# **Requirements Specifications for Energy Prediction/Forecasting**

# Introduction

## Purpose:

The purpose of this document is to provide a detailed specification for a system that predicts or forecasts energy consumption in large connected facilities with multi levels and groups of energy meters. The system will use timeseries data and profound models to analyse data from energy meters and generate accurate predictions or forecasts.

## Scope:

The system will be a software application that can be integrated with existing building management systems. The system will be able to handle large facilities with thousands of energy meters.

## Definitions, Acronyms, and Abbreviations

Facility: A large connected facility with multi levels and groups of energy meters.

Energy Meter: A device that measures energy consumption in a facility.

Prediction: An estimate of future energy consumption based on historical data.

Forecast: A prediction of future energy consumption based on trends and patterns.

Profound Model: A machine learning model that can analyze timeseries data and generate predictions or forecasts.

# Product Perspective:

The system will be a software application that can be integrated with existing building management systems.

## Product Functions:

* Collect and store energy meter data.
* Analyze energy meter data using profound models.
* Generate energy consumption predictions or forecasts.
* Display predictions or forecasts in a user-friendly interface.

## User Characteristics:

Facility managers, building owners, and energy analysts.

## Product Features

* Energy Meter Data Collection: The system will collect energy meter data from various sources, including building management systems and energy meters.
* Data Storage: The system will store energy meter data in a secure and scalable database.
* Data Analysis: The system will analyze energy meter data using profound models to generate accurate predictions or forecasts.
* Prediction Display: The system will display energy consumption predictions or forecasts in a user-friendly interface.

## Use Cases

* Facility Manager: A facility manager can use the system to monitor energy consumption in real-time and make informed decisions about energy usage and management.
* Building Owner: A building owner can use the system to track energy consumption and identify trends or anomalies that may indicate potential issues or opportunities for energy savings.
* Energy Analyst: An energy analyst can use the system to analyze energy consumption data and generate predictions or forecasts to help facility managers and building owners make informed decisions.

# System Architecture

## Data Collection

The first step in the architecture is to collect data from various sources such as energy meters, sensors, and building management systems. The data can be collected using various communication protocols such as Modbus, BACnet, or MQTT. The data is stored in a time-series database TimescaleDB (PostgreSQL extension).

## Data Pre-processing

The collected data may contain missing values, outliers, and noise. Therefore, it is essential to pre-process the data before feeding it into the forecasting model. The pre-processing steps can include:

* Data cleaning: removing missing values, outliers, and noise.
* Data normalization: scaling the data to a common range.
* Data transformation: converting the data into a suitable format for the forecasting model.

## Feature Engineering

Feature engineering is the process of selecting and transforming the relevant features from the data to improve the accuracy of the forecasting model. The features can include:

* Time-based features: hour of the day, day of the week, month, etc.
* Weather-based features: temperature, humidity, wind speed, etc.
* Historical consumption: past energy consumption data.
* Event-based features: Weekends, Sundays, Weekdays.

## Model Selection

There are various forecasting models available for time series data analysis. The selection of the model depends on the nature of the data and the accuracy requirements. Some of the commonly used models are:

* Prophet: a procedure for forecasting time series data based on an additive model where non-linear trends are fit with yearly, weekly, and daily seasonality, plus holiday effects.
* Multi-Prophet: an extension of Prophet that allows for multiple seasonalities and trends.
* FFT: Fast Fourier Transform is a method for transforming a signal from the time domain to the frequency domain and vice versa.
* Linear Regression: a statistical method for modelling the relationship between a dependent variable and one or more independent variables.

## Model Training

Once the model is selected, it needs to be trained using the pre-processed data. The training process involves optimizing the model parameters to minimize the forecast error. The training can be done using various optimization algorithms such as gradient descent, genetic algorithms, or particle swarm optimization.

## Model Evaluation

After training the model, it is essential to evaluate its performance using various metrics such as mean absolute error (MAE), root mean squared error (RMSE), or mean absolute percentage error (MAPE). The evaluation can be done using various techniques such as cross-validation, time-series split, or walk-forward validation.

## Model Deployment

Once the model is trained and evaluated, it can be deployed in the production environment. The deployment can be done using various techniques such as containerization, virtualization, or cloud computing. The deployed model can be used to generate real-time forecasts for energy consumption.

## Integration

The architecture should be integrated with other systems and platforms to provide a seamless user experience. The integration can include various activities such as API development, data integration, and workflow automation.

## Testing

The architecture should be tested using various techniques such as unit testing, integration testing, or acceptance testing to ensure its functionality, performance, and usability.