Industrial IoT Gateway Configurator

M. Prathyusha[1], P. V. S. Maruthi Rao[2] and Dr. N.V. Rao[3]
[1]Asst. Professor, CVR College of Engineering/IT Department,Hyderabad, India
Email: prathyushareddy2008@gmail.com
[2]Director, Vidcentum R&D, Hyderabad, India
Email: hr@vidcentum.com
[3]Professor, CVR College of Engineering/CSE Department, Hyderabad, India
Email: nvr@ieee.org

***Abstract:* Internet of Things (IoT) is a fast-growing phenomenon that attracted the attention of researchers and many business organizations. It has enhanced the demand for connecting various networks that include heterogeneous networks with different protocols. From this point of connectivity through networks, sensors have a limited capability as they cannot offer connectivity to complex and large networks such as internet. The present paper is based on the experience in the design and implementation of an industrial IoT gateway configurator for real time monitoring that enhances the performance of assets. It includes configuration of the gateway, asset and transducer to build solutions efficiently in various areas like agriculture, industry, health, transportation, etc.,**

***Index Terms:* IoT, an Asset, the Gateway, Sensor, a Web-socket connectivity protocol**

I. Introduction

Now-a-days everything connects to the Internet, which is growing to send information to servers for processing, storage and to provide full control of physical devices over the web. It is become medium for everyday communication.

The Thing- A device/ a conceptual device becomes a thing if it is at-least the following:

a) Self-described b) Reconfigurable c) Interacting with some decision systems d) Autonomous e) Safe and Secure f) Uniquely identified g) Connected h) Reliable i) Resilient.

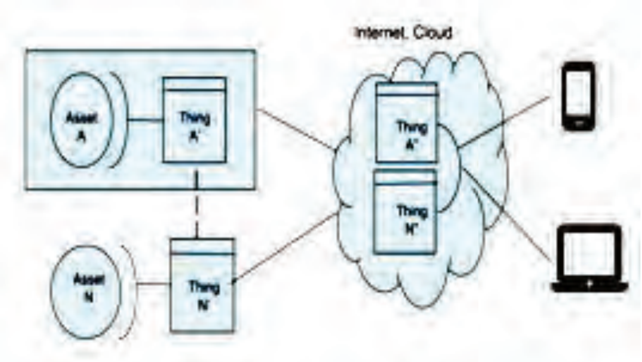


Figure 1. Internet of Things

- In Fig. 1, shown are the Assets A and N connected as A", N" in the cloud.
- This is the crux of IoT designs. Once the software objects of Assets are present in the "Compute", environment, it opens doors of "Connected" Applications.

The Internet of things (IoT) is the network of physical devices, home appliances and other items embedded with electronics, software, sensors, actuators and transducers to setup the objects, which communicates and exchanges information [1][2][3]. Each thing has unique identification which collaborates with the existing Internet infrastructure. Objects are remotely sensed across existing network frameworks, to create opportunities for direct integration of the physical world into computer-based systems and the results improved according to accuracy, efficiency and economic benefit.

A. IoT Edge Analytics Gateway

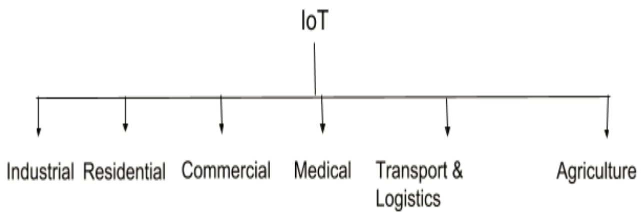


Figure 2. Internet of Things

- As shown in Fig. 2, IoT is applicable to all business where "Assets" are connected.
- IoT is a new channel to observe and control the industrial production.

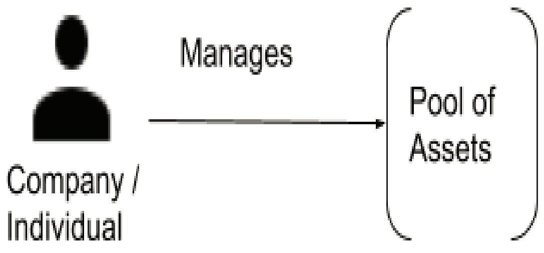


Figure 3. Asset Overview

- As shown in Fig. 3, a business manages "Assets" and creates value; it is very obvious that those "Assets" are subjects of IoT.

B. Asset Management System

Asset management includes IoT Edge Gateway and Cloud Decision Support applications.

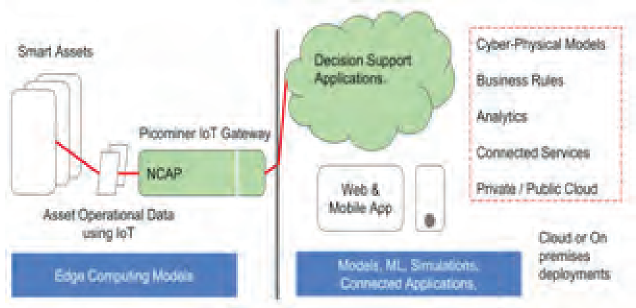


Figure 4. Asset Management System

NCAP (Network Capable Application Processor)

As depicted in Fig. 4, the NCAP Runs on the SBC (Single Board Computer) such as Raspberry Pi, Beaglebone etc., interfaces with Transducer Boards (Anybus Communicators, Texas Instruments, Arduino, Custom Boards). It handles millions of data points at the gateway and supports a variety of communication protocols.

II. Design And Implementation

A. *Gateway Architecture*

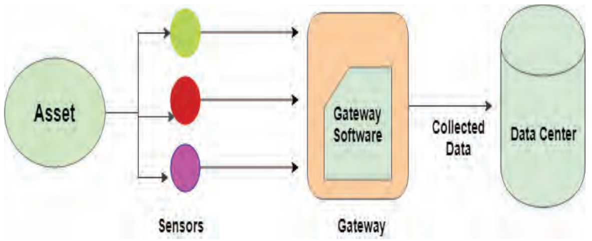


Figure 5. Gateway Architecture

The input of the application is an asset and the output are to configure the gateway by monitoring the asset. Device-to-gateway communication model is used in-order to implement and configure the IoT gateway.

As shown in the Fig. 5, the gateway architecture is the most prevalent architectural design where the assets are attached to sensors to monitor the asset resource but gateway itself is not setup the sensors, the gateway software which is installed on the device is held for collecting information from the various sensors, pre-processing that data and sending the results of the collected data to the data center with the help of web-socket connectivity protocol.

Asset

Asset is an input resource controlled by the thing. Assets like industrial floor, packaging machine whose are attached with sensors to monitor the asset data.

Sensor

Sensor is a device which is used to identify the changes in an environment in the system. Sensors like temperature, pressure, weight etc., Major role of sensors is to hit the physical phenomenon and converts it into an electric signal.

Gateway

Gateway acts as an interface between the sensors and cloud servers. The Gateways can connect to the IoT devices through web-socket connectivity protocol, store and parse the data and then send them over to cloud servers for processing and analytics. The IoT gateways not only acts as an interface for communication but also provides the security in the form of unique id for the gateways while transmitting the information.

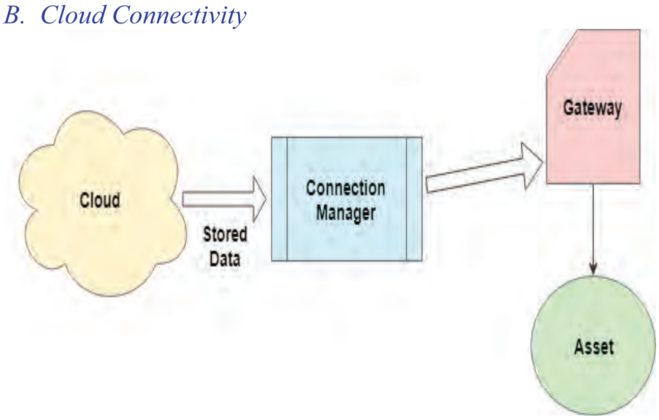
The Gateway Software

The gateway software is responsible for collecting the data from sensors and sending them to the data center for pre-processing the data and it will decide that at any stage of processing the data in temporary, kept in-memory. The gateway software application is the heart of the gateway. It will handle the failures like when power is off while using the application, it should be bootstrapped and started automatically as power returns to the device, and it should continue to work where it was interrupted.

Device Management and Configuration

In this architecture, an IoT gateway handles up to eight different types of sensors and their data. To store the connected sensor information, IoT Gateway device is required. In addition to this, IoT Gateway manages and configures the information of all the devices.

B. *Cloud Connectivity*



Cloud <> Gateway (for decision events)

Figure 6. Cloud Connectivity Diagram

Data Communication Protocols

The IoT Gateway connects with the cloud over web-socket connectivity protocol to stream the input data. Two-way communication channel is established from asset to the cloud for exchanging of data to get the decision events.

Cloud Connectivity Manager

The connection manager is responsible for consistent connectivity from the gateway to the cloud. It handles the reconnection when power is in on/off state and maintains the heart-beat message from the cloud to gateway device through web-socket connectivity protocol. The same has been displayed in Fig. 6.

Custom Application

IoT Gateway application is designed as per business needs. The Gateway software application interacts with services, functions to manage data between sensor node and the gateway to the cloud in an efficient, secure and responsive manner.

C. Platform Model

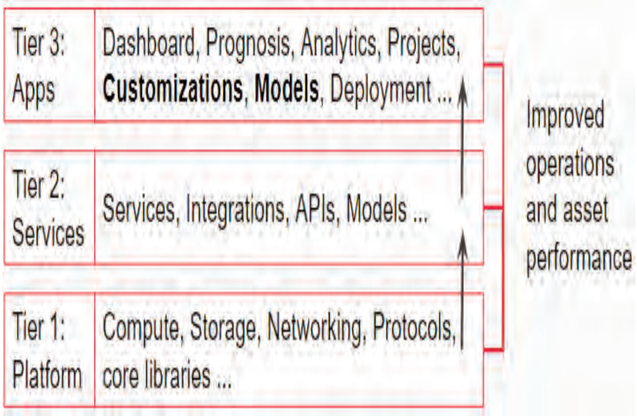

Tier 3:
Apps
Dashboard, Prognosis, Analytics, Projects, Customizations, Models, Deployment ...
Tier 2:
Services
Services, Integrations, APIs, Models ...
Tier 1:
Platform
Compute, Storage, Networking, Protocols, core libraries ...
Improved operations and asset performance

Figure 7. Tier 1, 2 are core platform offerings. Tier 3 (Apps) is a service layer.

Tier 1 – Web-Socket Connectivity Protocol [4]

From Fig. 7, web sockets allow two-way (full-duplex) communication between asset and cloud through a single connection [5][6]. The protocol is helpful to reduce unnecessary network traffic and latency. Web-Socket is used to communicate and exchange data at the same time. This protocol is used to optimize the performance, resulting in more responsive web applications. The data which is collected from the gateway is sent to the cloud using the web-socket connectivity protocol, the protocol will stream all the data, pre-process it and send back to the gateways and assets using the same connectivity protocol. RFC 6455 is the latest specification of Web Socket protocol, it is supported in all browsers like Mozilla Firefox, Internet Explorer, Google Chrome, etc.

Following is the API which creates a new Web-Socket object.

var socket = new Web-Socket(url, [protocol]);

The first argument url, specifies the URL (Uniform Resource Locator) to connect. The second argument, protocol is used to connect the server.

Tier 2 - Restful Services of API

“Reference [7]” proposed an asynchronous RESTful web service recursively. To invoke the RESTful web services from the Internet of Things, there will be various methods explained. To access the data for a resource is via HTTP URI, in-order to accept the input data is via PUT method and to access output data is via GET method [8]. To minimize latency and network communication, RESTful web services will be helpful.

APIs Code to login in to the application (client side)

This API is developed using typescript.

```
    // Let's Login to the Portal (Application) using the APIs.
Const mylogin = {

    'email' :emailId,

    'password' : passwd

    };

    this.headers = new HttpHeaders({'Accept':
'application/json','Content-Type':'application/json'
, 'Access-Control-Allow-Origin': '*'});
    this.httpClient.post<any>(
`https://dashboard.picominer.com/es/login,
    mylogin,{ headers: this.headers}).subscribe(
    // callback
    (Success)(val) => {
    if (val === null || val === undefined) {
            // Something wrong. Check the API.
            return;
    }
    // Let's extract the data from the API response.
If(val['api-response']['header']['result'] === null ||
val['api-response']['header']['result'] === undefined) {
    return; }
    // Portal API returns success (result = 0) if everything is
OK.
    if (val['api-response']['header']['result'] !== 0)
    {
    constappError=val['api-response']['header']['response'];
this.stopPortalSpinner();return; }
    // Now set the Application State Variables.
this.appState.setUserData(val['api-
response']['header']['email'],val['api-
response']['header']['login],val['auth-token'],val['api-
response']['header']['accessed_at'],`${new
Date().toLocaleTimeString()}`);
    // User has logged into the Portal.. Redirect to Dashboard
pages.
    Returnthis.router.navigate(['/site-analytics']);
    }
```

Tier 3: Dashboard

Node.js

To use Angular framework, it requires Node.js version 8.x or 10.x. Node.js is an open source JavaScript engine developed by Google on top of V8 [9]. It is efficient for real-time applications because it uses an event-driven model. The features and functionalities of Angular, the Angular CLI (Command Line Interface) and Angular apps are provided by libraries which are available as npm packages, npm package manager must be required to download and install npm packages.

Based on the package.json file information, Node Package Manager (NPM) is included inside to provide the installations of the third-party modules and dependencies.

Data Visualization (client side)

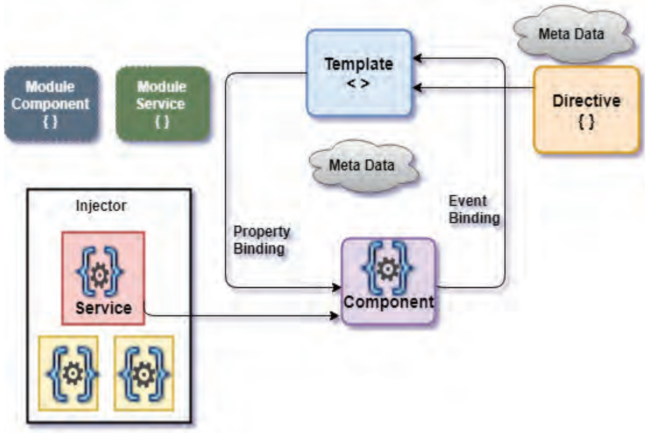


Figure 8. Angular JS Architecture [10]

To develop client side of web applications, Angular framework is very helpful [10]. To build angular applications, HTML, JavaScript language is used. Angular framework is in type script language which then compiles into java script language. This framework contains plenty of built-in libraries that some installed by default and others are optional.

As given in the Fig. 8, Angular JS architecture shows the component classes which are created to manage the templates, application logic is added in services, components and services are included in modules to inject the objects and classes. The main advantage of angular JS framework is routing i.e. navigating from one page to other page without reloading the entire page and binding the data in terms of property, event. Directives are helpful in an application in-order to perform the actions in an application.

Angular 6 Portal Application Example -Installation of Angular6

To set up the node.js in a device, for using the angular framework.

Step 1: Go to required directory then install the following commands:

```
npm install –g @angular/cli –save
ng new filename
```

The project will be created with some components and modules in type script.

For web-socket connectivity to cloud, install the following dependency:

```
npm install -S rxjs-websockets
npm install -S queuing-subject
```

The following sample code is the web socket connectivity to the cloud to stream the input data.

Code to connect the server using angular framework

Connect to the server with the help of web-socket connectivity protocol using the address **'wss://dashboard.picominer.com:7681'**.

```
public connect() {
if (this.messages)
return
// Using share() causes a single web-socket to be created when the first observer subscribes.
// This socket is shared with subsequent observers and closed when the observer count falls to zero.
this.messages = websocketConnect(
'wss://dashboard.picominer.com:7681',
this.inputStream = new QueueingSubject<string>()
).messages.share()
}
public send(message: string):void {
// If the websocket is not connected then the QueueingSubject will ensure
// that messages are queued and delivered when the websocket reconnects.
// A regular Subject can be used to discard messages sent when the websocket
// is disconnected.
this.inputStream.next(message)
}
```

First, connect to the server using web-socket connectivity protocol and try to subscribe to the portal with 34 characters of length.

```
ngOnInit() {
this.socket.connect();
this.socketSubscription =
this.socket.messages.subscribe((message: string) => {
console.log('received message from server: ', message);
});
// send message to server, if the socket is not connected it will be sent
this.socket.send('prathyushaprathyushaprathyushahello');
}
```

D. Design of Portal Application to Configure the Gateways

Fig. 9, shows the flow of the gateway, transducer and asset configuration to the cloud.

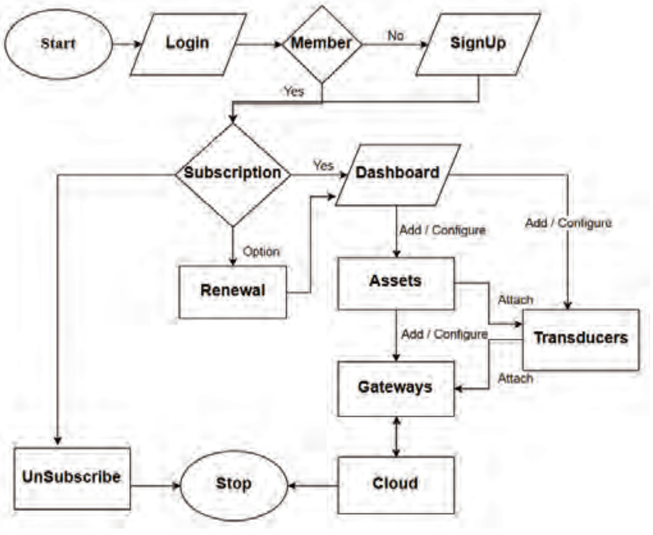


Figure 9. Flow Chart Diagram

Steps for Application:

1. First, login to the application using the **https://dashboard.picominer.com/v/2.0/#/login** with an email id and a password, otherwise register to the application.
2. After login into the application with proper authentication, dashboard will be displayed with sites, HMI, knowledge base, assets, gateways, transducers and account settings.
3. The user will select the site if required.
4. And the user will be add/configure the assets, gateways and transducers.
5. After configuration, the real time data for assets, gateways and transducers will be displayed in the form of various charts like radar, pie, line chart with state and status.
6. The application having the support for help and account settings to change password and logout from the portal.

E. Configuration

Configuring the Gateways

// Login to the Portal Application using Application using**dashboard.picominer.com/v/2.0/**

1. There are various gateways like an industrial machine gateway, a sensor node gateway and video processing gateway to add and configure the required gateway.
2. After selecting the required gateway, add the basic information like the name of the gateway, the description, local id, unique id, version id and cloud gateway information like ipv4 address, port number, domain name, like-wise physical gateway. After that, by clicking on the add button, displays a message in a dialogue box whether it is the correct gateway or not in-order to configure the gateway.
3. After adding the required gateway, update the details of the gateway if required, by clicking on configure gateway.
4. Update the gateway information if required and click on apply button then it will be displayed in the real-time gateway data with state, status of the gateway and data in the form of charts like radar, pie and line charts.

Configuring the Transducers

1. There are various transducers like temperature(LM35), pressure(PX40 series) and weight(HX711) to add and configure the required transducer.
2. Then, select the transducer and add the basic information like the name of the transducer, the description, local id, unique id, version id, latitude, longitude and location. Likewise, calibrations, engineering units, interfaces like serial SPI, serial I2C, Wi-Fi and simulations as capabilities, distributors, generators like linear congruential engine for the required transducer. After that, by clicking on the add button, it displays a message in a dialogue box whether it is the correct transducer or not in-order to configure the transducer.
3. After adding the required transducer, update the details of transducer if required, by clicking on calibrate transducer and attach the transducer to gateway and asset, by clicking on configure transducer.
4. Update the information of the transducer if required, then click on apply and close it.
5. And attach the required transducer to the gateway and asset, then click on apply button, it will be visible in real- time gateway data list in the form of various charts.

Configuring the Assets

1. The assets like an industrial floor, the packaging machine, etc., to add and configure the required asset.
2. After selecting the asset, add the basic information like the name of an asset, the description, local id, unique id, version id, latitude, longitude and location. After that by clicking on the add button, it displays a message in a dialogue box whether it is the correct asset or not in-order to configure the asset.
3. After adding the required asset, update the details of an asset if required, by clicking on the configure button.
4. Update the asset information if required and click on apply button then it will be display in the real-time asset data with state, status of asset and data in the form of charts like radar, pie and line chart.

Finally, the overall performance of the information of site analytics shows the equipment availability, equipment efficiency, production quality, overall equipment efficiency, filling volume accuracy, product spill, sustained peak output, batches completed, alerts to staff, data archival, equipment service time and equipment up time with date and time.

The configured gateways like an industrial machine gateway, sensor node gateway and the video processing gateway about the gateway details and state, status of the configured gateway will be visible in the list of real-time gateways.

The following resulted displays show the real-time data of an industrial machine gateway with eight transducers of data in the form of various charts like radar(traffic data), pie(resources), line chart(trends) and information of the device.

Fig. 10, shows the real time data for the configured industrial machine gateway in radar chart i.e. traffic data for gateway:

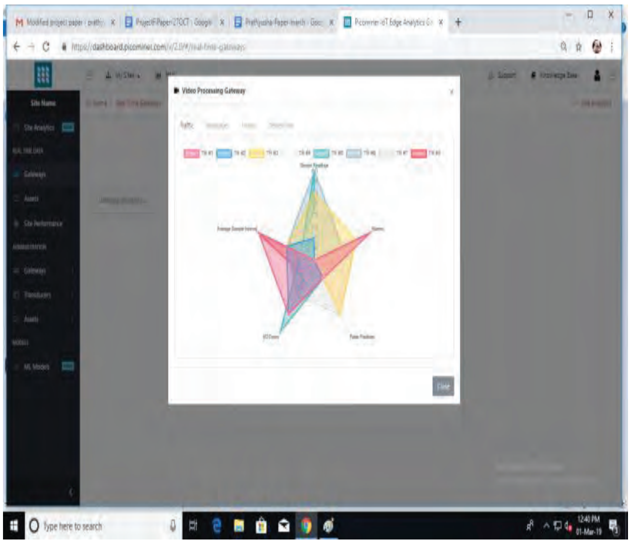


Figure 10. Traffic data of gateway in radar chart

Fig. 11, shows the real time data for the configured industrial machine gateway in pie chart i.e. resources of gateway:

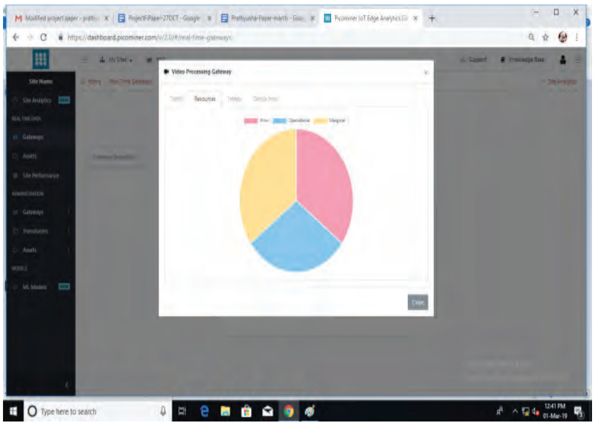


Figure 11. Resources of gateway in pie chart

Fig. 12, shows the real time data for the configured industrial machine gateway in line chart i.e. trends of gateway:

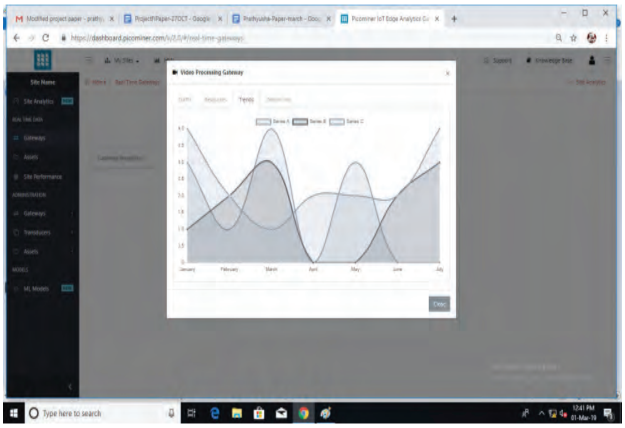


Figure 12. Trends of gateway in line chart

Fig. 13, shows the real time transducer data of event statistics in the form of pie chart and readings of the transducer which are attached to the gateway.

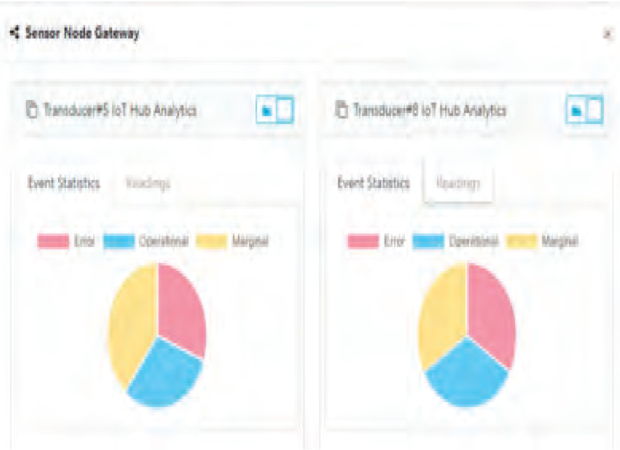


Figure 13. Event statistics of transducer in pie chart

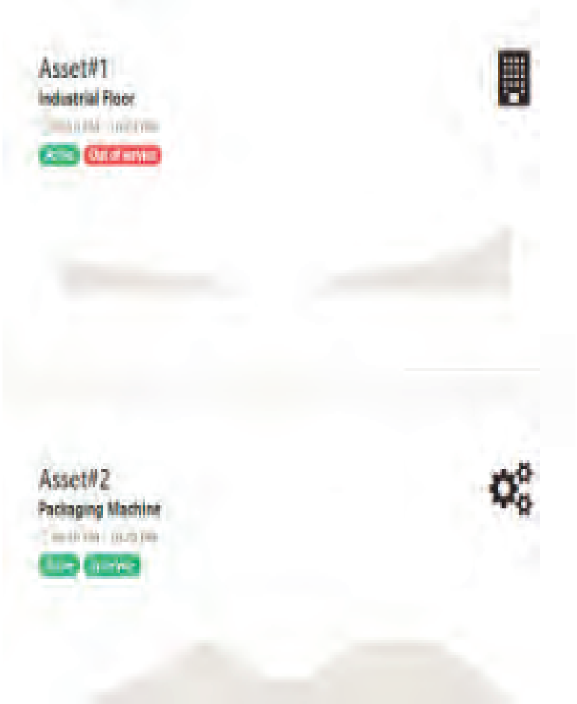


Figure 14. Real time assets data

Fig. 14, show the real time assets data with state and status of industrial floor and packaging machine asset.

III. Conclusions

The IoT has been making our life simple and more convenient through new applications. This paper proposed a practical implementation of an IoT gateway dedicated to real-time monitoring and to improve an asset performance. Thus, Industrial IoT gateway configurator is a crucial business step in building solutions. The elegant solution is to build efficient turn-around time.

Future scope is to implement the millions of sensors, thousands of gateways and hundreds of data centers/ clouds. A view is given as the Fig. 15.

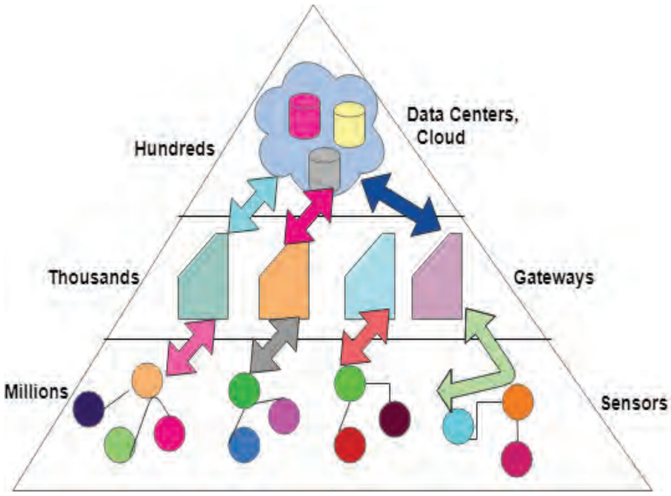


Figure 15. Future IoT Gateway Configurator

REFERENCES

[1] "Internet of Things Global Standards Initiative". *ITU*. Retrieved 26 June 2015.

[2] "Internet of Things Global Standards Initiative". ITU. Retrieved 26 June 2015.

[3] Hendricks, Drew. "The Trouble with the Internet of Things". London Datastore. Greater London Authority. Retrieved 10 August 2015.

[4] Ian Fette, Alexey Melnikov (December 2011). "Relationship to TCP and HTTP".RFC 6455 The Web Socket Protocol.IETF .sec.1.7.

[5] "Glossary:WebSockets". Mozilla Developer Network. 2015.

[6] HTML5 WebSocket: A Quantum Leap in Scalability for the Web

[7] Zhang Lili, et al., "Research on IOT RESTful Web Service Asynchronous Composition Based on BPEL.", Intelligent Human-Machine Systems and Cybernetics (IHMSC), IEEE Sixth International Conference on, Vol. 1, 2014.

[8] Fielding, Roy (June 2014). "Hypertext Transfer Protocol (HTTP/1.1): Semantics and Content". IETF. Internet Engineering Task Force (IETF). RFC 7231. Retrieved 2018-02-14.

[9] Node JS Docs [online]. URL: https://nodejs.org/

[10] Angular Docs [online]. URL: https://angular.io/