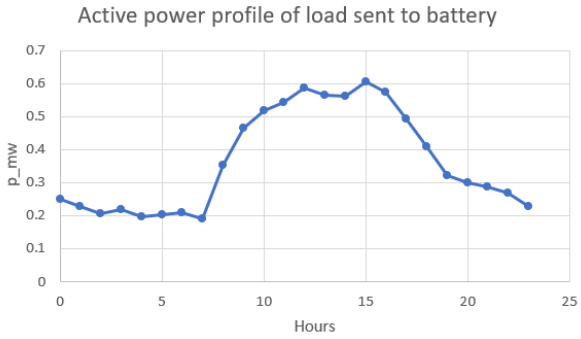
4.1.2. Message Broker

As mentioned in section 3.2.4, mosquito broker is realised with the physical Raspberry PI (RasPi) server and its IP address is configured in the YAML configuration file. The details regarding the back end configuration of the communication service can be found in table 4.5.

Table 4.4: Line and Connection Parameters of the LV European Network Benchmark

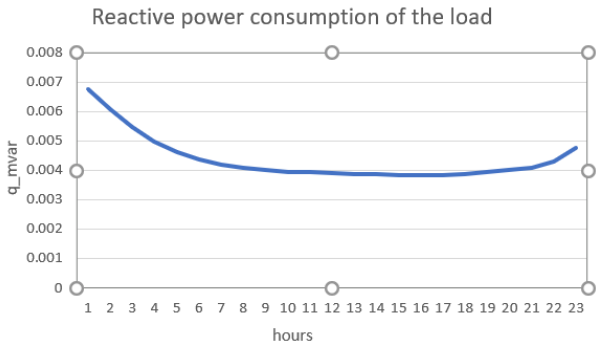
Index	Name	From_bus	To_bus	l(km)	R'_{p}h (ohm/km)	X'_{p}h (ohm/km)
0	Line R1-R2	2	3	0.035	0.162	0.0832
1	Line R2-R3	3	4	0.035	0.162	0.0832
2	Line R3-R4	4	5	0.035	0.162	0.0832
3	Line R4-R5	5	6	0.035	0.162	0.0832
4	Line R5-R6	6	7	0.035	0.162	0.0832
5	Line R6-R7	7	8	0.035	0.162	0.0832
6	Line R7-R8	8	9	0.035	0.162	0.0832
7	Line R8-R9	9	10	0.035	0.162	0.0832
8	Line R9-R10	10	11	0.03	0.162	0.0832
9	Line R3-R11	4	12	0.035	0.822	0.0847
10	Line R4-R12	5	13	0.035	0.822	0.0847
11	Line R12-R13	13	14	0.03	0.822	0.0847
12	Line R13-R14	14	15	0.03	0.822	0.0847
13	Line R14-R15	15	16	0.03	0.822	0.0847
14	Line R6-R16	7	17	0.03	0.822	0.0847
15	Line R9-R17	9	18	0.03	0.822	0.0847
16	Line R10-R18	11	19	0.03	0.822	0.0847

[53]



(a) Active power consumption of the load which sent to battery reserve for charging & discharging

[52]



(b) Reactive power consumption of the load

Figure 4.5: Illustration of active & reactive power load profile

Table 4.5: Back end configuration parameters

Config Parameter	Values
Message Borker	192.168.1.133
Quality of Service	1
Maximum queue size	10
Time Delay	15

4.2. Usage of Middleware system

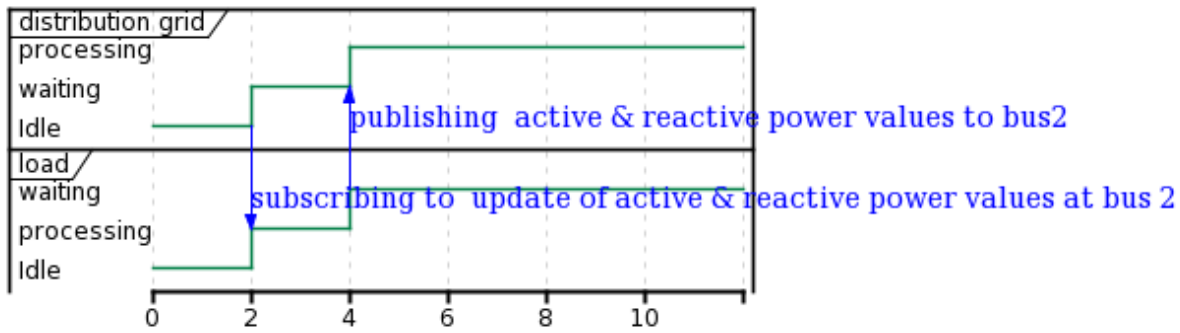
Now that we know the design level choices of the application layer. Let's see how this case study is implemented through the middleware system.

All the application layer would be given a list of interface calls and middleware characteristic in the form of catalogue to let the end-user to properly make use of the middleware system before it first invokes the middleware layer. All API calls from the application layer go to the functionality through the middleware system as it encapsulates all the functionalities within as shown in figure 3.3.

Middleware layer can be used by the end-user without remembering anything but with hardcore input through YAML files or getting user inputs. As shown in figure 4.1 and 4.2, demand-side congestion in the distribution grid is controlled by controlling the battery charging and discharging rates.
various steps involved in this case study is realised with the help of the middleware system as follows:

1. All the end users/application layers will first register and discovers the services the middleware system offers as described in section 3.2.1.
2. Required topic hierarchy is established which is essential instrumentation is enabling the exchange of information between the applications as reflected in section 3.2.2.
3. Back end configuration of the communication layer is set in the YAML file.
4. Co-Simulation based battery Functional Mock UP Units which was realised in Open Modelica is interpreted with the help of energy component modelling service of the middleware systems.
5. All the power system related activities like establishing a new connection between the battery and distribution grid, creating a storage component in the panda power networks, updating the state of the grid by setting and getting the values of active and reactive power values, sending the load their nodal voltage after load flow analysis is taken care by the power system service of the middleware system as discussed in section 3.2.3 and its working are represented in figure 4.8, 4.6 & 4.7

(a) Representation of update functionality



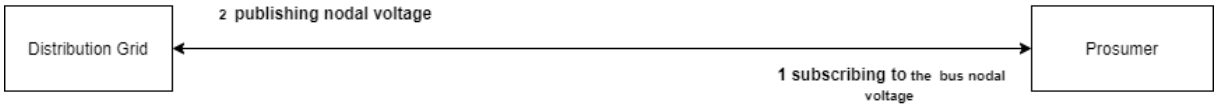
(b) Its related simplified timing diagram

Figure 4.6: Illustration of update functionality

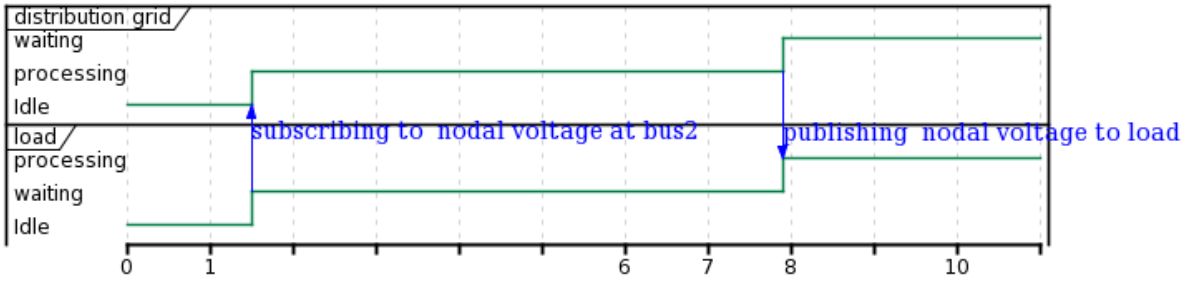
4.3. Results

In this section, the positive outcome of the case study is discussed in detail as well as the resulting plots are analysed. As shown in figure 4.1, demand side congestion management of the grid is solved with peak shaving mechanism in Intra Day and Day Ahead energy trading scenarios.

This particular case study highlights the importance and usage of a reserve battery. As said earlier in section 4.1.1.1, only the residential feeder is taken into account for realising this particular case study. Firstly, the active & reactive power of a particular load namely load R1 and connected to bus 2 is updated by using the updating functionality of the middleware system as shown in figure 4.6.



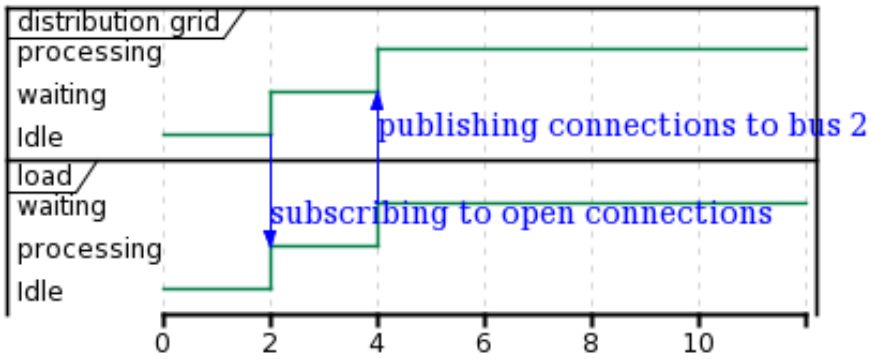
(a) Representation of nodal voltage



(b) Its related simplified timing diagram

Figure 4.7: Illustration of nodal voltage functionality

(a) Representation of Connect functionality

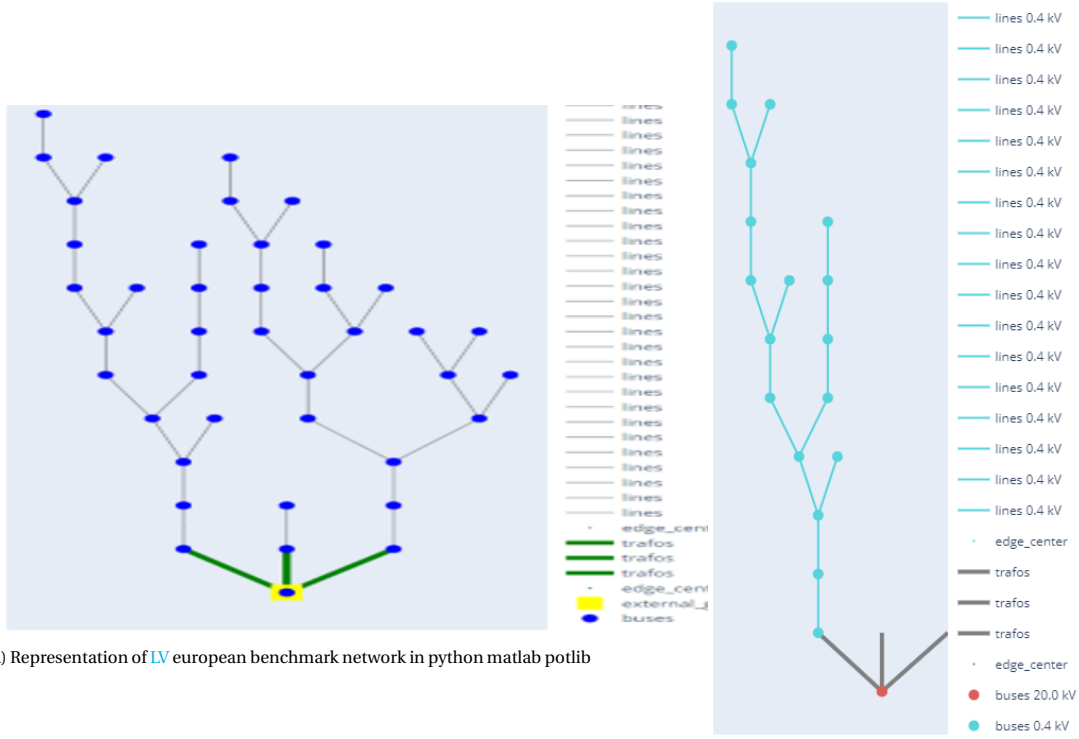


(b) Its related simplified timing diagram

Figure 4.8: Illustration of connection functionality

Next, the reserve battery is also connected to the same bus as the load by using the connection functionality as illustrated in figure 4.8. If there is no congestion and power flow is successful then the nodal voltage is sent back to the load as shown in figure 4.7. If not as shown in figure 4.2, the active power values of the load is fed into the battery using updating functionality and also the threshold power rates to determine charging and discharging is set by the distribution grid. Again by using the updating functionality the newly updated charging and discharging value is sent to the distribution grid to solve the congestion.

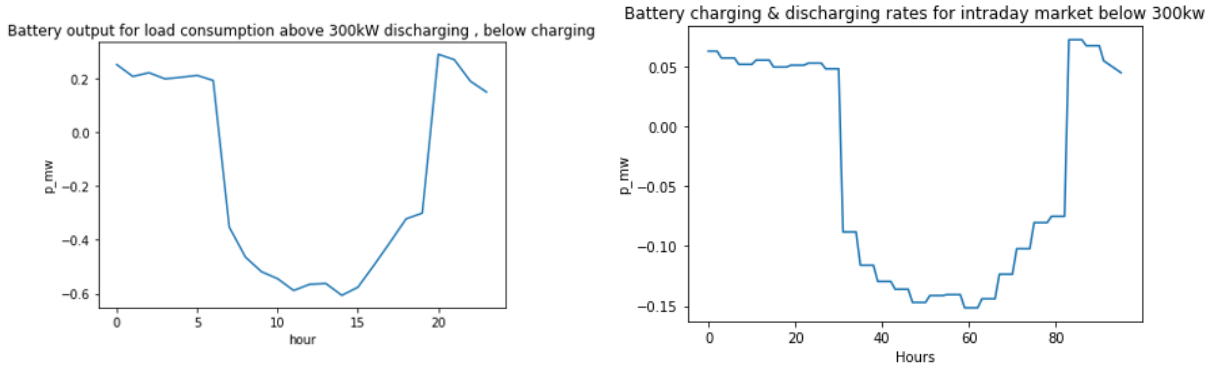
As shown in the figure 4.10, it can be seen the battery can be controlled in time steps to control the Day Ahead and Intra Day market as well as help in resolving congestion with the help of peak shaving mechanism.
The cigre Low Voltage (LV) topology is modified to induce congestion in the grid thereby raising power flow not converged exception. Once, this happens the distribution grid will send a request to the reserve battery to help solve the congestion.



(a) Representation of LV european benchmark network in python matlab potlib

(b) It highlights only the residential feeder which is used in this case study

Figure 4.9: Illustration of cigre Lv benchmark topology used in this case study



(a) Battery output for day ahead market

(b) Battery Output for intra day market

Figure 4.10: Battery output after peak shaving mechanism

4.4. Validation

The entire case study is validated in a single python script, thereby getting rid of all the exchanging information and waiting for the result from the previous function. It is quite interesting to note that both of the methods yield the same result. In case of the usage of Middleware system, there is also supportive visual which can be used to implement hardware and to highlight the flow of power and energy.

But, there is some time gap to go from one functionality to other and if one functionality fails to execute there is a cascading effect on other functionalities, which is not the case in the former method.

Also, usage of the Middleware system gives the user leverage to reconfigure or model the communication services and configuration services to selected grid topology and also let's user change the input values during run time as well. Middleware system is made flexible to cater to most of the Energy System Integration (ESI) scenarios.

5

Evaluation & Discussions

In this chapter, the outcomes of the selected and implemented the design of the Midway Energy(Middleware System) would be evaluated against the design requirements as specified in section 2.5, 2.6. Also, the usage of the middleware system for the proposed case study as discussed in section 4.2. It would be further evaluated for positive and negative test case scenarios to assess the performance, measurable key performance indicators as established in section 5.1.1 will be validated.

5.1. Evaluations

In this section, the evaluation of the constructed middleware system will be dealt with in a broader sense. This particular session is important in checking if the built middleware system adhere to the system requirements and design consideration, as they form the basis of developing the layered approach of the Illuminator - Energy System Integration Development Tool Kit and in particular Middleware system.

The design choices made for implementing the Middleware system as discussed in chapter 3 will be revisited and evaluated for the system requirements and design criteria from section 2.5, 2.6.

The Middleware system functionalities include the state-of-the-art Information and Communication Technology (ICT) like Message Queuing Telemetry Transport, Paho MQTT and Internet of things (IOT) technologies , power system modelling tools as such as Panda Power and Energy Sim as specified in the system requirements section 2.5 . Midway energy - a traditional python package/library is developed in a general sense, so that most of the Energy System Integration (ESI) scenarios are realised , it can be reusable, re configurable as it is implemented in the state of the art open source coding language python.

Communication service is realised as the backbone of the entire system, after thorough research, it was implemented with the help of the inter-process communication protocol Message Queuing Telemetry Transport keeping in mind the future expansion and the resilient feature which is most important for the exchanging messages between different components.

As Raspberry PI (RasPi) are the components which are going to be utilised for demonstrating the Energy System Integration (ESI) as defined by the scope of the project Illuminator - Energy System Integration Development Tool Kit, all the functionalities, internal packages have been tested before it was encapsulated inside the middleware system as a package as shown in appendix B

Power system modelling capabilities of the Middleware system is enhanced by the use of pandapower(an open-source traditional power system modelling software), even the futuristic models of the renewable energy models are integrated with the middleware system with the help of energysim, a Co-Simulation based Functional Mock UP Units interpreted in python.

For realising the real-world power system challenges during the demand side congestion case study the Low Voltage (LV) European network benchmark as developed by Cigre task force[C6.04.02] is used, real-world LV based battery data samples were used.

Pre-existing Operating System (OS) layer is utilised in testing the developed midway energy package and the memory shared by the client Raspberry PI (RasPi) as part of the net booting is exploited for the implementation of the Middleware system. The configuration level flexibility of exploring/customizing the communication layer is also made possible through the back end YAML configuration files.

The entire Middleware system is realised with a sustainable coding practice, thereby enhancing the longevity of the tool, till the technology is still used, can be reusable and reconfigured easily. The middleware system is made user friendly and makes the work of the application layer easier and easily understandable.

5.1.1. Important Measurable Key Performance Indicators

Developed Middleware system is multi-faceted and there is a needs to monitor several different Key Performance Indicators's to provide insight into how the Middleware system is performing. Some of these measurements may be more project scope related, like the number of power system components that can be integrated. While other Key Performance Indicators's can be more involved, such as latency and server availability.

Once Key Performance Indicators goals are defined, it is important to understand how those measurements can be used to enhance the performance of the system. [55] Key Performance Indicators with regards to Middleware system can be categorized as technical aspects related Key Performance Indicators and project scope related Key Performance Indicators's. which is discussed further in the upcoming session. These two categories are considered to assess the performance from the Middleware perspective and also for evaluating the future of the Illuminator - Energy System Integration Development Tool Kit.

5.1.1.1. Project scope related to Key Performance Indicators

These KPIs are based on the overall performance factor of the Illuminator - Energy System Integration Development Tool Kit setup and can help in predicting the future and possible modification during scalability of the same. Few of the selected KPI's are listed below

1. **No of case study possible** - This particular kpi is used to check the flexibility of the middleware system towards Energy System Integration (ESI) scenarios and an important Key Performance Indicators in deciding the energy transition capability of the Illuminator - Energy System Integration Development Tool Kit.

2. **No of power system components integrated** - This KPI is most important in evaluating the power system capable of the middleware system, as it is a most important feature of the middleware system.

3. **No of Application layer components that can be connected at a time** - No of components that can be connected at the time is important to realise the extent of a case study and how many components can be involved in one case study

4. **No of services offered by the Middleware system** - Quality and quantity of the services offered by the Middleware system would help in the end-user in realising the ESI scenarios.

5.1.1.2. Technical aspects related to Key Performance Indicators

These KPI are used for assessing the performance of the services realised as part of the middleware system and can help us track the performance of the middleware system as part of Illuminator - Energy System Integration Development Tool Kit layered approach. Some of the selected KPI's are listed below

1. **Latency** -It determines how fast the contents within a pipeline can be transferred from the client to the server(message broker) and back, it would be important KPI in assessing the performance of the communication services. [56]

2. **Server Availability** - Availability of server is important in routing the message through the server from one client to another and also acts as an important attribute in net booting setup.

3. **Delay Time** - It is the time difference between the execution of two different functions, This KPI is important in not missing out on publisher message and also gives extra time to process the information.

4. **Maximum queue size** - It is an important characteristic feature of the publisher, which helps in limiting the number of messages which can be published at a time.

5. **No of bus realised in the grid topology** - This particular KPI helps in identifying the number of nodes that can be made possible as part of a given grid topology or IEEE bus system, assessing the capabilities of the power system modelling

6. **No of message broker possible** - This KPI would be useful in checking for backup option if one message broker is not available at the given time.

7. **Throughput** - it is an important characteristic feature of the communication service which ascertain the amount of data which can be transferred over a given time frame.[56]

8. **Maximum In-flight message size** - It is an important characteristic feature of communication layer which can limit the sluggish nature of the Message Queuing Telemetry Transport protocol by limiting the number of in-flight messages present for a particular client.

9. **No of Functional Mock UP Units which can be integrated with one raspberry pi** - This particular KPI can help in evaluating the computational capability of the raspberry pi with regards to the number of Functional Mock UP Units's it can incorporate through the energy component modelling services.

There are some of the KPI's selected for evaluating the performance of the middleware system and categorized. Almost all the KPI's are validated for the performance in the upcoming session. But, few KPI's such as the number of case study possible with the current setup and No of Functional Mock UP Units which can be integrated with one raspberry pi cannot be explored more due to time constraints and need for more resources to perform the tasks.

5.2. Discussions

The project Illuminator - Energy System Integration Development Tool Kit is first of its kind, it has been successfully defined for the scope of RasPi's and has a pre-defined layered approach, implemented till the Middleware system. Even though, there are quite a several existing demonstrators in the field of game theory and real-time simulator. Illuminator - Energy System Integration Development Tool Kit can be realised combining both the theories and also backed up by real-time visualization.

Middleware system has come a long way starting from defining the functionalities of the same to selecting the architecture to implement, validating with a case study and KPI's. Generic visualization of the middleware system gives special leverage to let the end-users modify it for their specifications and can be easily modified in future. As stated, in the session 4.1, a very simple first level prototyping of the ESI scenario has been made possible for demonstrating the demand side congestion management with peak shaving mechanism by BESS and its positive test case is analysed in section 4.3.

There is still a lot of case study which can be implemented with the help of the existing middleware system, which would be the future work for the end-users trying to assemble the application layer with the help of the middleware system. Some of the features needed to handle the scenarios like Cyber Security of power grids still need to be explored. Even though the Middleware system is preciously constructed, it is still in the basic skeleton level and there is a lot of scope for improvements. Basic core functionality and capabilities of the middleware system have been defined, which would be the foundation for upcoming versions.

With the existing middleware system, we can realise all the demand side congestion-related case studies as well as load flow studies. PV, EV, battery system can be used by making use of the energy component modelling feature of the middleware system. But, still, an energy component modelling service can be better explored and suited for the demonstrating ESI scenario's. As we have already seen the positive scenarios of the middleware system, let's discuss more the negative test cases and how it is handled. Moreover, let's explore more details regarding the validation of selected KPI's, advantages and limitations of the designed middleware system in upcoming sessions.

5.2.1. Test Cases

As stated already, all the positive outcomes of the middleware system are discussed in chapter 4 and have been validated. In this section, let's discuss more the possible negative outcomes of the middleware system and how it can be resolved.

As shown in the table 5.1 below, there are quite several exceptions which can disrupt the functioning of the middleware system. Some of the most important error handlings should be emphasised for the availability of message broker, as it is the heart of message routing feature of the communication service and also the backbone of the whole middleware system.

The most important and common errors that can be possible from implementation are null values and case sensitivity of the python language. Exception handling should be imposed for the same. Also, No file exists exception can be possible if there was no prior data stored in the data storage until the particular function is called. Data not found, bus not found and redundant data exception can intrude in the way of running of the demonstrator and can destroy the purpose. Therefore, all these exceptions need to be properly handled.

All these exceptions might be a small red flag during the execution of the demonstrator but can disrupt the purpose if not handled properly. Even though the middleware system is handled for the possible errors, negative outcomes, there is still a high scope of optimizing the exception handling codes as well as checking for a few more negative case scenarios.

Table 5.1: Exception Handling of the Middleware system

Possible Exceptions	Exception Handling
Null	Null checks implemented all along the code
Case Sensitive	All the sensitivity related to the alphabets are evaluated
No file exists	Before, doing any operations in particular file, its existence is first checked
Redundant data	Duplicate data are managed before storing the data storage
Message queuing issues	Maximum message queue size is fixed and messages are retrieved in FIFO
Data not found	If there is no particular information not found it throws a data not found exception
Bus not found	If there is no particular bus in the network,then it gives rise to bus not found exception
Message broker is not found	If there is no message broker, default IP address will be used as message broker
Data type conversions	All the data are exchanged in the JSON format to enable easy readability and accessibility
Availability of the message broker	before instrumenting the communication services, the message broker availability is checked through IP pings

5.2.2. Validating the Measurable Key Performance Indicators

KPI's are defined in session 5.1.1 for evaluating the performance of the built Middleware system and categorized into scope related and technical related KPI's.

As shown in from table 5.2, at the moment we can say we can realise a case study with three different application, this is the case because as we use netbooting setup and there are only 4 wired slots available in the router for the Raspberry PI (RasPi)s. Out of which the server Raspberry PI (RasPi) would be used only as a message broker for the communication service and also for facilitating the net booting feature, thereby limiting the capability of the demonstrators, this can be resolved by using wireless net booting setup or some other booting setup of the Raspberry PI (RasPi).

Around 5 different services are offered by the middleware system namely discover, configuration, power system modelling, communication and energy component modelling at this time. Still, some services like the monitoring can be incorporated in the upcoming version, the present features of the middleware system are more than enough to realise most of the Energy System Integration (ESI) scenarios. But, still, scenarios like Cyber Security feature can be explored and incorporated.

Table 5.3 depicts the technical related KPI values of the middleware system, as presented the latency of the Message Queuing Telemetry Transport is quite low and it is good for exchanging messages between different components, server availability totally depends on the pi server and it is also responsible for the net booting aspect of the RasPis. Right now, the server availability tested for an entire case study as discussed in session 4.1 would be 2-3 minutes. But, the major disadvantage is that if the server is out of order then the client raspberry pi's cannot boot up and the server-client setup is useful in realizing the current middleware system as they share common space, data can be accessed by all the existing RasPi's. [57]

Throughput of the selected Message Oriented Middleware (MOM) based Message Queuing Telemetry Transport protocol is good for sending continuous messages and make the protocol fast in nature. Delay Time between activating a publisher and subscriber is around 15 secs. Thereby giving some time for the subscriber to connect first and the publisher to publish messages later. [56] [58]

Maximum -in-flight message size and maximum queue size is important in handling the traffic during message exchange and can be an important KPI in evaluating the performance of the communication services. By

default the maximum-in-flight message size and maximum queue size is set to 10, it can extend up to 665535 messages.[51]

No of message broker possible at this moment is only one but it can be increased in future by using clustered message broker as offered by Hive MQ application or using web socket feature of Zero MQ. It has still not been explored as Hive MQ is commercial closed solution.[59] [60]

No of bus realised in the grid topology is a vital KPI for realising n IEEE bus system as part of power system modelling services. Right now it depends on the capability of the Panda Power python software. But, with the case study, we are safe to say we can realise up to 44 bus and can be extended

Most of the technology-related KPI's are related to communication service as it is the backbone of the established middleware system. There can still be more which can help evaluate the middleware system. Therefore, based on the outcome, we can say communication service is well established and suiting the purpose of the Illuminator - Energy System Integration Development Tool Kit and contribute to the vital part of the Middleware system.

Even though there is a limitation of the applications that can be realized for the case study. But, it is safe to say the designed Middleware system is well defined to suit the purpose of demonstration of small scale LV grid topology. Most of the power system components such the generators, transformer, bus, load, storage, line and state-of-the-art renewable energy models are integrated into the system.

Table 5.2: Evaluation of Project Related Key Performance Indicators's

Project Related Key Performance Indicators's	**Recorded Value**
No of power system components integrated	7
No of Application layer components that can be connected at a time	3
No of services offered by the Middleware system	5

Table 5.3: Evaluation of Technical related Key Performance Indicators's

Technical Kpi's	**Recorded Value**
Latency	120 milliseconds
Server Availability	All through the execution of the case study 3-4 mins
Delay Time	15 seconds
Maximum queue size	10
No of bus realised in the grid topology	It depends on the grid topology selected and IEEE bus system selected Right now it is 19
No of message broker possible	1
Throughput	1 message per second per publisher
Maximum In-flight message size	10

5.2.3. Advantages

This particular middleware system has explored and integrated large scope of power system capabilities with the help of Panda Power and Energy Sim power system modelling software and N number of bus and different grid topologies can be realized as part of the power system modelling services such as high voltage, medium voltage, and low voltage distribution networks.

The resilient feature of the communication service is realized by exploiting the inbuilt resilient feature of the Message Queuing Telemetry Transport and state-of-the-art ICT service is achieved by the help of Paho MQTT software and mosquitto message broker. Communication service has low latency and high throughput, proper queuing system to tackle the message trafficking, also the dynamic message brokerage of the Message Queuing Telemetry Transport protocol is exploited to cater to any possible exceptions in the communication services.

Sustainable coding practices are practised which can help in enhancing the reusability, re reconfigurability, and scalability of the middleware system. Laid a foundation for the possible integration of the state of the art renewable energy models which not found in the traditional power system modelling with the help of Co-Simulation based Functional Mock UP Units. Data storage associated with the middleware system is made redundant free and.

5.2.4. Limitations

Even though, there are several benefits associated with the middleware system.
There is quite some trade-off which was compromised in establishing the middleware system which can limit some of the design consideration of a distributed network such as the need for physical message broker/server and message Routing feature of the communication services.

RasPi pi server is central and it takes care of the net booting setup as well as acts as the message broker for communication services. But, the major limitation is if there is an issue with the pi server there is no backup message broker unless the open-source message broker is utilised and also limit the working of other client Raspberry PI (RasPi)'s.

Although the net booting server-client setup is useful for realising the middleware system by sharing the common spaces of the client, due to presence of the router and wired connections it limits the number of raspberry pi which can be used for demonstrating the ESI scenarios as part of Illuminator - Energy System Integration Development Tool Kit

6

Conclusion & Recommendations

In this chapter, the conclusion of the entire graduation thesis is described. At the end of this part, the answers to the research questions as well as the recommendations for future work is detailed upon.

Even though there are several existing demonstrators and the project is still in the budding stage of implementations, there is a huge scope for the Illuminator, as it is first of its kind. It can be transported anywhere as a portable device, it helps in better visualization of energy system integration challenges, and it can be applied as a practical education toolkit. The illuminator is aimed to be scaled to real-time simulations of energy scenarios as well as spatially scaled across the multiple sites of TU Delft.

During this project tenure, we have established the core functionalities and architecture of the middleware system, which would be the foundation for upcoming versions and realized a very simple case study that can be realized through the functionalities of the developed middleware system. The Middleware system is generic and can be used for instrumenting most of the Energy System Integration (ESI) scenarios in the application layer as part of Illuminator - Energy System Integration Development Tool Kit.

This project has explored the integration of traditionally available power system modelling tool Panda Power, state-of-the-art inter-process communication protocol , and also explored the possibility of integrating the futuristic renewable energy models through Co-Simulation based Functional Mock UP Units's into the middleware system which can be full visualized in the Raspberry PI (RasPi).

The middleware system designed is capable of catering to most of the Energy System Integration (ESI) scenarios such as the demand side management,Electric Vehicles (EV) & Photo Voltaic Panel (PV) integration. But, still, it can be made more flexible towards Energy System Integration (ESI) scenarios such as demonstrating the impacts of the Cyber Security, also the futuristic models of the renewable energy models have to be explored more in details and further discussions regarding the same can be found in section 6.3.

Although the developed middleware has shortcomings in terms of the distributed aspect of the communication services, It needs more research in the direction of integration of state-of-the-art renewable models and to tackle scenarios like Cyber Security. Overall the basic skeleton level approach of the middleware system is well defined. It has evolved a long way from the architecture presented in appendix B.

6.1. Main Contributions

- This graduation project has been developed to establish the core functionalities and architecture of the middleware system as part of the whole illuminator project.
- A very simple first prototyping of the illuminator was done with an example of the case study on the demand side congestion management with the help of the peak shaving mechanism, which is realized with the help of the middleware system functionalities, and its results were validated.
- Another key contribution is the development of the python-based middleware system library which encompasses all the middleware system functionalities specified till now and can be expanded/changed in the future to cater to the ever-growing needs in the Energy System Integration (ESI) domain.
- Sustainable coding practices have been followed, that give the middleware leverage to be scaled up, reconfigurable, and reusable in the future.
- The research findings of this particular graduation project has been submitted to the International Young cigre 2020 Paris NGN showcase competition under the study committee D2 - Information systems and telecommunications

Some of the answers to the research questions as proposed in chapter 1 is answered in the upcoming sessions.

6.1.1. Results to Research Questions

6.1.1.1. Design Considerations

What are the important design considerations to be considered while developing the middleware system as a distributed Network?

Based on the sub research questions the main research question is answered

1. *What are the Existing demonstrators, what can be learned from them and their drawbacks?*

   Several existing demonstrators are dealt with in chapter 2, they are categorized as game theory Application of Game Theory to Power Grids and Real-Time Simulator-Demonstrators available.
   Most of the demonstrator listed above is focused on the decision-making capabilities of the humans in the energy system integration and energy transition. Few of them are used for real-time simulation purposes, which helps in better visualization of the power system on the grounds of energy system integration and transition.
   Two of the demonstrators namely the smart power flow control & monitoring and Embedded toolkit uses raspberry pi's for their computation and illustrations. Flexmeter is also realized in the same layered manner as that of the illuminator and the message-oriented middleware approach would be used for implementing the illuminator.
   Need for Illuminator Most of the existing demonstrators are closed web page based solution. whereas, Illuminator is going to be the first of its kind.

2. *What should be considered for System-level requirements ?*

   Based on the present energy integration scenarios and need for integrating the ICT & IOT technologies and also keeping in my mind the need for real-time simulators and the complexity, necessary services needed for a power system engineer to model or demonstrate the energy system integration challenges and benefits. System-level requirements are defined in section 2.5

3. *What are the Design consideration to be taken into account and associated challenges?*

   Some of the design considerations to be taken into account as stated in section 2.6 are scalability, configurable, res usability, flexible to Energy System Integration (ESI) scenarios, the realization of distributed networks and use of open-source software and off the shelve hardware requirements.

6.1.1.2. Design Analysis

What are the different software architectures possible for the middleware system?

This particular research question is answered with the following set of sub research questions.

4. What are the different middleware configurations? Which is the best-suited message exchange format?what are the available open-source software for building middleware? Which are Key functionalities of the middleware? How to integrate steady-state/dynamic models of the power system into middleware?is the middleware system reconfigurable? How? are the results of the distributed system setup validated with the single panda power simulation? is there a necessity for security level in the communication layer? Why? Why not? is the Communication protocol resilience enough? How?

The Middleware system can be built as a python package encapsulating all the key functionalities as shown in chapter 3 or else it would be difficult and time-consuming to define it for each ESI scenarios as shown in the application-centric approach in appendix B. But, the generic middleware system has more leverage over the later as it can be further expanded and can be used in a distributed setup. JSON is preferred for exchanging messages as it is easy to read and interpret Some of the open-source software used in building the middleware include mosquito brokers, Paho-MQTT python package, pandapower power system modelling tool, and energy sim, Co-Simulation based Functional Mock UP Units python interpreter.

Some of the key functionalities of the middleware include power system modelling, energy component modelling, communication service, configuration services, and discovery services are implemented till now as described detail in section 3 and few more services can also be defined in the future. Co-Simulation based Functional Mock UP Units's are used for integrating the steady-state/dynamic models of the power system into the middleware system. Yes, there is not much a difference between the implemented distributed setup and python

single script yes, the middleware system can be reconfigured to cater to the needs of the application layer yes, the resilient feature of the communication protocol Message Queuing Telemetry Transport is implemented using the Quality of Service (QOS) and will message function of the publisher and subscribers. No, the security level of the communication protocol is not required as it is well defined by itself, and introducing the security level might further complicate implementation.

6.1.1.3. Key Performance Indicators

What are the measurable key performance indicators to be considered for the assessment of the middleware system? Key Performance Indicators's are categorized as technical related and project scope related to separately assess the future scope of the Illuminator - Energy System Integration Development Tool Kit and also the technical workings of the middleware system.

Few of the project related Key Performance Indicators's which was defined can be found in session 5.1.1,5.2.2 such as No of power system components that can be integrated, No of Application layer components that can be connected at a time, No of services offered by the middleware system. These KPI's can help in the evaluation of the scope of the project illuminator in terms of scalability and reconfigurability. Based on the evaluation it is safe to say the project can apply to small scale demonstrators with few applications at the moment, but at the same time, it can be scaled, modified to cater to the increasing demand of Energy System Integration (ESI) scenarios.

Whereas, the technical related Key Performance Indicators's help in assessing the performance of the middleware system in terms of reliability, resiliency, security, and also reconfigurability. If we take a close look at the evaluated output it can be said that the communication service is resilient, fast in action, and have high performance, but there is still a chance of failure as it is not a distributed network in terms of communication service and there is still no backup message broker other than the pi server.

Based on the evaluation of both the technical related and project-related KPI's, The developed middleware system is well defined in terms of functionality and can be reconfigured, scaled and can demonstrate small scale ESI scenarios with few applications.

6.2. Challenges Faced

There are quite several existing demonstrators in the field of energy, but still, the illuminator is the first of its kind which has all the components and capability of the demonstrators starting from mini computers like Raspberry PI (RasPi), software system, and hardware system. Therefore, realizing the functionalities of the middleware system was quite challenging, it needed some thinking out of the box and also thinking from the perspective of the end-user with an example of a case study.

It was challenging to define the design considerations and sticking to it, there was a high probability of getting distracted in the process. It was my first-hand experience with Raspberry PI (RasPi) and Linux Operating System (OS) and in creating python-based libraries therefore it took me some time to understand and explore the capabilities of the same. The development of the entire middleware system was done in desktop PC as it was really difficult to build a middleware system based on python package in the shell environment of the raspberry pi without a raspbian desktop. It was quite a challenge to integrate the Co-Simulation based Functional Mock UP Units's into the raspberry pi as there is a prevailing problem with the ctype python libraries.

6.3. Recommendations

Based on the scope and result of the middleware system constructed till now in this graduation project, some recommendations regarding the existing middleware system are suggested in this section, and also the future work of both the middleware system and Illuminator is described.

As said earlier, there is still a huge scope of exploring some more aspects of the middleware system in terms of integration of state of the art renewable energy models, defining and developing the monitoring services based on some of the pre-existing visualization tools such as InfluxDB[61], Telegraf[62], and Grafana[63].

More research can be done on the side of making the middleware system flexible towards scenarios like Cyber Security impacts and functionalities like the monitoring services have to be explored. Furthermore, as there is still some trade-off in the communication services as it is still not a distributed system, clustered message broker services of the Hive MQ(()) [59] can be utilized provided it becomes open source down the lane or the possibility of using Zero MQ [60] library in the middleware system can be explored in the future.

This particular thesis laid the foundation for the middleware key functionalities and architecture, it can still evolve over the period, and code optimization, unit testing can be done. Moreover, there are still limitations for the number of the components which can be represented as part of the application layer as there are a few slots in the router for Local Area Network (LAN) connections and also due to net booting setup. Wireless net booting or other ways of booting the RasPi can be explored to help in the scalability of Illuminator - Energy System Integration Development Tool Kit.

More Key Performance Indicators's can be defined and validated to help in better understanding the capability of both middleware and the illuminator. Future scope of the illuminator can be extended to a multidisciplinary project which can be scaled up across the TU Delft campus and also demonstrate real-time simulations in Energy System Integration (ESI) scenarios.

BIBLIOGRAPHY

[1] *Importance of energy transition*, 2020. [Online]. Available: https://www.spglobal.com/en/research-insights/articles/what-is-energy-transition.

[2] *Leading countries in energy transition*, 2020. [Online]. Available: https://www.smart-energy.com/renewable-energy/revealed-the-countries-leading-the-way-in-renewable-energy/.

[3] B. News, *Energy transition - a challenge with no historical precedence (part one)*, 2020. [Online]. Available: https://balkangreenenergynews.com/energy-transition-a-challenge-with-no-historical-precedence-part-one/.

[4] *Energy transition*. [Online]. Available: https://www.irena.org/energytransition.

[5] *Tu delft ewi energy transition*. [Online]. Available: https://www.tudelft.nl/ewi/onderzoek/energy-transition/.

[6] *Consumer vs prosumer: What's the difference?* [Online]. Available: https://www.energy.gov/eere/articles/consumer-vs-prosumer-whats-difference.

[7] *The illuminator*. [Online]. Available: https://www.tudelft.nl/powerweb-institute/research/the-illuminator/.

[8] J. Morgan, *A simple explanation of 'the internet of things'*, Apr. 2017. [Online]. Available: https://www.forbes.com/sites/jacobmorgan/2014/05/13/simple-explanation-internet-things-that-anyone-can-understand/.

[9] *Hardware in the loop*. [Online]. Available: https://www.mathworks.com/help/physmod/simscape/ug/what-is-hardware-in-the-loop-simulation.html.

[10] *The illuminator - energy integration development kit*. [Online]. Available: https://www.tudelft.nl/en/eemcs/the-faculty/departments/electrical-sustainable-energy/intelligent-electrical-power-grids-iepg-group/projects/current-projects/illuminator/.

[11] P. D. Justo, *Raspberry pi or arduino? one simple rule to choose the right board: Make:* Jan. 2017. [Online]. Available: https://makezine.com/2015/12/04/admittedly-simplistic-guide-raspberry-pi-vs-arduino/.

[12] G. Hollingworth, T. Archer, J. Pallant, J. Witts, M. Redrobe, J. Sanderson, P. v. d. Hijden, S. Edgar, A. Ellis, H. Kirkman, and et al., *The raspberry pi piserver tool*, Jan. 2018. [Online]. Available: https://www.raspberrypi.org/blog/piserver/.

[13] *Network boot your raspberry pi*. [Online]. Available: https://www.raspberrypi.org/documentation/hardware/raspberrypi/bootmodes/net_tutorial.md.

[14] C. Cawley, *6 causes for a raspberry pi that won't boot (and how to fix them)*, Feb. 2020. [Online]. Available: https://www.makeuseof.com/tag/raspberry-pi-wont-boot-fix/.

[15] J. R. Molina, "The role of middleware in distributed energy systems integrated in the smart grid", in *Energy Management of Distributed Generation Systems*. 2016.

[16] M. Rouse, *What is application layer? - definition from whatis.com*, Mar. 2018. [Online]. Available: https://searchnetworking.techtarget.com/definition/Application-layer.

[17] M. O'malley, B. Kroposki, B. Hannegan, H. Madsen, M. Andersson, W. D'haeseleer, M. F. Mcgranaghan, C. Dent, G. Strbac, S. Baskaran, and et al., "Energy systems integration. defining and describing the value proposition", 2016. DOI: 10.2172/1257674.

[18] *Intelligent electrical power grids (iepg) group*. [Online]. Available: https://www.tudelft.nl/en/eemcs/the-faculty/departments/electrical-sustainable-energy/intelligent-electrical-power-grids-iepg-group/.

[19] *Powerweb institute*. [Online]. Available: https://www.tudelft.nl/powerweb-institute/.

[20] *Stimulering onderwijs en onderzoek elektrische energietechniek*. [Online]. Available: https://www.stichting3e.nl/beurzen/.

[21] D. McNulty, *The basics of game theory*, Jan. 2020. [Online]. Available: https://www.investopedia.com/articles/financial-theory/08/game-theory-basics.asp.

[22] J. Chen, *Nash equilibrium*, Mar. 2020. [Online]. Available: https://www.investopedia.com/terms/n/nash-equilibrium.asp.

[23] *Networks*. [Online]. Available: https://blogs.cornell.edu/info2040/2018/09/17/game-theory-in-power-grid-management/.

[24] G. Pepermans, "European energy market liberalization: Experiences and challenges", *International Journal of Economic Policy Studies*, vol. 13, no. 1, pp. 3–26, 2018. DOI: 10.1007/s42495-018-0009-0.

[25] L. De Vries and Harsha, *Electricity markets simulation game*. [Online]. Available: http://fieldsofview.in/projects/electricity-market/.

[26] *Balance of power*. [Online]. Available: https://www.usef.energy/the-game/.

[27] *Universal smart energy framework*. [Online]. Available: https://www.usef.energy/.

[28] *Create your own energy future*. [Online]. Available: https://energytransitionmodel.com/.

[29] *Real-time simulation*. [Online]. Available: https://www.mathworks.com/discovery/real-time-simulation.html.

[30] *Smart grid demonstrator*. [Online]. Available: https://www.iosb.fraunhofer.de/servlet/is/14625/.

[31] A. DGO, *Heat*, Jun. 2020. [Online]. Available: https://vimeo.com/143874353.

[32] S. Hollanders and J. Wijnen, *Heat 3d design tool*. [Online]. Available: https://amsterdamsmartcity.com/products/heat.

[33] M. E. Hernandez, G. A. Ramos, M. Lwin, P. Siratarnsophon, and S. Santoso, "Embedded real-time simulation platform for power distribution systems", *IEEE Access*, vol. 6, pp. 6243–6256, 2018. DOI: 10.1109/access.2017.2784318.

[34] S. S, S. P, and N. G .R, "Smart power flow monitoring and controlling using raspberry pi", *IJSTE - International Journal of Science Technology & Engineering*, 2349-784X, vol. 3, no. 10, Apr. 2017.

[35] R. Gross, R. Hanna, E. Gazis, J. S. Edge, and A. Rhodes, "Unlocking the potential of energy systems integration: An energy futures lab briefing paper", *Research Gate*,

[36] C. Cambini, R. Congiu, T. Jamasb, M. Llorca, and G. Soroush, *Energy systems integration: Implications for public policy*, May 2020. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0301421520303463.

[37] "Energy systems integration", *Integration of Renewable Energy Systems*, pp. 121–136, 2016. DOI: 10.1115/1.861240_ch11.

[38] *The import system*. [Online]. Available: https://docs.python.org/3/reference/import.html.

[39] L. Thurner, A. Scheidler, F. Schafer, J.-H. Menke, J. Dollichon, F. Meier, S. Meinecke, and M. Braun, "Pandapower—an open-source python tool for convenient modeling, analysis, and optimization of electric power systems", *IEEE Transactions on Power Systems*, vol. 33, no. 6, pp. 6510–6521, 2018. DOI: 10.1109/tpwrs.2018.2829021.

[40] D. Gusain, M. Cvetkovic, and P. Palensky, "Energy flexibility analysis using fmuworld", *2019 IEEE Milan PowerTech*, 2019. DOI: 10.1109/ptc.2019.8810433.

[41] M. H. is product manager for Modelon's FMI products. She helps customers integrate and implement FMI in workflows, *Fmi standard: Understand the two types of functional mock-up units*, Oct. 2019. [Online]. Available: https://www.modelon.com/fmi-functional-mock-up-unit-types/.

[42] *Object request broker*. [Online]. Available: https://www.sciencedirect.com/topics/computer-science/object-request-broker.

[43] M. Rouse, *What is remote procedure call (rpc)?*, May 2020. [Online]. Available: https://searchapparchitecture.techtarget.com/definition/Remote-Procedure-Call-RPC.

[44] D. A. Chappell, *Enterprise service bus*. [Online]. Available: https://www.oreilly.com/library/view/enterprise-service-bus/0596006756/ch05.html.

[45] *What is message oriented middleware (mom)? - definition from techopedia*. [Online]. Available: https://www.techopedia.com/definition/27589/message-oriented-middleware-mom.

[46] S. Celar, E. Mudnic, and Z. Seremet, "State-of-the-art of messaging for distributed computing systems", *Proceedings of the 27th International DAAAM Symposium 2016 DAAAM Proceedings*, pp. 0298–0307, 2016. DOI: 10.2507/27th.daaam.proceedings.044.

[47] *Mqtt*. [Online]. Available: http://mqtt.org/.

[48] *What is amqp and why is it used in rabbitmq?* [Online]. Available: https://www.cloudamqp.com/blog/2019-11-21-what-is-amqp-and-why-is-it-used-in-rabbitmq.html.

[49] *Zero mq*. [Online]. Available: http://wiki.zeromq.org/whitepapers:brokerless.

[50] L. Bhatia, "Message queues in industrial iot",

[51] Mvmn, *Paho mqtt client, max in-flight messages for qos > 0*, Jul. 2016. [Online]. Available: https://mvmn.wordpress.com/2016/07/23/paho-mqtt-client-max-in-flight-messages-for-qos-0/.

[52] R. H. G. Tan and G. K. Tinakaran, "Development of battery energy storage system model in matlab/simulink", *International Journal of Smart Grid and Clean Energy*, vol. 9, no. 1, Jan. 2020. DOI: 10.12720/sgce.9.1.180-188.

[53] *Benchmark systems for network integration of renewable and distributed energy resources*. CIGRÉ, 2014.

[54] K. Rudion, A. Orths, Z. Styczynski, and K. Strunz, "Design of benchmark of medium voltage distribution network for investigation of dg integration", *2006 IEEE Power Engineering Society General Meeting*, 2006. DOI: 10.1109/pes.2006.1709447.

[55] *Key performance indicators: Kpi examples & their importance*, Dec. 2019. [Online]. Available: https://signatureanalytics.com/understanding-key-performance-indicators-kpi-examples-and-importance/.

[56] *What is latency and how to reduce it - keycdn support*. [Online]. Available: https://www.keycdn.com/support/what-is-latency.

[57] *Measuring publish-subscribe latency of mqtt*. [Online]. Available: https://iotify.help/network/latency/mqtt.html.

[58] *Benchmark of mqtt servers*, 2020. [Online]. Available: http://www.scalagent.com/IMG/pdf/Benchmark_MQTT_servers-v1-1.pdf.

[59] D. Obermaier. [Online]. Available: https://www.hivemq.com/blog/hivemq-open-source/.

[60] *Zeromq*, 2020. [Online]. Available: https://zeromq.org/.

[61] *Influxdb: Purpose-built open source time series database*, Jul. 2020. [Online]. Available: https://www.influxdata.com/.

[62] *Telegraf open source server agent*, Jun. 2020. [Online]. Available: https://www.influxdata.com/time-series-platform/telegraf/.

[63] *Grafana: The open observability platform*. [Online]. Available: https://grafana.com/.

A

List of Hardware and Software

A.1. Hardware

4 Raspberry Pi's
1 Server - 3 Clients Raspberry pis set up for net booting
1 WIFI Router
Several LAN cables and power chords

A.2. Software

Coding language used : Python
Software developed: Python traditional library.
Development IDE: Spyder Environment
Internal Packages used : Energy Sim , Panda Power , Paho MQTT
Inter process communication protocol used : Message Queuing Telemetry Transport

B

Application Centered Specific Model

The present middleware system has evolved a long way from being an application-centric model to generic python library/package catering to most of the Energy System Integration (ESI) as well full filling all the design considerations and system requirements are set in sessions 2.5, 2.6.

As shown in session 3.2.4, firstly the interprocess communication protocols were evaluated and based on the research MOM based inter-process communication protocols were found suited for the purpose. Moreover, the MOM based inter-process communication protocols were further classified as Message Queuing Telemetry Transport , Zero MQ Transport Protocol and Advanced Message Queuing Protocol. After a thorough comparison but with few trades off, Message Queuing Telemetry Transport was first integrated into the raspberry pi, and simple pro-typing of message exchange between raspberry pi was validated as shown in figure 2.5 by setting the server as mosquitto message broker.

Secondly, the Panda Power software compatibility with the RasPi was checked, it was found that the computational power of Raspberry PI (RasPi) needs to be increased by using a numerical package of python but there was some error in the processing the same. It also needs to be corrected for increasing the speed of the raspberry pis. After the testing of integration of Panda Power software with Cigre LV European network as detailed in session 4.1.1.1.

Next, the both the Message Queuing Telemetry Transport and Panda Power package was tested in combination between different Raspberry PI (RasPi)'s with the data set as described in session 4.1.1.2. After that, the battery model as described in session 4.1.1.3 was made in Open Modelica software and converted in Functional Mock UP Units, integrated into the raspberry pi with the help of Energy Sim.

Finally, access criteria were set for all the three components namely distribution, controller and the battery load. Where, the distribution has the access to the power system modelling package Panda Power and power flow calculations, handle all the connection request from controller and battery, sends restricted power flow output to both controller and the battery.

The congestion scenario was introduced in the Cigre Low Voltage (LV) European network by tweaking it, as shown in figure 4.2. The battery helps in resolving the congestion by peak shave mechanism as detailed in chapter 4.

The only difference between this application-centric model and the model constructed as shown in chapter 3 is that there is a presence of a middleware system package which help to instrument the application layer and it is generic and can be reconfigurable, scalable, reliable, distributed in nature. Therefore, We can appreciate the advantage of a middleware system in instrumenting the Illuminator - Energy System Integration Development Tool Kit.

Without, the middleware system, for every scenario, all the functionalities should be defined and integrated. But, with the help of the developed middleware package midway energy, it would be easy to reuse the function, overwrite them and define few more classes, functionalities based on the arising requirements. Therefore, Middleware system would be the most vital layer of Illuminator - Energy System Integration Development Tool Kit.