VISUAL CONNECT

Scada for IoT with infinite possibility

ALLIED ELECTRICALS
Gala No.21, Lucy Indl. Estate, Near Blue chip Indl Estate,
Sativali,Vasai (E), Maharashtra, India.
vagish@alliedelectricals.com
9029901703

Table of Contents

1 Introduction

Visual Connect is an IoT platform that enables rapid development, management and scaling of IoT projects.

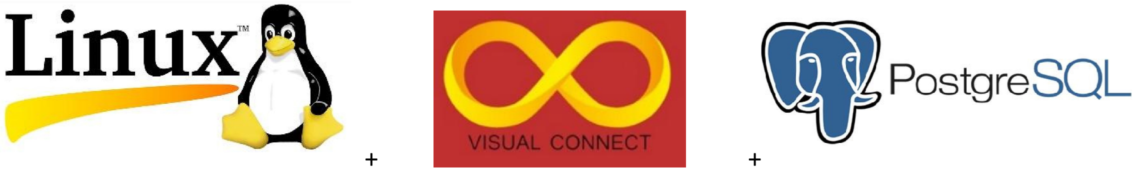
With Visual Connect you are able to:

- Provision and control devices.
- Collect and visualize data from devices.
- Analyze device data and trigger alarms.
- Deliver device data to other systems.
- Enable use-case specific features using customizable rules and plugins.

Visual Connect is:

- **scalable**: horizontally scalable platform, build using leading open-source technologies.
- **fault-tolerant**: no single-point-of-failure, every node in the cluster is identical.
- **robust and efficient**: single server node can handle tens or even hundreds thousands of devices depending on use-case. Visual Connect cluster can handle millions of devices.
- **customizable**: adding new functionality is easy with customizable widgets, rule engine, and plugin system.
- **durable**: never lose your data.

Software can be deployed in cloud services like AWS, Azure etc. Or User can also deploy on company server.



Operating Platform **IoT service** **Database Tool**

Visual Connect is like any other scada available for PLC system, It is application software which can be easily understood by automation engineer.

2 Visual Connect Architecture

Visual Connect is designed to distribute workload across multiple nodes without a single point of failure. Each Visual Connect node is identical and can handle request from both device and server-side applications.

2.1 High level Overview

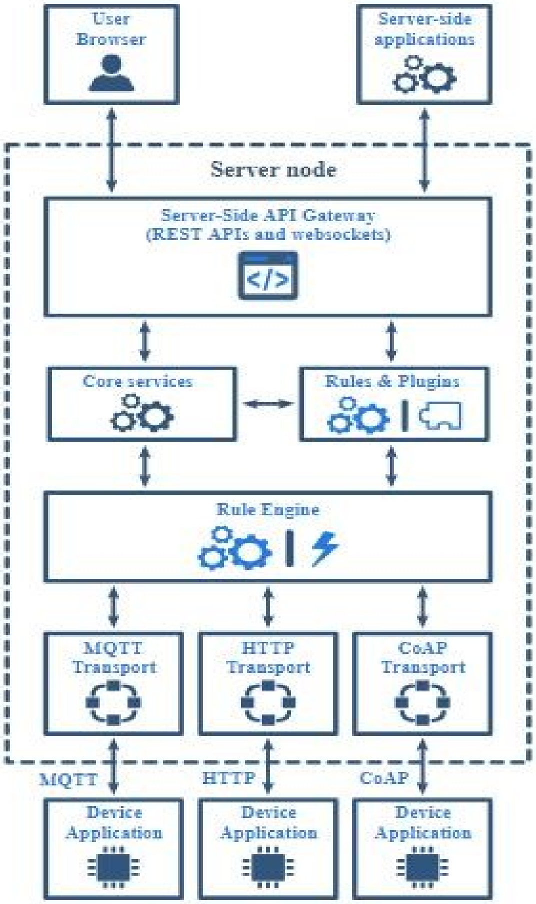


Figure 2.1-1: VISUAL CONNECT architecture

2.1.1 Device connectivity

Visual Connect supports MQTT, CoAP and HTTP protocols for device connectivity. It is possible to plugin support of different protocols or customize existing implementations.

Visual Connect contains set of core services that allow managing the following entities:

- Devices and their credentials
- Rules and Plugins
- Tenants and customers
- Widgets and Dashboard
- Alarms and Events
- Rules and Plugins are able to invoke a certain subset of this APIs. For example, a rule can create an alarm for certain device.

2.2 Features

VISUAL CONNECT handle huge amount of data that’s why security is much more important for us.

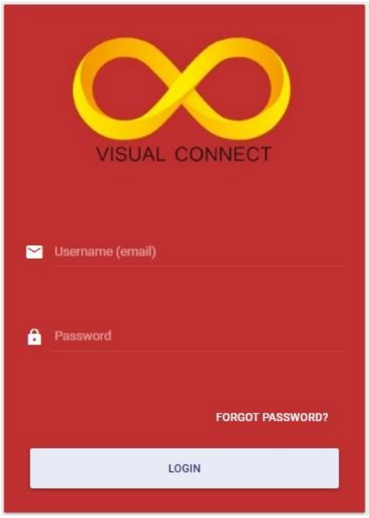


Figure 2.2-1: VISUAL CONNECT Login Page

2.2.1 Transport Encryption

As a system administrator, you are able to configure Visual Connect to use secure sockets layer for HTTP(s) and MQTT transports.

2.2.2 Device authentication

Visual Connect is designed to support many types of device credentials. Current release provides support of token based credentials for all protocols and support of X.509 certificate based credentials for MQTT protocol.

Add device from dashboard window. Once device is created copy the access Token created by visual connect. This Access token is used to call different APIs.

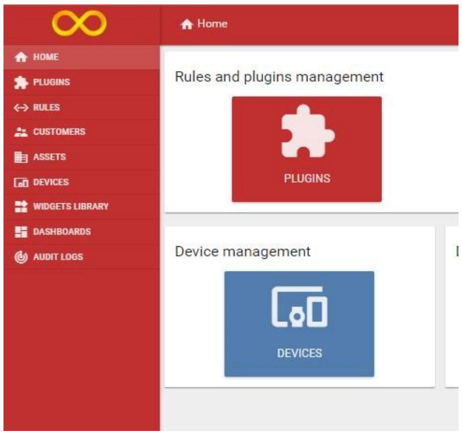
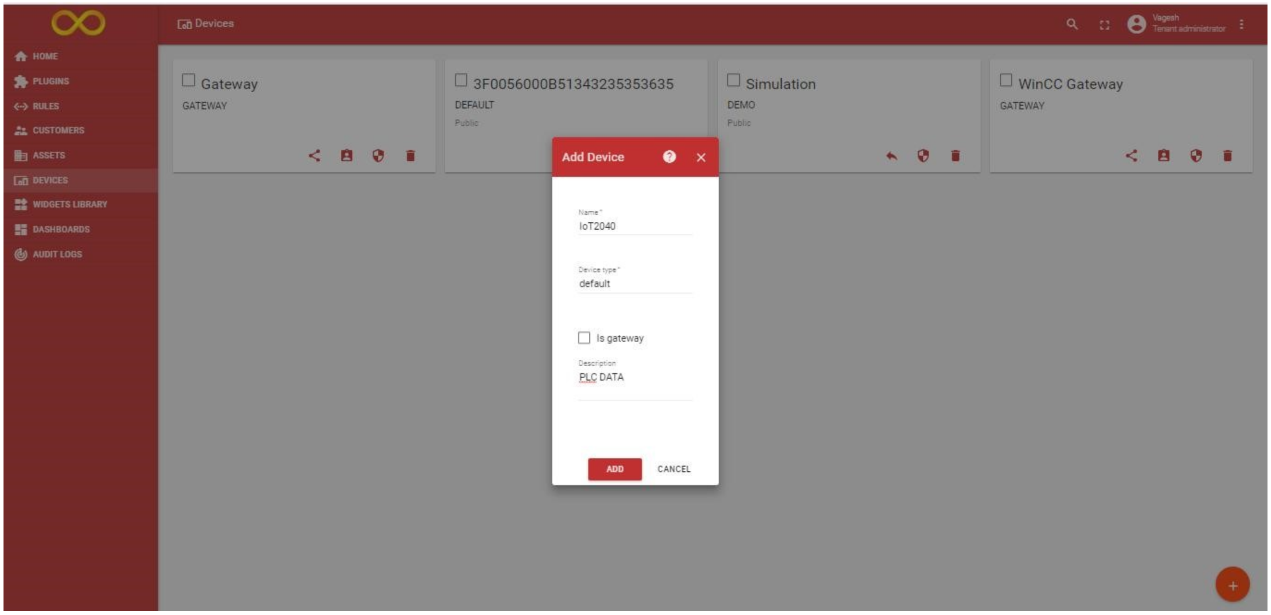


Figure 2.2.2-1: Device View on Dashboard



Add Device

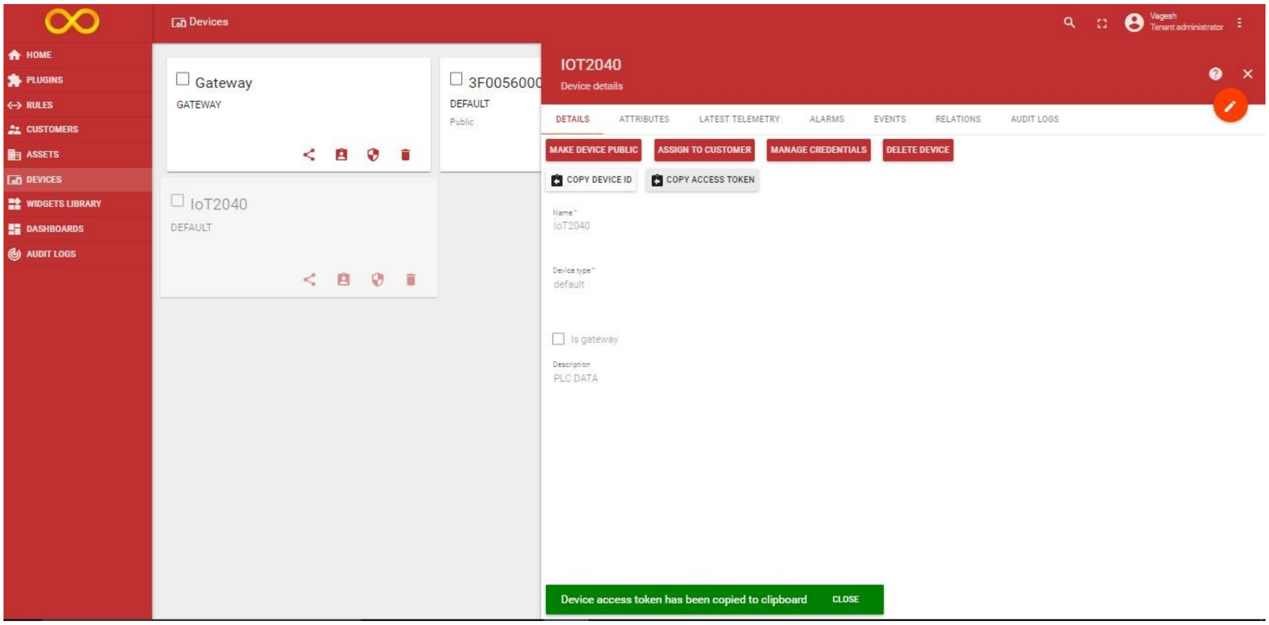
Name* IoT2040

Device type* default

□ Is gateway

Description PLC DATA

Figure 2.2.2-2: Add Device



IOT2040

Device details

Name* IoT2040

Device type* default

□ Is gateway

Description PLC DATA

Device access token has been copied to clipboard

Figure 2.2.2-3: Copy Access Token

2.2.3 Customer management

Every Customer is managed by administrator Tenant. Username is email based and first time user need to verify their email ID by using activation link. **Software has the facility to create Administrator tenant for Customer. By using this account customer has full authority for all the feature** like he can create devices, asset, user and rules.

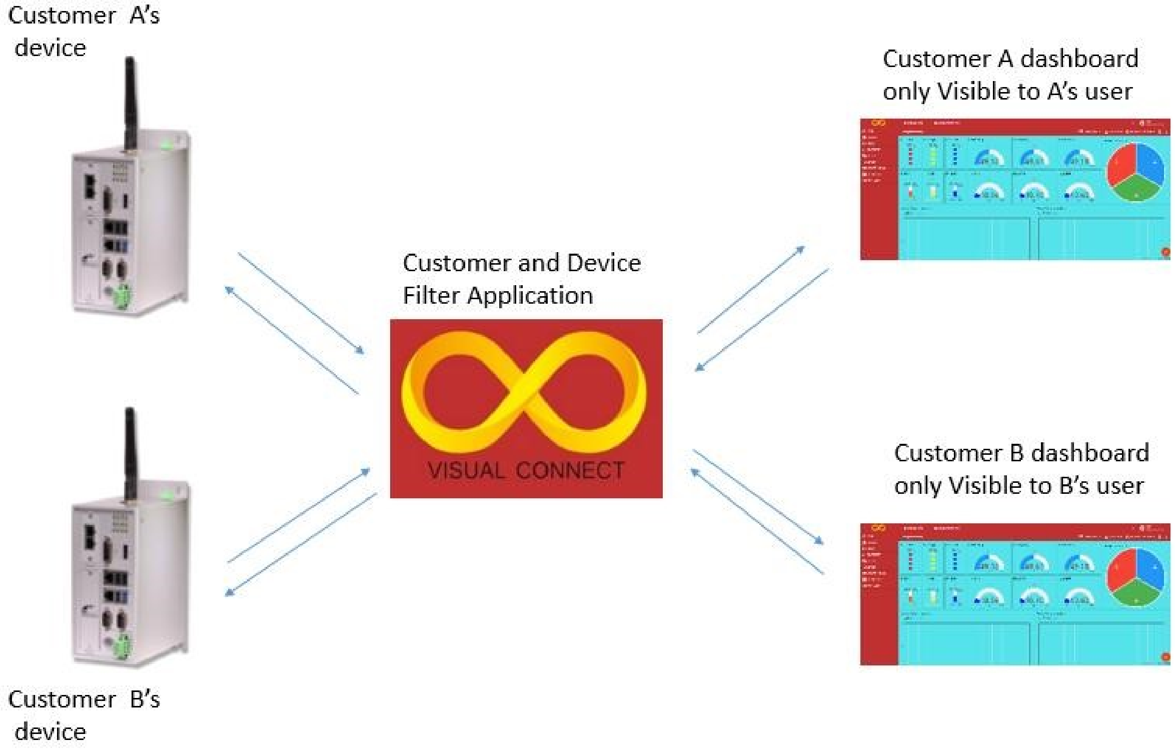

Customer A's device
Customer A dashboard only Visible to A's user
Customer and Device Filter Application
VISUAL CONNECT
Customer B dashboard only Visible to B's user
Customer B's device

Figure 2.2.3-1: Customer Management

Customer Management:

Customer can be created using customer management tool on dashboard. Each customer may have multiple user.

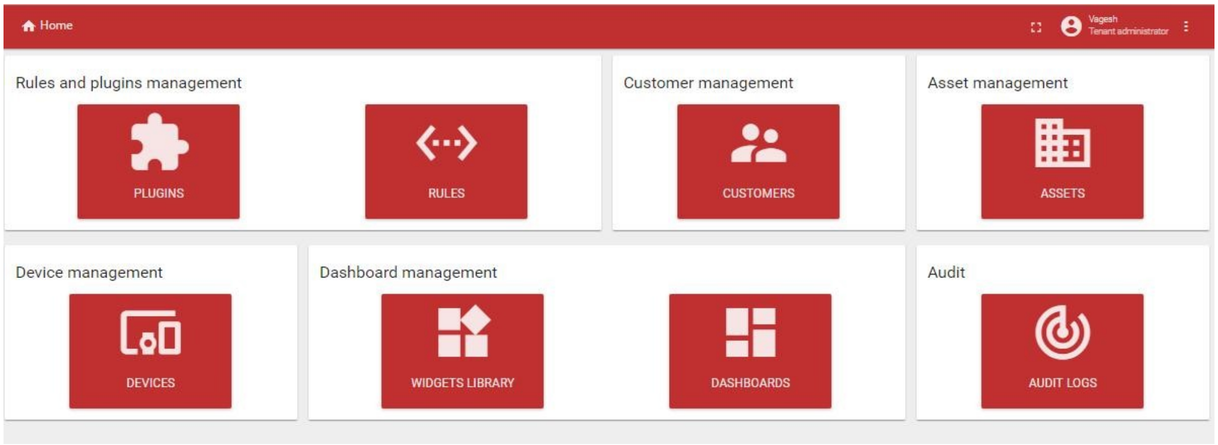
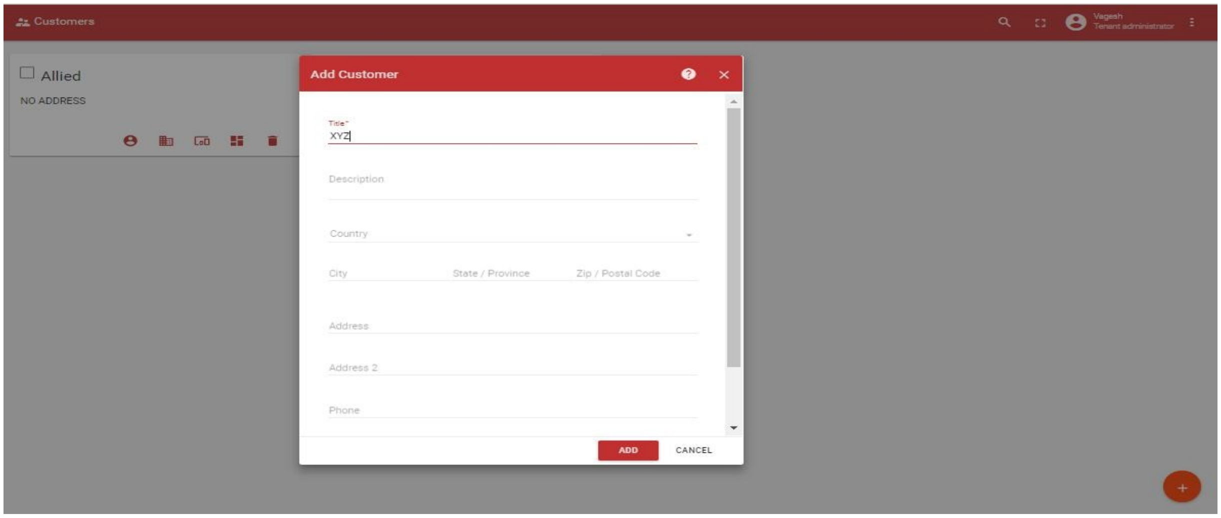
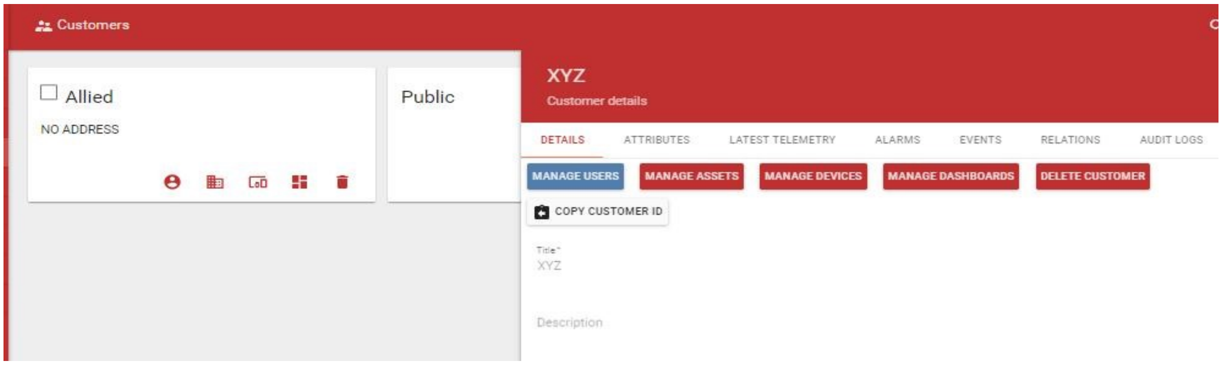


Figure 2.2.3-2: Dashboard for Administrator Tenant



Title*
XYZ

Figure 2.2.3-3: Customer creation



Title*
XYZ

Figure 2.2.3-4: Customer based User Creation

2.2.4 Device management

Device or Gateway can be created using device management tool available on Visual Connect Dashboard. Each device can be assigned to specific customer. Software also shows the information of Device attributes like Firmware Version, and model number.

Below is the Dashboard View for Customer

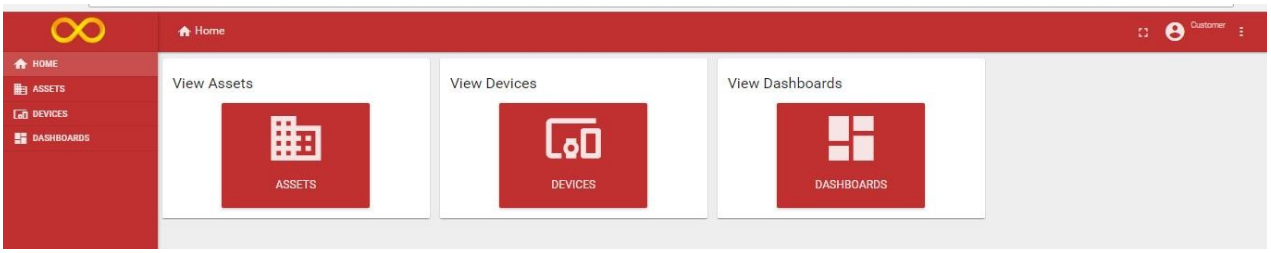
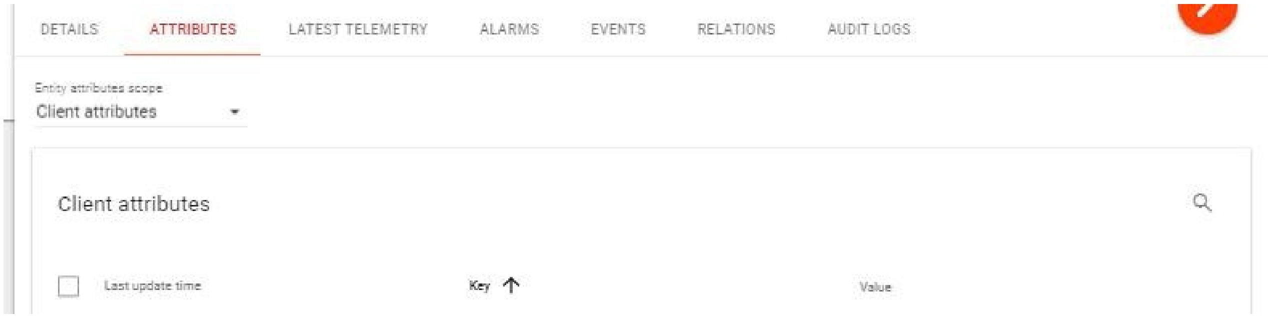


Figure 2.2.4.1: Dashboard (Customer View)

Figure 2.2.4-1: Dashboard View for Customer Login



Entity attributes scope: Client attributes

Figure 2.2.4-2: Device Attributes

2.2.5 Asset/relation management

Different types of asset can be created and each asset can be interrelated by means of relation. Asset can be assigned to different customers.

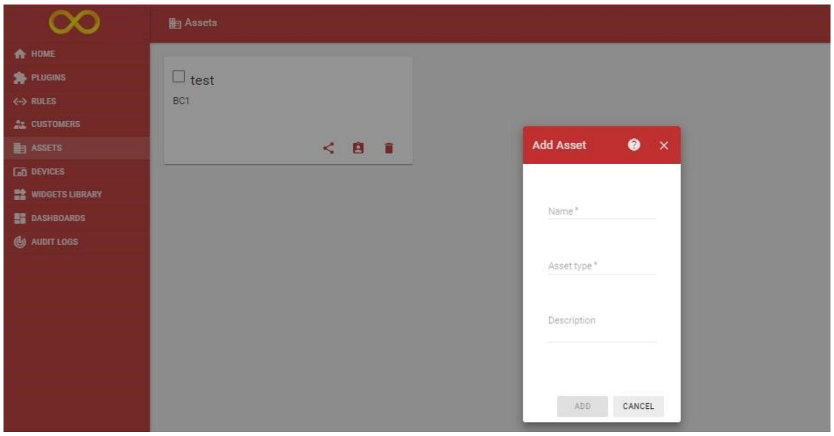
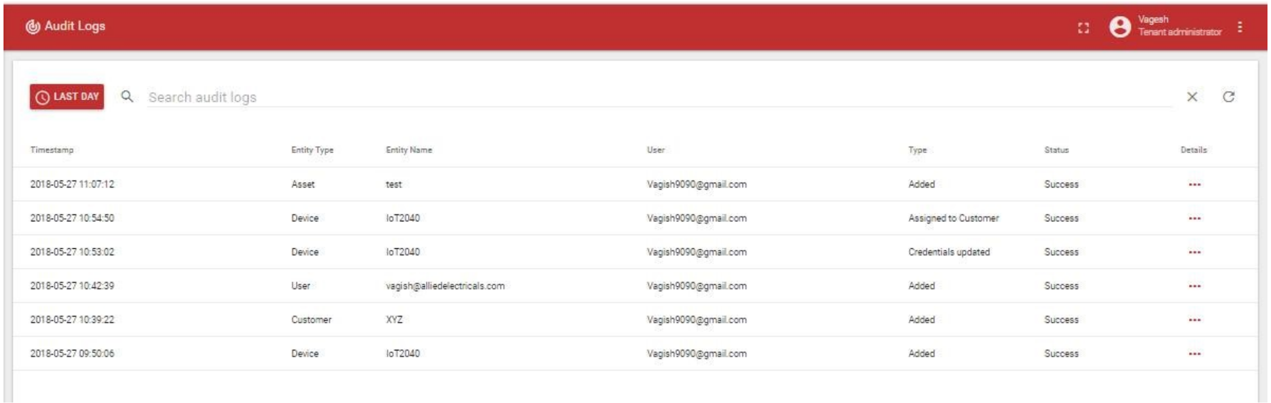


Figure 2.2.5-1: Creating Asset

2.3 Audit log

Each activity/ change in the system is logged in audit log section. Normal alarm can be configured and history can be maintained of all the fault occurrence.



Timestamp	Entity Type	Entity Name	User	Type	Status	Details
2018-05-27 11:07:12	Asset	test	Vagish9090@gmail.com	Added	Success	•••
2018-05-27 10:54:50	Device	IoT2040	Vagish9090@gmail.com	Assigned to Customer	Success	•••
2018-05-27 10:53:02	Device	IoT2040	Vagish9090@gmail.com	Credentials updated	Success	•••
2018-05-27 10:42:39	User	vagish@alliedelectricals.com	Vagish9090@gmail.com	Added	Success	•••
2018-05-27 10:39:22	Customer	XYZ	Vagish9090@gmail.com	Added	Success	•••
2018-05-27 09:50:06	Device	IoT2040	Vagish9090@gmail.com	Added	Success	•••

Figure 2.3-1: Audit Log

3 Rules and plugins

We will review following example to explain the difference between Rule and Plugin. Let's assume you want to send an email to an engineer when the engine temperature is too high. In this case, Rule is responsible for analyzing telemetry data and building email (body, to, cc, etc). However, the plugin is responsible for actual communication with the email server and sending emails. So, multiple rules may use the same email plugin that is configured once.

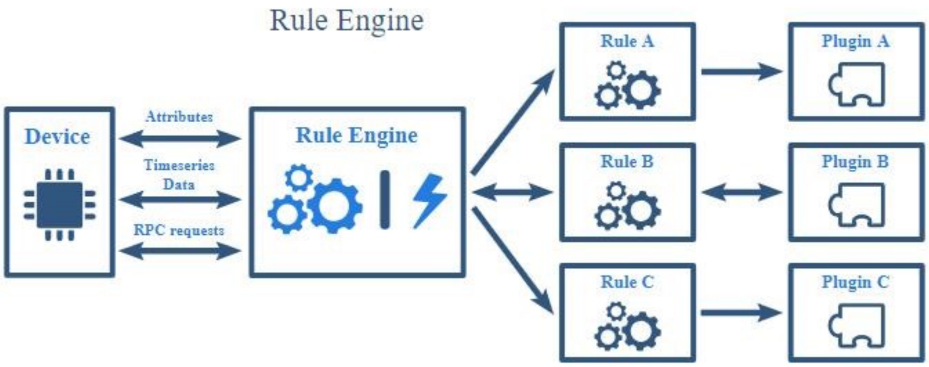


Figure 3-1: Rules and Plugins Relation

3.1 Message Filter

This component allows filtering incoming messages by type. This filter is very efficient. We recommend using this filter in almost every rule to quickly ignore irrelevant messages.

Configuration: You are able to select multiple types: "Get Attributes", "Post Attributes", "Post Telemetry" and "RPC Request".

3.2 Device Attribute Filter

This component allows filtering incoming messages by attributes of the device. This filter is very useful if you want to apply the rule only to the certain sub-set of your device. Filter expression is a javascript expression and basically defines this sub-set. You are able to use any attribute types.

Configuration:You are able to write boolean javascript expression using following bindings:

- cs - client-side attributes map.
- ss - server-side attributes map.
- shared - shared attributes map.
  If you are not sure that certain attribute is present, you can add check it's type for undefined.

3.3 Device Telemetry Filter

This component allows filtering incoming telemetry messages by their values. This filter is very useful if you want to apply rule based on certain values of telemetry. For example, an engine controller may periodically report its temperature. When engine temperature is higher than 100 degrees you may raise an alert. The filter expression is written in javascript.

Configuration: You are able to write boolean javascript expression using bindings that match keys of your telemetry message. If you are not sure that certain key is present in your message, you can add check it’s type for *undefined*.

3.4 Method name Filter

This component allows filtering incoming RPC request messages by method name. This filter is very efficient and useful to forward RPC request to particular plugins that handle them.
You are able to select multiple method names in one filter. For example, if you want to have two plugins (their functionality is just for the demo purposes):

- plugin A allows getting current time
- plugin B allows getting the weather forecast You may implement plugin A to handle getTime method and plugin B to handle getWeather method. In this case you will need to configure two rules:
- rule A that points to plugin A based on “getTime” method filter
- rule B that points to plugin B based on “getWeather” method filter

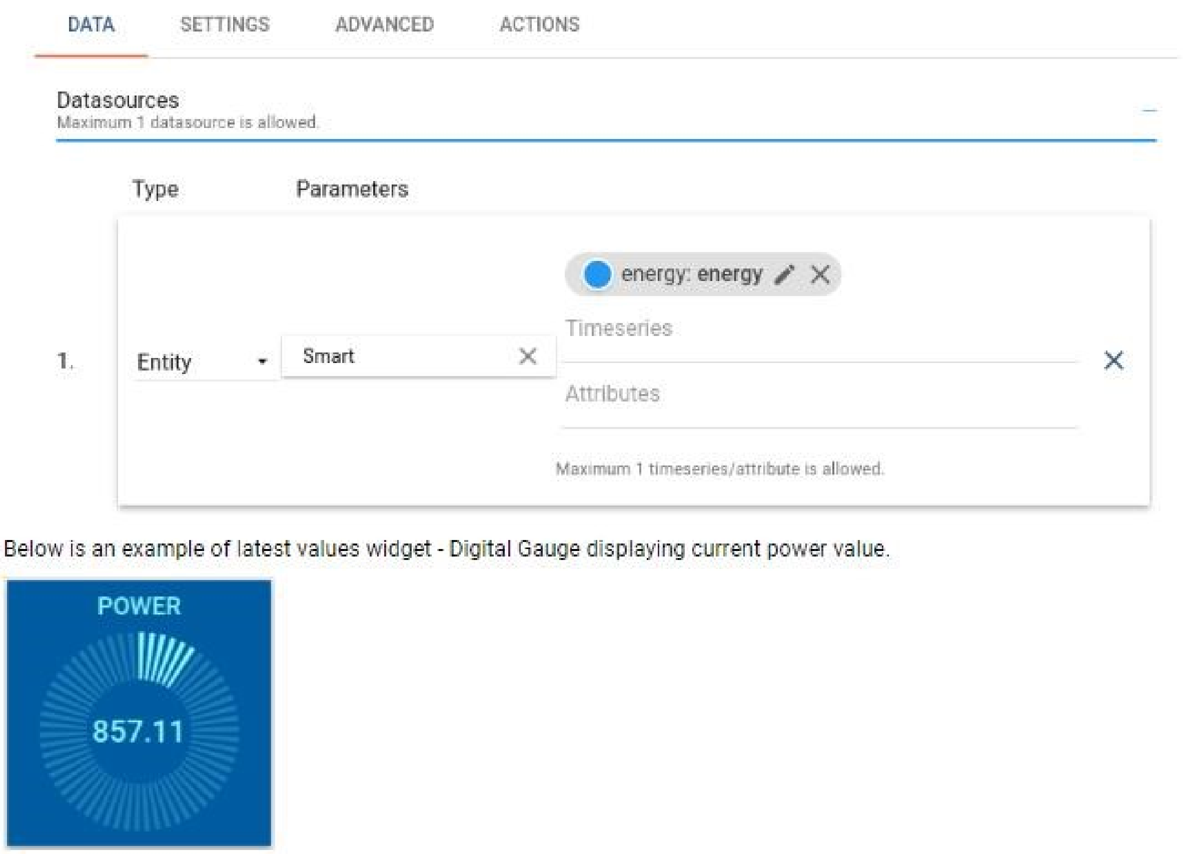
4 Visualization

Visual connect has number of UI interface for data. Visual Connect has following widgets library

- Digital Gauges
- Analog Gauges
- Charts
- GPIO widgets
- Control widgets
- Maps widgets
- Cards
- Alarm widgets

4.1 Latest Value

Displays latest values of particular entity attribute or timeseries data point (for ex. any Gauge Widget or Entities Table widget). This kind of widgets uses values of entity attribute(s) or timeseries as datasource.



Datasources
Maximum 1 datasource is allowed.

Type Parameters

1. Entity Smart

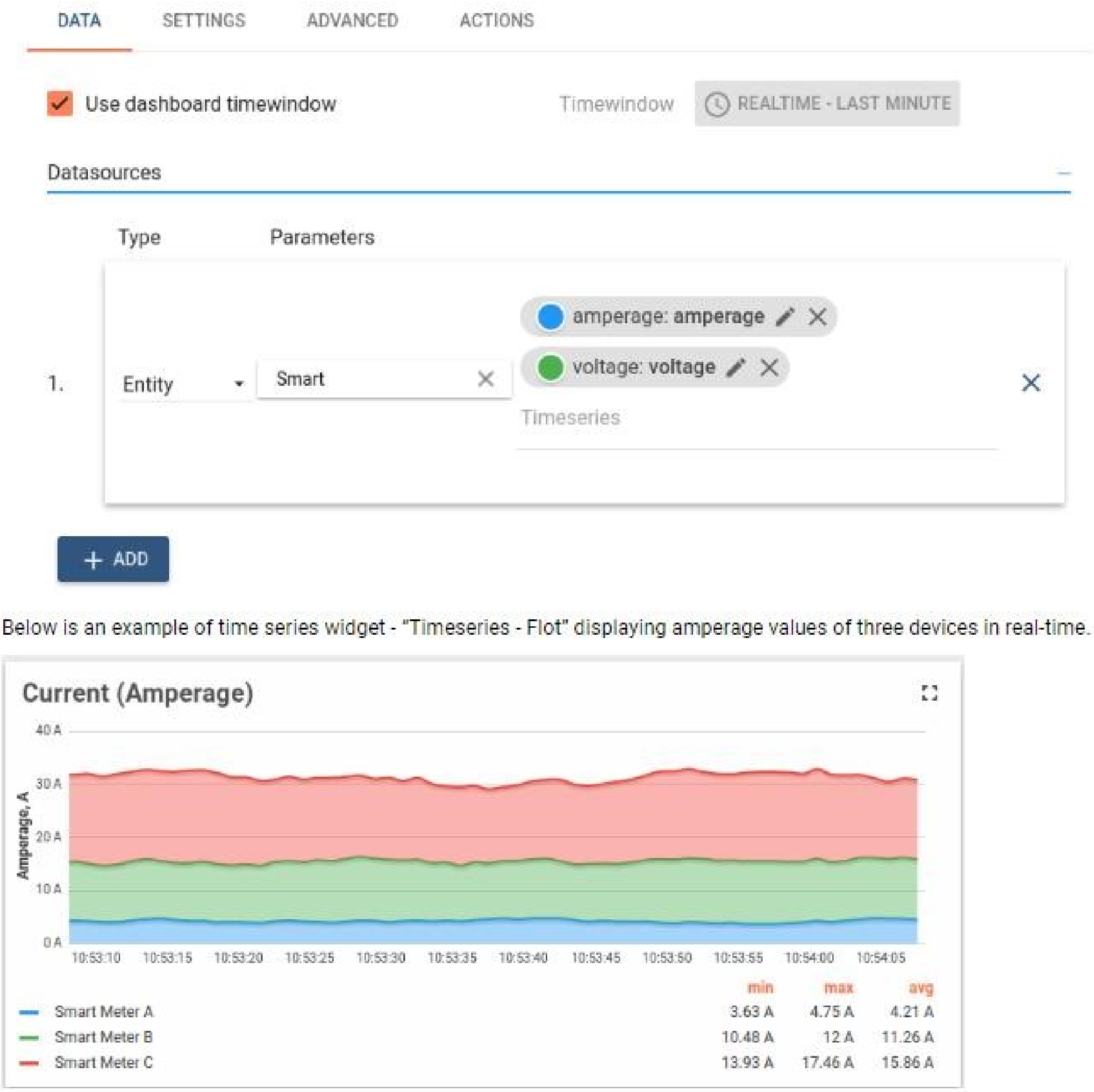
energy: energy
Timeseries
Attributes
Maximum 1 timeseries/attribute is allowed.

Below is an example of latest values widget - Digital Gauge displaying current power value.

Figure 4.1-1: Latest Value

4.2 Time-Series

Dispay Hstorical values for the selected time period or latest values in the certain time window (for ex. “Timeseries - Flot” or “Timeseries table”). This kind of widgets uses only values of entity timeseries as datasource. In order to specify the time frame of displayed values, Timewindow settings are used. Timewindow can be specified on the dashboard level or on the widget level. It can be either realtime - dynamically changed time frame for some latest interval, or history - fixed historical time frame. All these settings are part of Time-series widget configuration



☑ Use dashboard timewindow

Timewindow: REALTIME - LAST MINUTE

Datasources

Type	Parameters	
1. Entity	Smart	amperage: amperage voltage: voltage Timeseries

Below is an example of time series widget - "Timeseries - Flot" displaying amperage values of three devices in real-time.

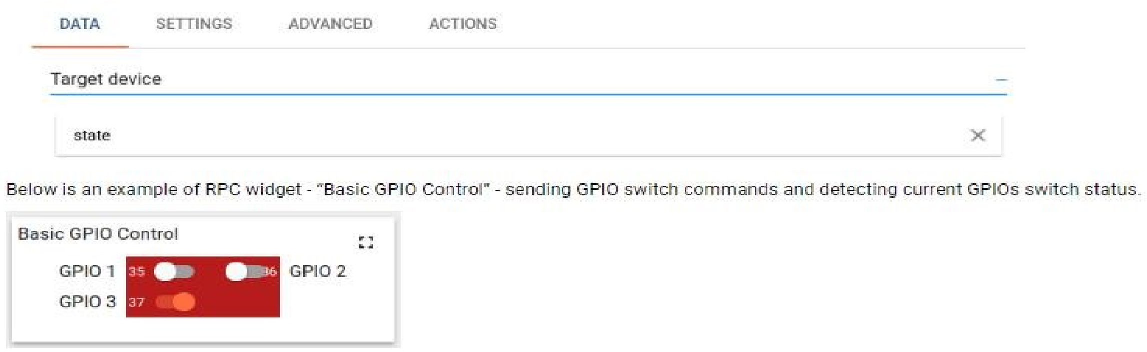
Current (Amperage)

	min	max	avg
Smart Meter A	3.63 A	4.75 A	4.21 A
Smart Meter B	10.48 A	12 A	11.26 A
Smart Meter C	13.93 A	17.46 A	15.86 A

Figure 4.2-1: History View

4.3 RPC (Control Widgets)

Allows to send RPC commands to devices and handles/visualize reply from the device (for ex. “Raspberry Pi GPIO Control”). RPC widgets are configured by specifying target device as target endpoint for RPC commands.



Target device

state

Below is an example of RPC widget - "Basic GPIO Control" - sending GPIO switch commands and detecting current GPIOs switch status.

Basic GPIO Control

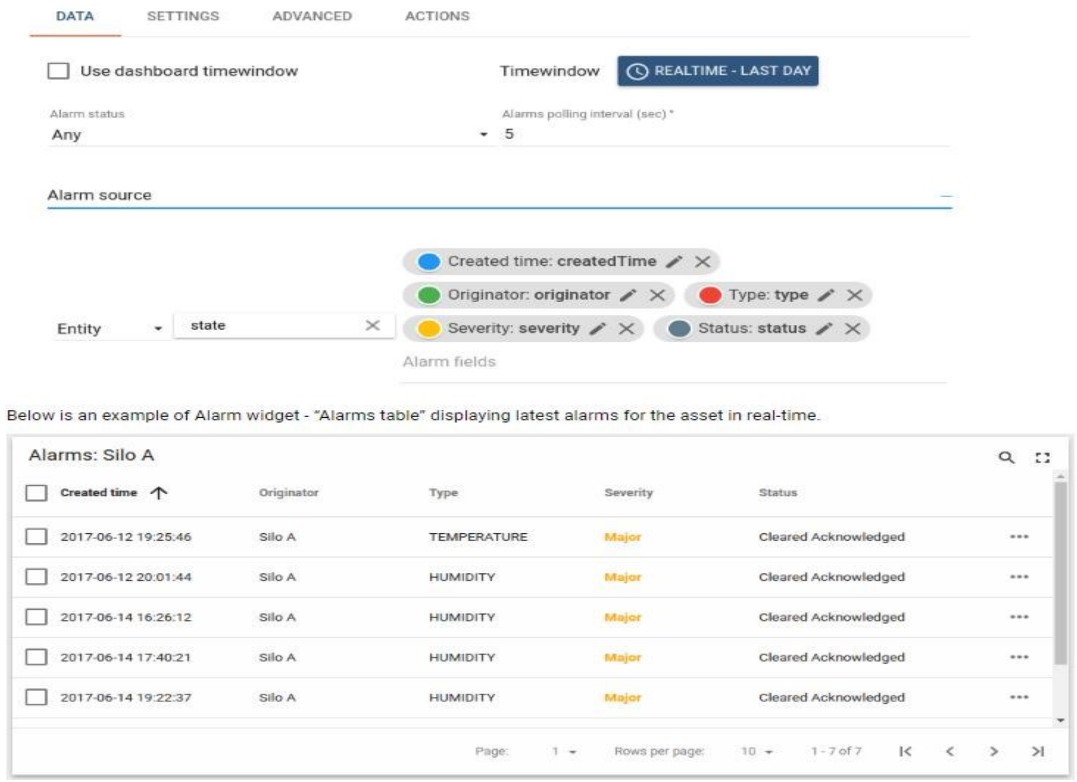
GPIO 1 35 · 36 GPIO 2

GPIO 3 37

Figure 4.3-1: RPC

4.4 Alarm Widgets

Display alarms related to the specified entity in the certain time window (for ex. “Alarms table”). Alarm widgets are configured by specifying entity as alarms source and corresponding alarm fields. Like Time-series widgets alarm widgets have the timewindow configuration in order to specify the time frame of displayed alarms. Additionally configuration contains “Alarm status” and “Alarms polling interval” parameters. “Alarm status” parameter specifies the status of alarms being fetched. “Alarms polling interval” controls alarms fetching frequency in seconds.



□ Use dashboard timewindow

Timewindow: REALTIME - LAST DAY

Alarm status: Any

Alarms polling interval (sec) *: 5

Alarm source

Entity: state

Created time: createdTime

Originator: originator

Type: type

Severity: severity

Status: status

Alarm fields

Below is an example of Alarm widget - "Alarms table" displaying latest alarms for the asset in real-time.

Alarms: Silo A

□	Created time ↑	Originator	Type	Severity	Status	
□	2017-06-12 19:25:46	Silo A	TEMPERATURE	Major	Cleared Acknowledged	•••
□	2017-06-12 20:01:44	Silo A	HUMIDITY	Major	Cleared Acknowledged	•••
□	2017-06-14 16:26:12	Silo A	HUMIDITY	Major	Cleared Acknowledged	•••
□	2017-06-14 17:40:21	Silo A	HUMIDITY	Major	Cleared Acknowledged	•••
□	2017-06-14 19:22:37	Silo A	HUMIDITY	Major	Cleared Acknowledged	•••

Page: 1 Rows per page: 10 1 - 7 of 7

Figure 4.4-1: Alarm

4.5 Digital Gauge

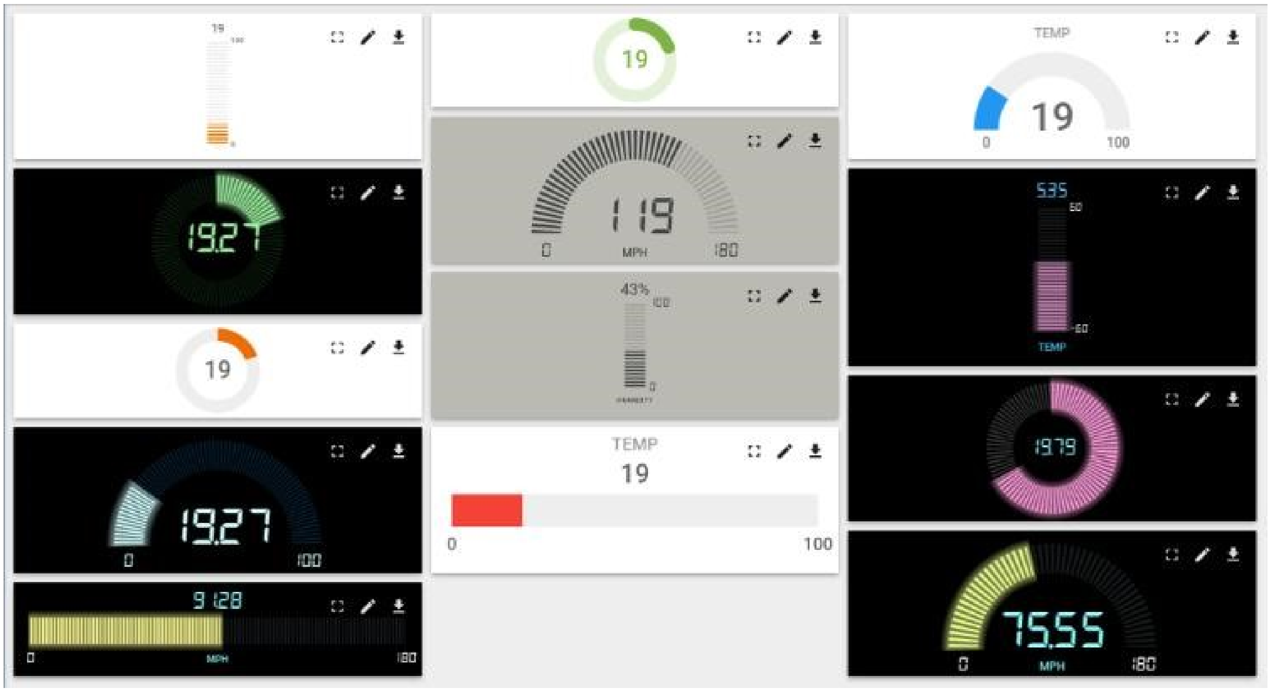


Figure 4.5-1: Digital Gauge

4.6 Analog Gauge

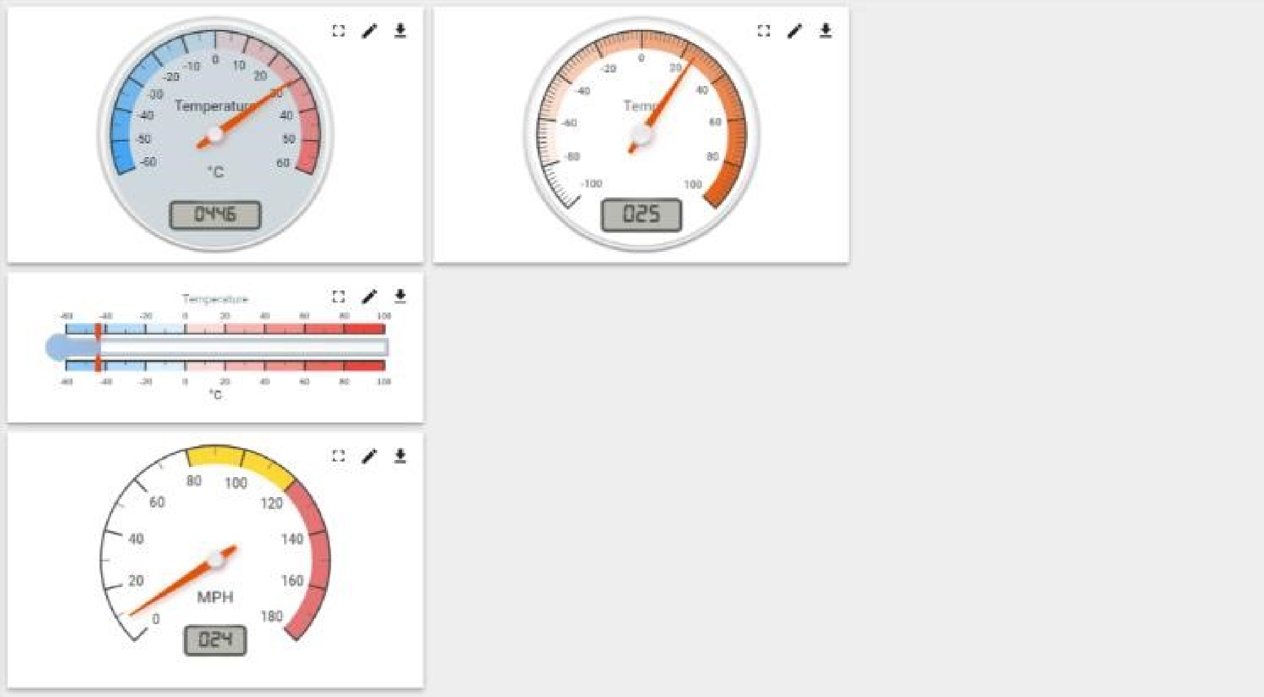


Figure 4.6-1: Analog Gauge

4.7 Charts

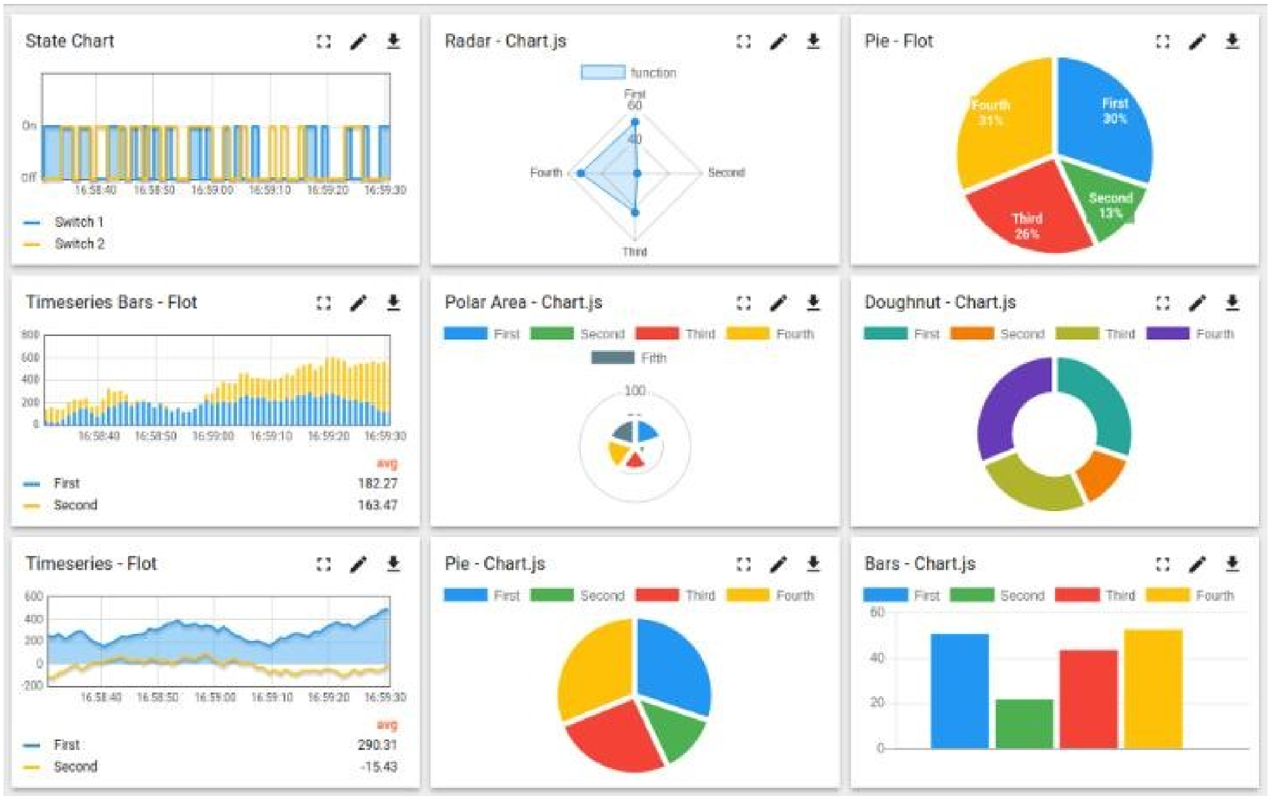
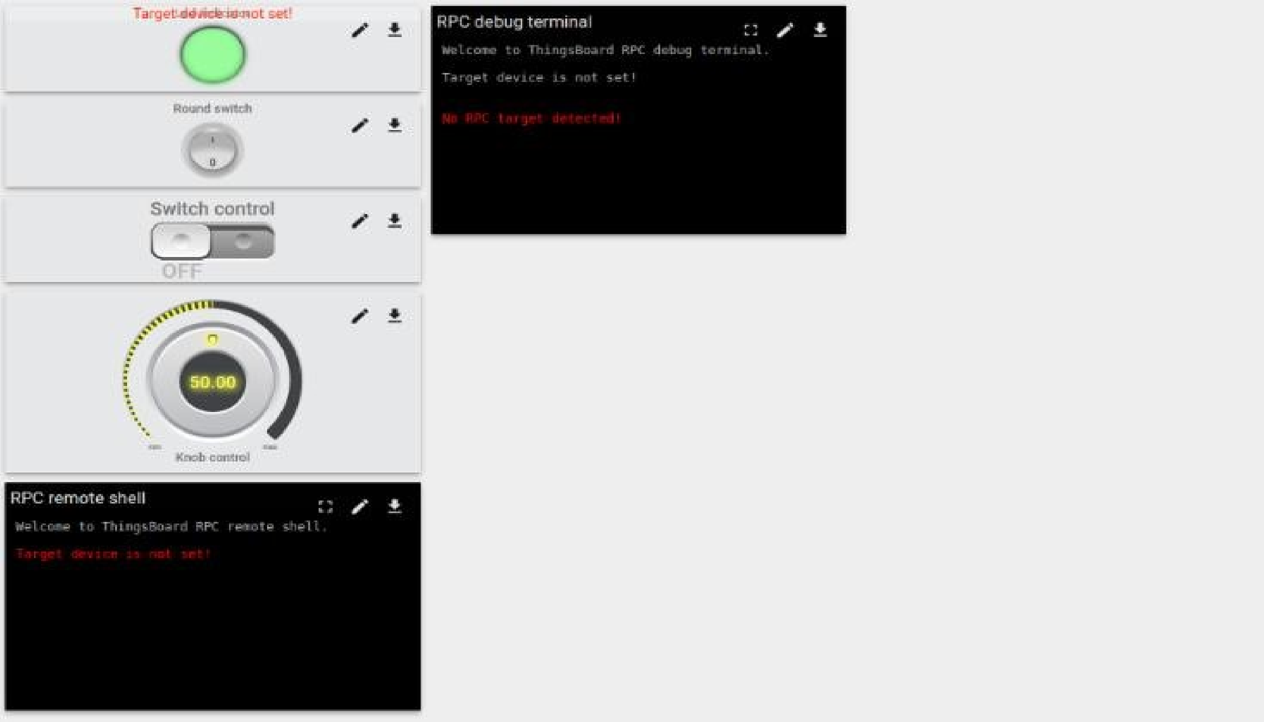


Figure 4.7-1: Charts

4.8 Control widgets

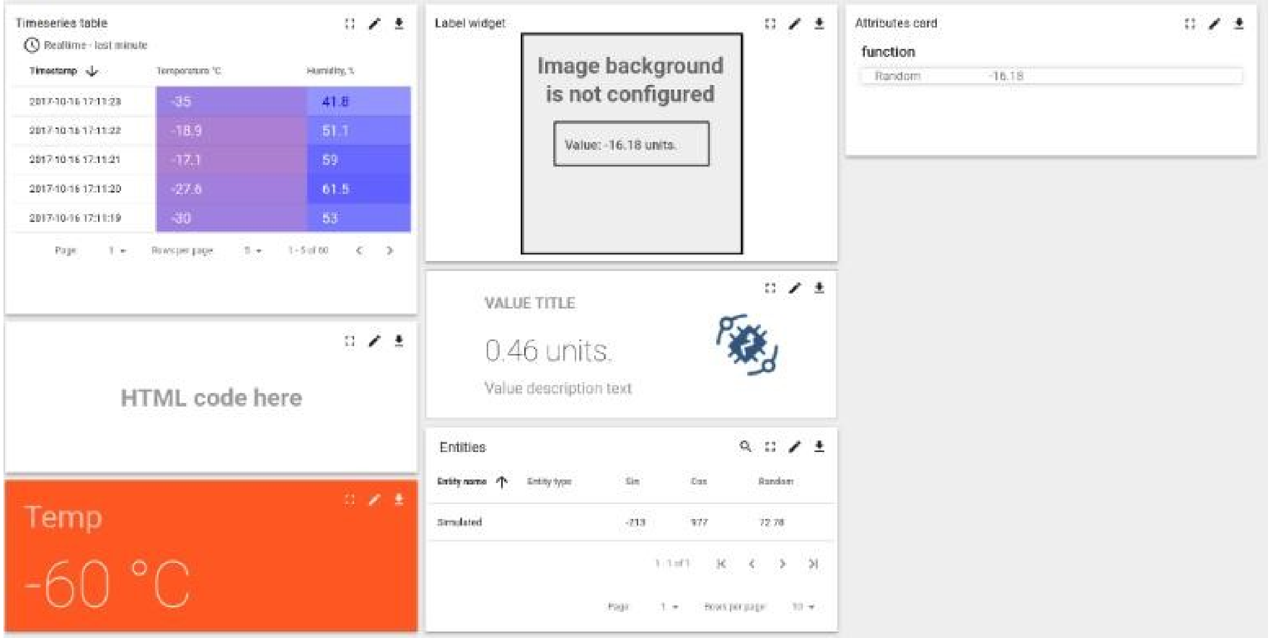


```
Welcome to ThingsBoard RPC debug terminal.
Target device is not set!
No RPC target detected!
```

```
Welcome to ThingsBoard RPC remote shell.
Target device is not set!
```

Figure 4.8-1: Control Widgets

4.9 Cards



Timeseries table

Realtime - last minute

Timestamp ↓	Temperature °C	Humidity, %
2017-10-16 17:11:23	-35	41.8
2017-10-16 17:11:22	-18.9	51.1
2017-10-16 17:11:21	-17.1	59
2017-10-16 17:11:20	-27.6	61.5
2017-10-16 17:11:19	-30	53

Page 1 Rows per page 5 1 - 5 of 60

Label widget

Image background is not configured

Value: -16.18 units.

Attributes card

function

Random -16.18

HTML code here

VALUE TITLE

0.46 units.

Value description text

Temp

-60 °C

Entities

Entity name ↑	Entity type	Sin	Cos	Random
Simulated		-213	977	72.78

1 - 1 of 1

Page 1 Rows per page 10

Figure 4.9-1: Cards

4.10 Maps

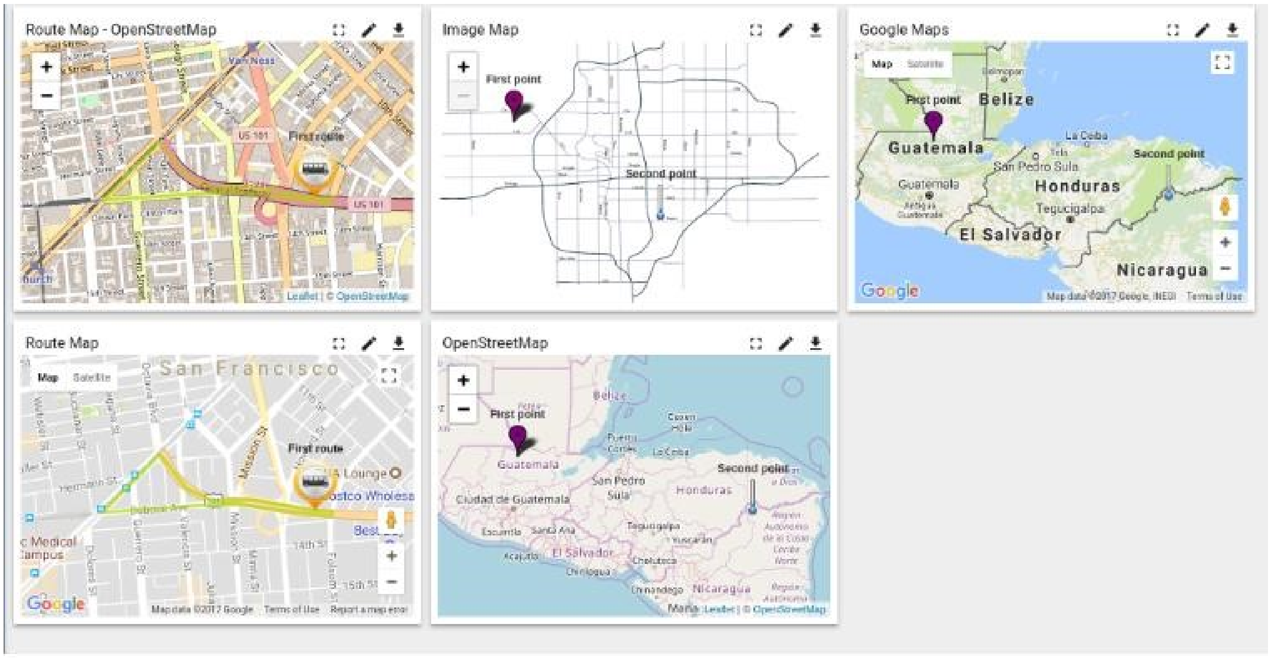


Figure 4.10-1: Cards

5 Security

Mqtt over SSL: Visual Connect provides the ability to run MQTT server over SSL. Both one-way and two-way SSL are supported. To enable SSL, you will need to obtain a valid or generate a self-signed SSL certificate and add it to the keystore. Once added, you will need to specify the keystore information in .yml file.

Device Authentication option: Device credentials are used in order to connect to the Visual Connect server by applications that are running on the device. Visual Connect is designed to support different device credentials. There are two supported credentials types at the moment:

- **Access Tokens** - general purpose credentials that are suitable for wide range of devices. Access Token based authentication may be used in not encrypted or one-way SSL mode.
    - **Advantages:** supported by resource constrained devices. Low network overhead. Easy to provision and use.
- **X.509 Certificates** - PKI and TLS standard. X.509 Certificate based authentication is used in two-way SSL mode.
    - **Advantages:** high level of security using the encrypted network connection and public key infrastructure.

Access Token Based Authentication: Access Token Based Authentication is the default device authentication type. Once the device is created in Visual Connect, the default access token is generated. It can be changed afterwards. In order to connect the device to a server using Access Token based authentication, the client must specify the access token as part of request URL (for HTTP and CoAP) or as a user name in MQTT connect message.

X.509 Certificate Based Authentication: X.509 Certificate Based Authentication is used in Two-Way SSL connection. In this case, the certificate itself is the client's ID, thus, Access Token is no longer needed. Instructions below will describe how to generate a client-side certificate and connect to the server that is running MQTT over SSL. You will need to have the public key of the server certificate in PEM format.

6 Hosting and Support

We are hosting Visual connect on AWS and it's available to customer as SaaS. AWS has most secure architecture. Visual connect is deployed on **Amazon EC2**. Amazon Elastic Compute Cloud (Amazon EC2) is a web service that provides secure, resizable compute capacity in the cloud. It is designed to make web-scale cloud computing easier for developers.

Amazon Relational Database Service (Amazon RDS) makes it easy to set up, operate, and scale a relational database in the cloud. It provides cost-efficient and resizable capacity while automating time-consuming administration tasks such as hardware provisioning, database setup, patching and backups. It frees you to focus on your applications so you can give them the fast performance, high availability, security and compatibility they need.

Elastic Load Balancing automatically distributes incoming application traffic across multiple targets, such as Amazon EC2 instances, containers, and IP addresses. It can handle the varying load of your application traffic in a single Availability Zone or across multiple Availability Zones. Elastic Load Balancing offers three types of load balancers that all feature the high availability, automatic scaling, and robust security necessary to make your applications fault tolerant.

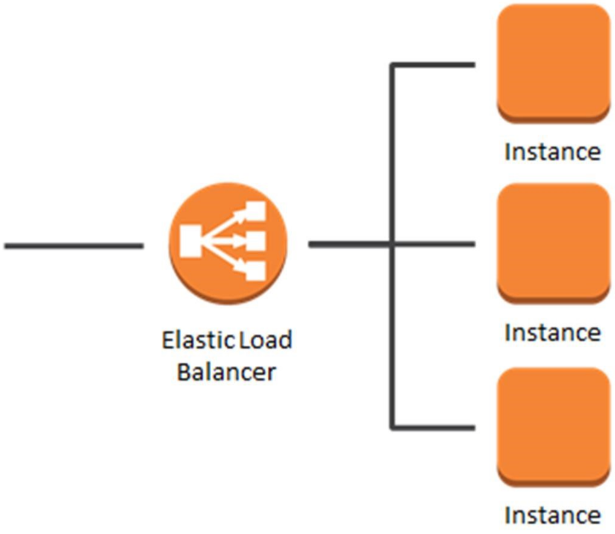


Figure 6-1: Elastic load balancing