IJCRT.ORG ISSN : 2320-2882

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)

An International Open Access, Peer-reviewed, Refereed Journal

ILENS

[1]M. Jahnavi, [2]M. Pooja Chowdary, [3]S. Mohammad Muzmil, [4]M. Harish Raju, [5]C. Narasimha

[1,2,3,4]Student, [5]Assistant Professor

Computer Science and Engineering,

Madanapalle Institute of Technology and Science, Madanapalle, India

Abstract: iLens provides a unique platform for smart combination with several devices/sensors in large enterprises, manufacturing industry, home, commercial properties etc. It will provide an MQTT (Message Queuing Telemetry Transport) interface for the unified integration of various sensor devices in the ground to capture time-series data in real-time. Based on pre-configured rules by the users, it will be able to generate alarms and alerts based on the rules. It uses artificial intelligence and machine learning to co-relate events received from various sources or devices to find interesting patterns in the data and generate intelligent insights. It can discover any device in an automated manner (automated device discovery), over-the-air upgrades and updates, data logging, live device monitoring, and security. With its ability to drag and drop GUI (Graphical User Interface) components to build various alerts and rules, iLens simplifies orchestrating and managing IoT devices.

Keywords **– MQTT, alerts, alarms, data logging, live device monitoring, Unique platform**

I. INTRODUCTION

iLens is the Industrial IoT and AI Platform for industry 4.0 facilitating smart integration with various sensors and devices to perform device management, edge computing, decision making, and real-time(live) streaming using predictive analytics. Intelligent Lens is a web-based platform that is used to monitor or manage the power usage of any remote device that is connected to the internet. In this fast-growing world, there is a need to manage renewable resource like energy. Energy management is the means to reducing and controlling one's organization energy or power consumption. It enables you to reduce costs which is becoming increasingly important as energy costs rise.

Connect: -

- Data logging, Live device monitoring and security
- Hardware and Protocol agnostic platform
- Automated device discovery as well as auto-provisioning

Integrate: -

- Integrates sensors, IoT devices and Enterprise or industrial data in real-time with pre-built connectors.
- State-of-the-art in-memory unique platform to crunch consumed data.

Analyze: -

- Institutionalize, monitor and adapt the insights using configurable events, alarms and business rule driven workflows.

Actionize: -

- Visualize Insights with built-in tools or using existing BI (Business Intelligence) investments.
- Build predictive models and time series forecasts right out of the Analytics workbench.

There will be no use in the advancement of technology if it won't solve real-life problems. The most valuable thing in the world is life. If life can be saved with the help of technology, then technology will be the long-lasting technology in the world. The key features of the project are the following:

- Automated Device Discovery.
- Realtime alerts and alarms with SMS and Email integration.
- Remote controls and configuration of devices.

II. LITERATURE SURVEY

Alfiya Abubaker et al. concluded that the degradation of water resources has become a common problem. The conventional methods of water quality monitoring involve the manual collection of water sample from different locations [1].

A. Python

Python is a popular programming language. It was created by Guido van Rossum, and released in 1991. It is used

- On a server to create web applications.
- As software to create workflows.
- Can connect to database systems. It can also read and modify files.
- To handle big data and perform complex mathematics.
- For rapid prototyping, or for production-ready software development.

Why Python?

- Python works on different platforms (Windows, Mac, Linux, Raspberry Pi, etc.,).
- Python has a simple syntax similar to the English language.
- Python can be treated in a procedural way, an object-oriented way or a functional way.

Good to know

- The most recent major version of Python is Python 3, which we shall be using in this tutorial. However, Python 2, although not being updated with anything other than security updates, is still quite popular.
- In this tutorial Python will be written in a text editor. It is possible to write Python in an Integrated Development Environment, such as Thonny, Pycharm, Netbeans or Eclipse which are particularly useful when managing larger collections of Python files.

B. Data Base

The database used is Mongo DB. MongoDB is a No SQL database. It is an open-source, cross-platform, document-oriented database written in C++. In simple words, you can say that - Mongo DB is a document-oriented database. It is an open-source product, developed and supported by a company named 10gen. It is available under General Public license for free, and it is also available under Commercial license from the manufacturer. The manufacturing company 10gen has defined MongoDB as:

"MongoDB is a scalable, open source, high performance, document-oriented database." - 10gen

MongoDB was designed to work with commodity servers. Now it is used by the company of all sizes, across all industry. Purpose of building MongoDB. All the modern applications require big data, fast features development, flexible deployment, and the older database systems not competent enough, so the MongoDB was needed. The primary purpose of building MongoDB is:

- Scalability
- Performance
- High Availability
- Develop Faster
- Scaling from single server deployments to large, complex multi-site architectures.

Features of MongoDB

1. Support ad hoc queries

   In MongoDB, you can search by field, range query and it also supports regular expression searches.

2. Indexing

   You can index any field in a document.

3. Replication

   MongoDB supports Master Slave replication. A master can perform Reads and Writes and a Slave copies data from the master and can only be used for reads or back up (not writes)

4. Duplication of data

   MongoDB can run over multiple servers. The data is duplicated to keep the system up and also keep its running condition in case of hardware failure.

5. Load balancing

   It has an automatic load balancing configuration because of data placed in shards.

6. It also supports:

   JSON data model with dynamic schemas

Jasmin Guth et al. described that the IoT is gaining increasing attention. The overall aim is to interconnect the physical with the digital world. Therefore, the physical world is measured by sensors and translated into processible data, and data has to be translated into commands to be executed by actuators [3].

C. Docker

Docker is a container management service. The keywords of Docker are develop, ship and run anywhere. The whole idea of Docker is for developers to easily develop applications, ship them into containers which can then be deployed anywhere. The initial release of Docker was in March 2013 and since then, it has become the buzzword for modern world development, especially in the face of Agile-based projects.

Features of Docker:

- Docker has the ability to reduce the size of development by providing a smaller footprint of the operating system via containers.
- With containers, it becomes easier for teams across different units, such as development, QA and Operations to work seamlessly across applications.
- You can deploy Docker containers anywhere, on any physical and virtual machines and even on the cloud.
- Since Docker containers are pretty lightweight, they are very easily scalable.

Components of Docker

- Docker for Mac − It allows one to run Docker containers on the Mac OS.
- Docker for Linux − It allows one to run Docker containers on the Linux OS.
- Docker for Windows − It allows one to run Docker containers on the Windows OS.
- Docker Engine − It is used for building Docker images and creating Docker containers.
- Docker Hub − This is the registry which is used to host various Docker images.
- Docker Compose − This is used to define applications using multiple Docker containers.

Mourvika Shirode et al. explained that the conventional method of testing water quality is to gather samples of water manually and send to the lab to test and analyze. This method is time consuming, wastage of man power, and not economical [2].

D. Flask URL Building

The url_for() work is remarkably helpful for powerfully fabricating a URL for a specific capacity. The capacity acknowledges the name of a capacity as first disputation, and at least one catchphrase contentions, each comparing to the variable piece of URL. Flask is a small and lightweight Python web framework that provides useful tools and features that make creating web applications in Python easier.

It gives developers flexibility and is a more accessible framework for new developers since you can build a web application quickly using only a single Python file. Flask is also extensible and doesn't force a particular directory structure. It uses the Jinja template engine to dynamically build HTML pages using familiar Python concepts such as variables, loops, lists, and so on. You'll use these templates as part of this project.

E. Kubernetes

Kubernetes in an open-source container management tool hosted by Cloud Native Computing Foundation (CNCF). This is also known as the enhanced version of Borg which was developed at Google to manage both long running processes and batch jobs, which was earlier handled by separate systems. It comes with a capability of automating deployment, scaling of application, and operations of application containers across clusters. It is capable of creating container centric infrastructure.

Features of Kubernetes

- Continues development, integration and deployment
- Containerized infrastructure
- Application-centric management
- Auto-scalable infrastructure

III. ARCHITECTURE

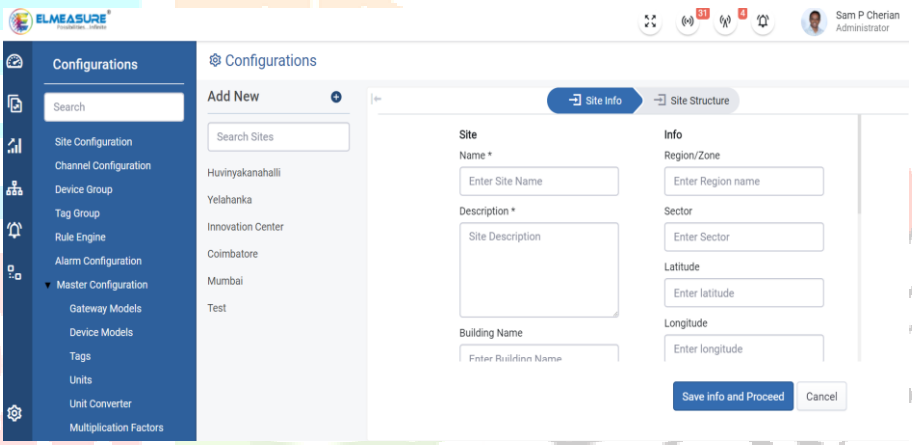

Figure 1: iLens overview

MQTT runs on top of TCP/IP using a PUSH/SUBSCRIBE topology. In MQTT architecture, there are two types of systems: clients and brokers. A broker is the server that the clients communicate with. The broker receives communications from clients and sends those communications on to other clients. Clients do not communicate directly with each other, but rather connect to the broker. Each client may be either a publisher, a subscriber, or both. MQTT is an event-driven protocol. There is no periodic or ongoing data transmission. This keeps transmission to a minimum. A client only publishes when there is information to be sent, and a broker only sends out information to subscribers when new data arrives.

IV. IMPLEMENTATION

MQTT

MQTT refers to Message Queuing Telemetry Transport. MQTT is a pub-sub (publish/subscribe), lightweight and extremely simple messaging protocol, intended for low-bandwidth and controlled devices, unreliable or high-latency networks. The design ideologies are to minimize device resource requirements and network bandwidth at the same time as also attempting to ensure some degree of assurance and reliability of delivery. These principles turn out to make the protocol ideal of the evolving "Internet of Things" or "machine-to-machine" (M2M) world of connected devices, and for mobile applications where battery power and bandwidth are at the best.

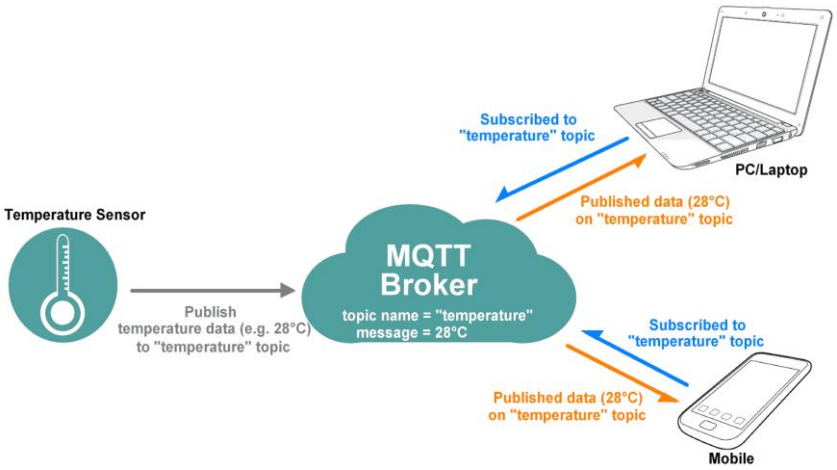

Figure 2: MQTT work flow

Overview:

The Message Queuing Telemetry Transport system contains of clients interacting with a server, also known as "broker". A client can be either a subscriber of information or a publisher of information. Publisher and subscriber both can connect to the MQTT broker. Information that was published by publisher to a particular topic is organized in a hierarchy of topics. When a publisher wants to publish a new item of data to distribute, it will send a control message along with the data to the connected broker(server). The broker further distributes the information that was taken by the publisher on a particular topic to any subscribers(clients) that have subscribed to that topic.

The publisher doesn't need to any type of data on the number of subscribers or locations of subscribers, and subscribers in turn need not to know about the publishers and do not have to be configured with any data about the publishers. On the off chance that an agent gets a theme for which there are no present supporters, it will dispose of the point except if the distributer demonstrates that the subject is to be held. This enables new endorsers of a theme to get the most current esteem instead of sitting tight for the following update from a distributer.

At the point when a distributing customer initially associates with the agent, it can set up a default message to be sent to endorsers if the representative distinguishes that the distributing customer has surprisingly disengaged from the merchant. Customers just associate with a representative, yet a framework may contain a few specialist servers that trade information dependent on their present supporters' subjects. An insignificant MQTT control message can be as meagre as two bytes of information. A control message can convey about 256 megabytes of information if necessary. There are fourteen characterized message types used to associate and disengage a customer from a specialist, to distribute information, to recognize receipt of information, and to direct the association among customer and server.

MQTT depends on the TCP convention for information transmission. A variation, MQTT-SN, is utilized over different transports, for example, UDP or Bluetooth. MQTT sends association qualifications in plain content organization and does exclude any measures for security or verification. This can be given by the fundamental TCP transport utilizing measures to shield the honesty of exchanged data from capture attempt or duplication. The letters "MQ" in "MQTT" originated from the IBM MQ (then 'MQSeries') message queuing product line. Nevertheless, queuing itself is not at all required to be supported as a standard feature in all circumstances.

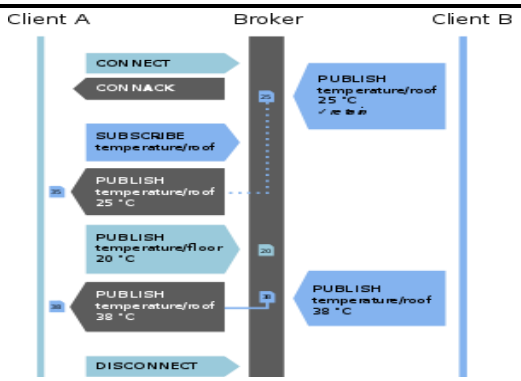

Figure 3: MQTT example

V. RESULTS AND ANALYSIS

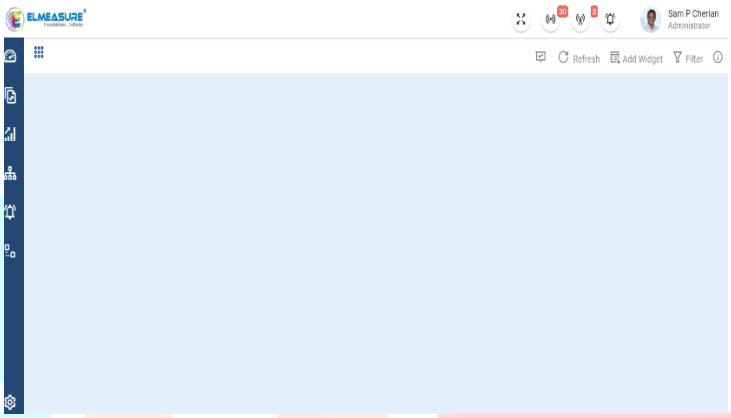

Figure 4: ELMeasure overview

The above figure depicts the outline of the project. The left most side bar will have all the features and specifications like

A. Configurations

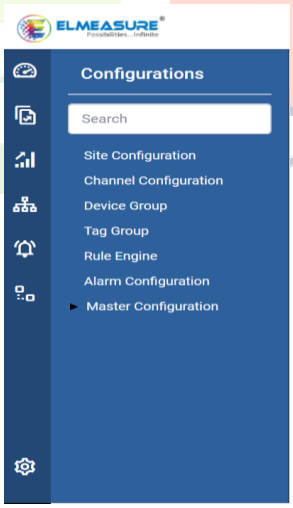

Figure 5: Configurations side bar

In order to fetch data from industry (IoT devices) and process through pipeline by using python, we need to configure

- Site Configuration
- Channel Configuration
- Device Group Configuration
- Tag Group Configuration
- Rule Engine
- Alarm Configuration
- Master Configuration

B. Reports

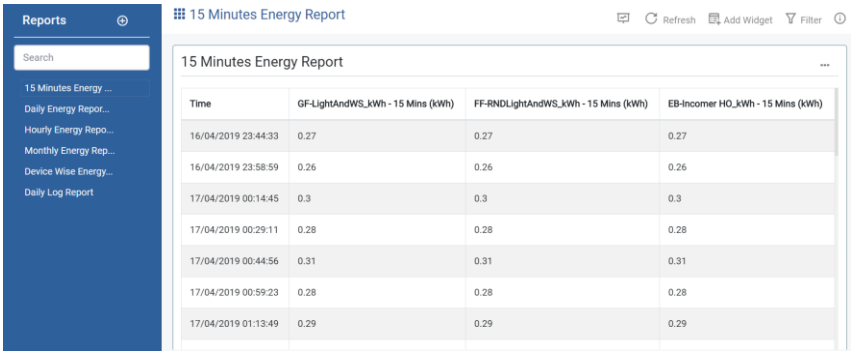

Figure 6: 15 Minutes Energy Report

As shown in the above figure, user can use this report to leverage the power usage in their industry. The above report will clearly show that much watts of power is consumed on which time for each device (light, RND light)

C. Trends

Figure 7: Trends in EIMeasure

Trends are very important for this project as it will give the chart like picture in order to analyze what trend is going on a particular device. For instance, we have AC which is configured on one industry and it's consuming some power. We can know how much power it is consuming in a live chat.

D. Dashboards

Dashboard is the main and core objective of the whole project. It shows the interesting patterns and facts about power usage of any device which is configured by the user in its respective industry.

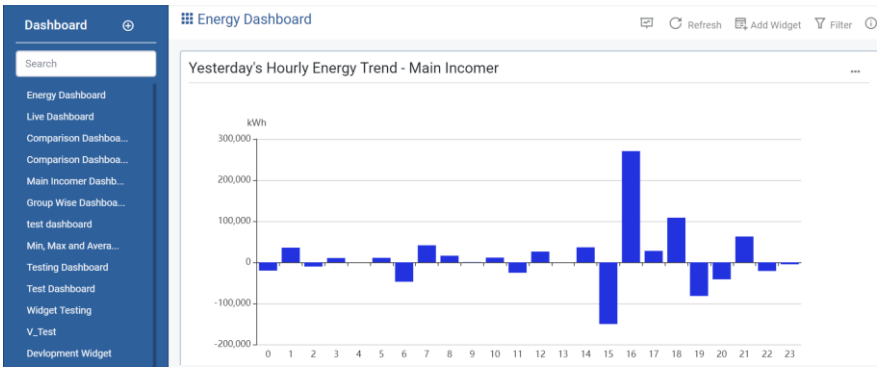

Figure 8: Dashboards

In the above figure the left side bar will have all dashboard that was configured by the user in order to know about the interesting facts about its devices.

E. Widgets

We can configure the widgets for any dashboard that we wish. Widget can be of any type like line chart, bar chart, scatter plot chart and pie chart. The below widget is of type bar chart for particular device.

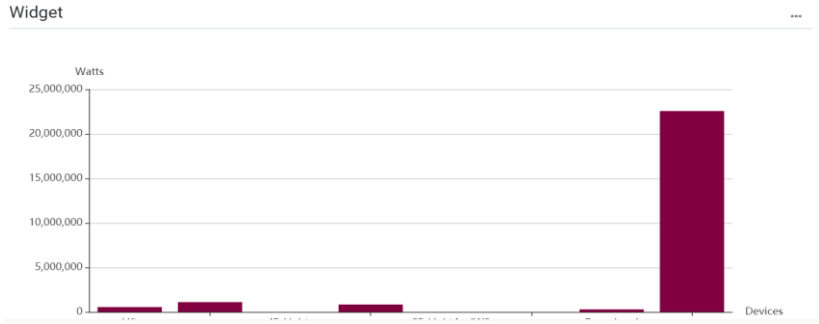

Figure 9: Sample Widget

VI. Conclusion

Energy management is the good practice of using energy more efficiently and effectively in an organization's operations. Energy is a valuable resource and a cost that can be controlled when we managed it effectively and efficiently. Energy management provides an opportunity to optimize energy costs by understanding energy flow as well as procurement and economics of energy, and reduce its harmful impact on our environment. It is an ongoing process and must be reviewed at regular intervals and fine-tuned as required, from time to time as and when required.

Future Enhancement:

- Elmeasure 1st version has been released on 02nd April 2019. It has sent to testing stage. If any bugs raised during testing, we need to fix out those services.
- Currently application is not stable one as it is still in development stage. Group of 2 members writing web services in python.
- We are using GitLab to store our code repositories. We have multiple branches like developer and master branch.

Acknowledgment

We would like to thank Mr. C. Narasimha (Assistant Professor, Dept of CSE, MITS, Madanapalle) for his valuable suggestions and advice that helped us to improve this works, and his support is greatly appreciated.

References

[1] Alfiya Abubaker, Kavya C R, Tilju Thomas, Nikhil Joseph, A project report on "A Study on IOT Approach for Monitoring Water Quality Using MQTT Algorithm" guided by Shifana Begum, Dept. of CSE, Srinivas School of Engineering.

[2] Mourvika Shirode, Monika Adaling, Jyoti Biradar, Trupti Mate, A project report on "IOT Based Water Quality Monitoring System", Department of Electronics & Telecommunication Keystone School of Engineering, Pune, Maharashtra, India.

[3] MangaloreSuhas K, Suhas N, Sumukh B, Sunil S, A project report on "Public distribution system" guided by Mrs. S Mala, Department of Electronics and Communication, SIT Tumakuru 2015-16.

[4] Jasmin Guth1, Uwe Breitenbücher1, Michael Falkenthal1, A project report on "A Detailed Analysis of IoT Platform Architectures: Concepts, Similarities, and Differences", Institute of Architecture of Application Systems, University of Stuttgart, Germany.

[5] https://www.abettercity.org/docs/sustainability/Effective%20Energy%20Managment.pdf

[6] https://www.knowledgelens.com/Products/iLens/

[7] https://www.w3schools.com/python/python_mongodb_getstarted.asp

[8] http://kairosdb.github.io/

[9] http://kairosdb.github.io/docs/build/html/QueryingData.html

[10] http://cassandra.apache.org/