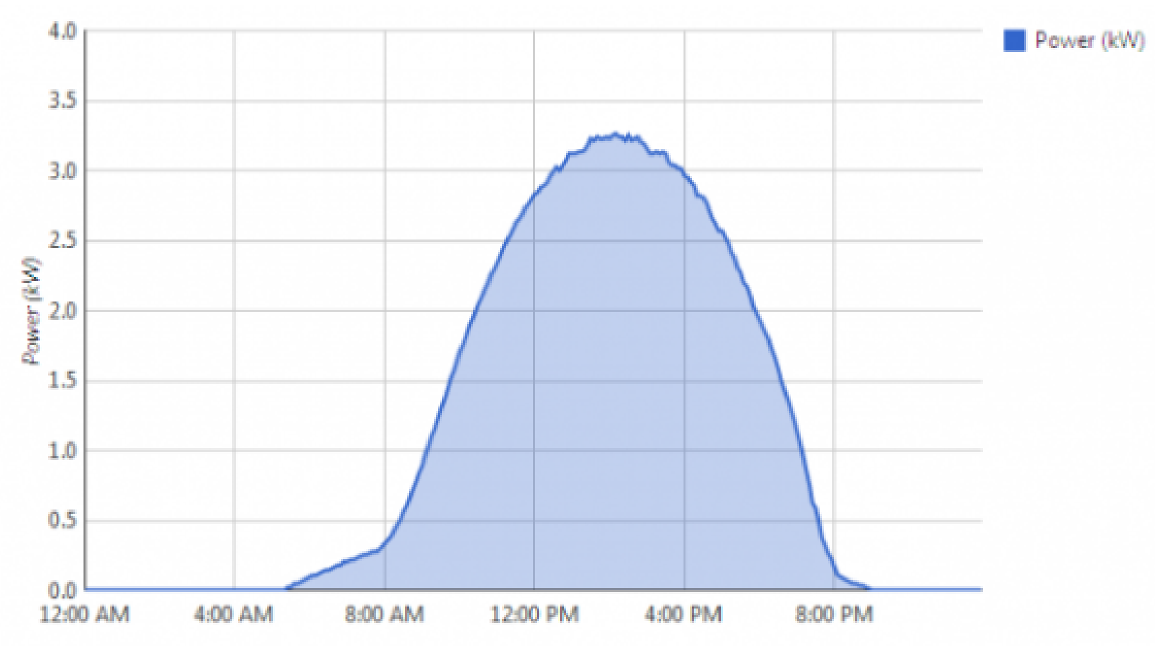
PV Simulator Challenge

In this little challenge, we will request you to build an application which allow us to study the interaction between PV power values generated with a Solar Panel and our building consumption (this last one is measured with a Meter connected to the grid).

PV generator curve

The following picture shows the curve of the power output (in kW) of our PV generator during a normal day. This is the power profile that need to simulate:



Architecture

The next diagram describes the interactions of the PV simulator service and other blocks.

METER →(Broker) → PV SIMULATOR →[OUTPUT]

A brief description of the components:

- **Meter**: This entity produces message to the broker between 0 and 9000 Watts -not kW!- and the cadence of messages produced is one per second. The rest of the implementation is up to you, as long as these two conditions are met. The idea is to simulate readings of the electrical consumption of our building.
- **PV generator**: It must listen to the broker for the meter values, generate a simulated PV power value based on our curve and subtract this value to the meter value, outputting the result.
- **Writing to a file**: The results of the simulation need to be saved in a .csv file, with the timestamp for the generated value, meter, power value (in kW), PV power value and the subtraction of the powers (meter - PV).

Objectives

A single run of the simulation should give us the result of the measures taken during a whole day, composed by multiple samples taken every couple of seconds.
Because we want to make multiple runs, the simulation *should not be in real time*. In other words, we want the results of an emulated complete day to be available in a couple of minutes.

A few more requirements

- Write the solution in Python. If you want to add a component on Rust, you are welcome too, but that's completely optional.
- Fell free to use any library and/or framework, except for the broker, which we would like to be RabbitMQ.
- A README with all the steps to run it.
- We will test your solution on a Debian based OS.
- Good practices for software development are always appreciated. Tests, even more.
- Delivery Date: Around 10 working days.
- If anything is not defined or clear in this document, then feel free to be creative and define it yourself! Just remember to document it for us.