
MyEMS

MyEMS Team

Jun 19, 2022

CONTENTS:

- 1 Introduction** **1**
- 1.1 Target audience 1
- 1.2 Reasons to use MyEMS 1

- 2 Architecture** **3**

- 3 Components** **5**

- 4 Screen Shoots** **7**

- 5 Compare Editions** **9**

- 6 Download** **11**

- 7 Installation** **13**
- 7.1 MyEMS API Service 13

- 8 Administrator Guide** **17**

- 9 User Guide** **19**

- 10 Developer Guide** **21**

INTRODUCTION

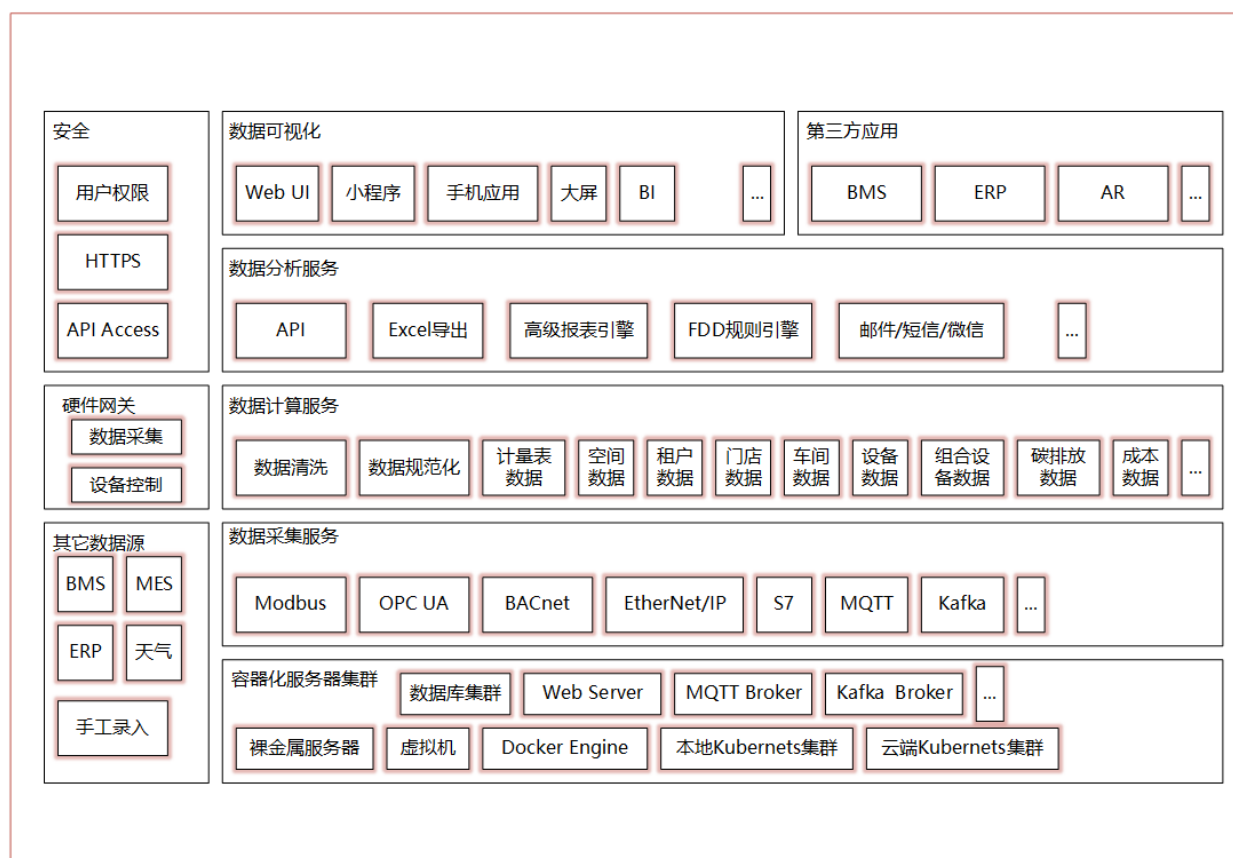
MyEMS is an industry leading open source Energy Management System that is built on cloud computing, IOT, Big Data and AI technologies. MyEMS can be used for a standard and powerful integrated energy management service platform. MyEMS is being developed and maintained by an experienced development team, and the system's source code is published under MIT license.

1.1 Target audience

1.2 Reasons to use MyEMS

ARCHITECTURE

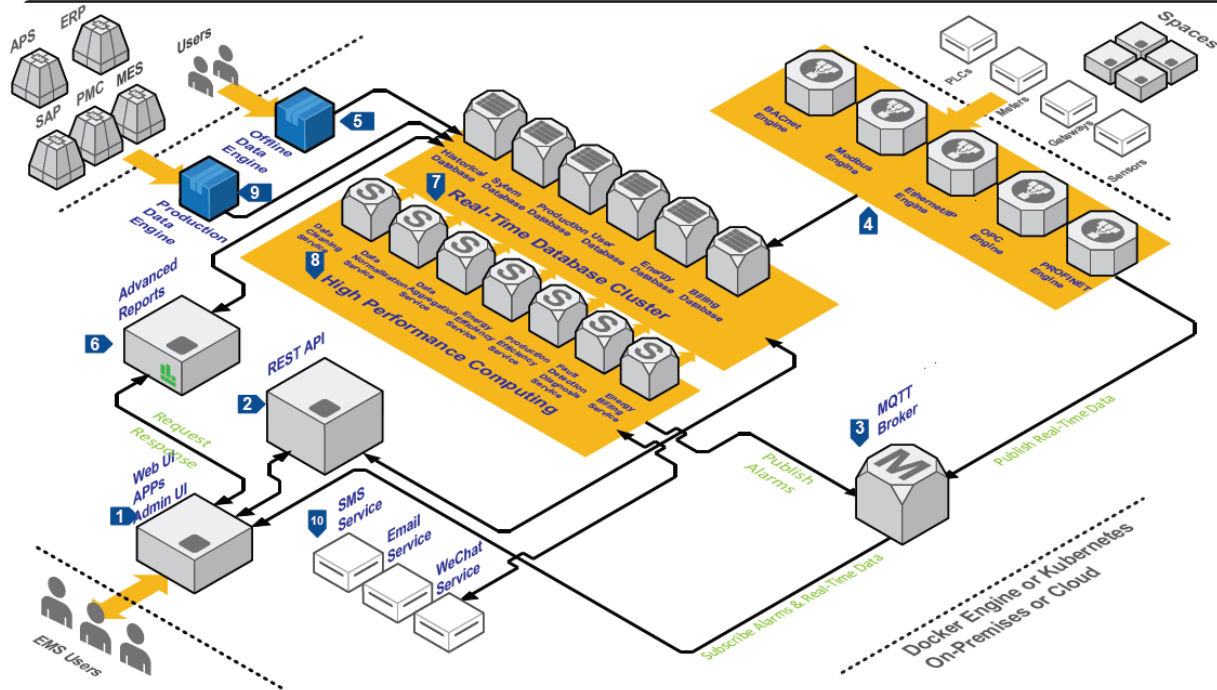
images/architecture-site-view.png



COMPONENTS

MyEMS Software Architecture

Copyright © 2022
MyEMS
All Rights Reserved



Components
Overview

- | | | | |
|---|---|--|---|
| 1 Responsive Web UI, Admin UI, iOS APP, Android APP, Mini Programs | 4 Acquisition of energy data from PLCs, Meters and Sensors | 7 Persistent storage for historical data, system, production, users, energy consumption and energy billing | 10 Sending energy efficiency alarms by Email, SMS or WeChat |
| 2 Providing RESTful API to Web UI or Mobile APPs | 5 Import offline energy or production data from Excel | 8 Data analysis services based on high performance computing platform | |
| 3 Using Broker to implement the MQTT protocol for distributing real-time messages | 6 Reporting server to provide reporting and analytics that can be embedded into the web application | 9 Acquisition of production data | |

Guangyu Zhang
01/14/2022

SCREEN SHOTS



images/screenshot-2020.12.01-16_53_12.png

COMPARE EDITIONS

DOWNLOAD

INSTALLATION

7.1 MyEMS API Service

7.1.1 Introduction

Providing REST API service for MyEMS Web APP, Android APP and iOS APP and/or third parties

7.1.2 Prerequisites

anytree
simplejson
mysql.connector
falcon
falcon_cors
gunicorn
openpyxl

7.1.3 Installation

- Install anytree:

```
$ cd ~/tools
$ git clone https://github.com/c0fec0de/anytree.git
$ cd anytree
$ sudo python3 setup.py install
```

- Install simplejson:

```
$ cd ~/tools
$ git clone https://github.com/simplejson/simplejson.git
$ cd simplejson
$ sudo python3 setup.py install
```

- Install MySQL Connector:

```
$ cd ~/tools
$ wget https://cdn.mysql.com//Downloads/Connector-Python/mysql-connector-
python-8.0.28.tar.gz
```

(continues on next page)

(continued from previous page)

```
$ tar xzf mysql-connector-python-8.0.28.tar.gz
$ cd ~/tools/mysql-connector-python-8.0.28
$ sudo python3 setup.py install
```

- Install Falcon

if you are behind proxy, use `-proxy` parameter

Refer to

<https://falconframework.org/>

<https://github.com/lwcolton/falcon-cors>

<https://github.com/yohanboniface/falcon-multipart>

```
$ mkdir ~/tools/falcon && cd ~/tools/falcon
$ pip3 download cython falcon falcon-cors falcon-multipart
$ export LC_ALL="en_US.UTF-8"
$ export LC_CTYPE="en_US.UTF-8"
$ sudo dpkg-reconfigure locales
$ sudo pip3 install --upgrade --no-index --find-links ~/tools/falcon cython falcon-
→falcon-cors falcon-multipart
```

- Install gunicorn, refer to <http://gunicorn.org>:

```
$ mkdir ~/tools/gunicorn && cd ~/tools/gunicorn
$ pip3 download gunicorn
$ sudo pip3 install --no-index --find-links ~/tools/gunicorn gunicorn
```

- Install openpyxl, refer to <https://foss.heptapod.net/openpyxl/openpyxl>

Get the latest version of `et_xmlfile` from https://foss.heptapod.net/openpyxl/et_xmlfile/

Get the latest version of `jdcal` from <https://github.com/phn/jdcal>

Get the latest version of `openpyxl` from <https://foss.heptapod.net/openpyxl/openpyxl>

```
$ cd ~/tools
$ wget https://foss.heptapod.net/openpyxl/et_xmlfile/-/archive/1.1/et_xmlfile-1.1.
→tar.gz
$ tar xzf et_xmlfile-1.1.tar.gz
$ cd ~/tools/et_xmlfile-1.1
$ sudo python3 setup.py install
$ cd ~/tools
$ git clone https://github.com/phn/jdcal.git
$ cd ~/tools/jdcal
$ sudo python3 setup.py install
$ mkdir ~/tools/pillow && cd ~/tools/pillow
$ pip3 download Pillow
$ sudo pip3 install --no-index --find-links ~/tools/pillow Pillow
$ cd ~/tools
$ wget https://foss.heptapod.net/openpyxl/openpyxl/-/archive/3.0.7/openpyxl-3.0.7.
→tar.gz
```

(continues on next page)

(continued from previous page)

```
$ tar xzf openpyxl-3.0.7.tar.gz
$ cd openpyxl-3.0.7
$ sudo python3 setup.py install
```

- Install gunicorn service for myems-api:

```
$ cd ~/myems-api
$ sudo cp -R ~/myems-api /myems-api
```

Check and change the config file if necessary:

```
$ sudo nano /myems-api/config.py
```

Change the listening port (8000 as an example) in gunicorn.socket:

```
$ sudo nano /myems-api/gunicorn.socket
ListenStream=0.0.0.0:8000
$ sudo ufw allow 8000
```

Setup systemd configure files:

```
$ sudo cp /myems-api/gunicorn.service /lib/systemd/system/
$ sudo cp /myems-api/gunicorn.socket /lib/systemd/system/
$ sudo cp /myems-api/gunicorn.conf /usr/lib/tmpfiles.d/
```

Next enable the services so they autostart at boot:

```
$ sudo systemctl enable gunicorn.socket
$ sudo systemctl enable gunicorn.service
```

Start the services:

```
$ sudo systemctl start gunicorn.socket
$ sudo systemctl start gunicorn.service
```

Run for debugging and testing:

```
$ cd myems-api
$ sudo gunicorn -b 127.0.0.1:8000 app:api
```


ADMINISTRATOR GUIDE

CHAPTER

NINE

USER GUIDE

DEVELOPER GUIDE