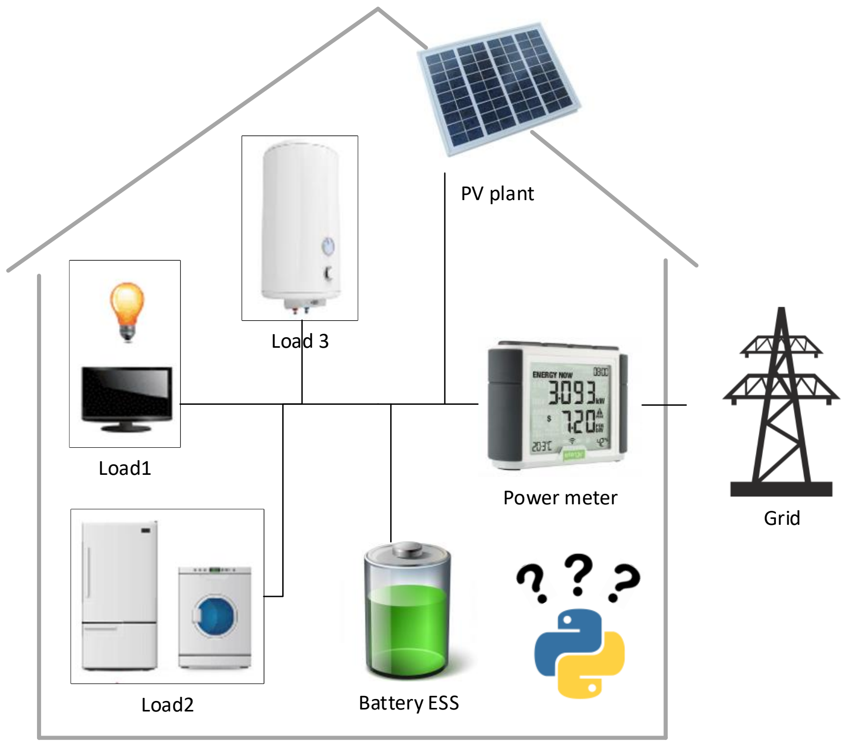
Python Hackathon Task



Consider the illustration above, depicting a set of elements, constituting a smart energy management house. The task for the participants is to design a control system that will drive all the assets to be described, in such a way that the cost for the smart energy management house is minimized. Generally, it gets electrical energy from the distribution grid, but there are alternatives to it, and using those alternatives – the house can even sell the electrical energy to the distribution company.

The power meter has the information about the grid status (on/off, since up-to-one-hour power losses are possible), the buying price of the electrical energy, as well as the selling price, at which the house sells the energy "to" the grid. Both the prices and the grid status are time-dependent and provided through the Typhoon framework model.

There are 3 loads in the house. The first one, the priority load, consists of Internet-dependent devices such as the computer or smart TV, and its penalty in case of a power loss is the highest. The secondary load is comprised of home appliances, such as the refrigerator and the stove, and the penalty when those devices are not supplied is significantly lower. The last one, the boiler, has the lowest priority and thus the lower power-loss penalty. The maximum total consumption for all the loads is 8kW. The total power consumption is defined by predefined load profiles and provided through the Typhoon framework model. All the load profiles are based on the referent one, with a deviation in the range of ±20%. Individual loads have a fixed share of the total load as defined in the table.

Load	Power share	One time power loss penalty	Down time penalty 1/min
Load1	20%	20	1
Load2	40%	4	0.4
Load3	40%	-	0.3

The solar panel has the ability to produce electrical energy, which can then be used to supply the house, or can be transferred back into the grid, and compensated for the corresponding price. Its nominal power is 5 kW. Solar power referent profile is provided through the Typhoon framework model, with a deviation in the range of ±50%.

The battery energy storage system (BESS) has a capacity of 20 kWh, and the maximum power it can provide is 5 kW. SOC information is always available (in the range of [0,1]). It has two operating modes: GC (grid connected) and UPS-mode (Uninterruptible Power Supply).

When connected to the main grid, BESS is operating in the GC mode where its power input/output is defined by the user control system via the power reference. The positive power sign means that the battery is being discharged and vice versa. When the grid is lost, BESS automatically switches to the UPS mode.

In the UPS mode, the battery is forming the local grid and providing the power needed to supply all the local loads. In this mode, the power reference is ignored. BESS will tolerate overload (power exceeding the nominal BESS power) and overcharge (charging the already full battery) conditions for 1 minute before it shuts down, which will cause a total power loss. Therefore, a quick reaction is required to prevent BESS from shutting down. After being shut down, BESS will automatically power on after 1 minute. When the main grid is restored, BESS automatically switches back to the GC mode.

Power balance must be preserved at all times; otherwise, all the loads and the sources shut down. Power balance equations are provided below:

Main grid on: $P_g + P_{bess} + P_{pv} = P_{l1} + P_{l2} + P_{l3}$, P_g calculated, all the rest defined

Main grid off: $P_{bess} + P_{pv} = P_{l1} + P_{l2} + P_{l3}$, P_{bess} calculated, all the rest defined

The system should provide the information regarding the activity of every asset, with time being checked on a minute-base. Calculation of the end cost for every test set is done by the Typhoon framework model; this option will be available to the participants during the whole Hackathon.