FIELD

Senior Backend Developer Technical Test

Overview

We are looking for readable code for independent services that can run locally. We are not looking for 100% test coverage or for outstanding trading strategy. We would also like to get an understanding of how you would go about deploying these services.

Part 1: Local simulation environment for our battery asset

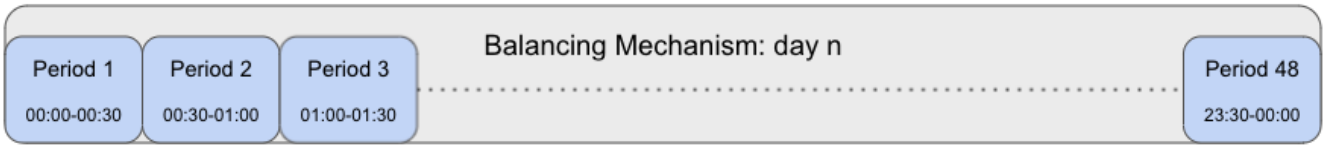
You are asked to code up a collection of services that simulate how we would operate one of our assets in the specific time period.

Simulation period:

2021-10-04 to 2021-10-10 (1 complete week)

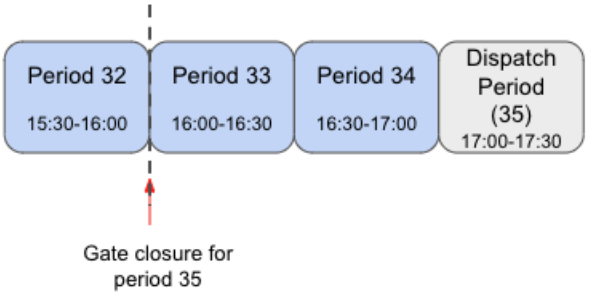
Trading market:

The market under question is the Balancing Mechanism (BM). The BM is divided into half hour settlement periods.



Times in local timezone, BST

In order to participate in a specific settlement period we need to submit bids/offers **1 hour ahead of that period**. An **offer** is defined as an increase in energy export to the grid from our stated baseline (you can assume the baseline to be zero), and a **bid** is an increase in energy import.
You can assume that you immediately find out (next section) if the submitted offers/bids are accepted.
So to participate in the settlement period of 17:00-17:30, it would look like this:

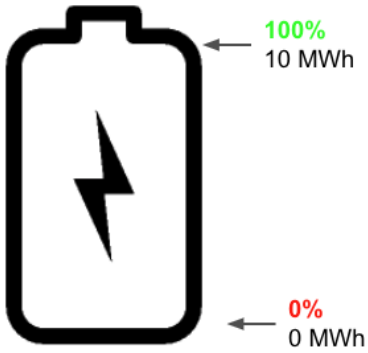


Example of gate closure (bid/offer submission deadline): shown for dispatch period 35

National Grid operators:

These are the people taking actions to ensure supply matches demand whilst meeting various network constraints. They have discretion to accept bids and offers. You can assume that our bids and offers have an **80% acceptance probability**.

Battery asset specifications:



1. **Total capacity of 10 MWh** (you can think of it as a basket that can hold 10kg of energy apples (?!)).
2. Can deliver/charge at a rate of **10 MWh/hour,** i.e. it can import/export a maximum of 5 MWh in one settlement period (30 minutes, as described above).
3. Can do a **maximum of 2 complete cycles on any given day**, i.e. **import a total of 20MWh and export a total of 20MWh**

Battery starting condition:

50% capacity (total capacity is 10MWh, so we start at 5MWh).

To mock machine learning models that predict bid/offer prices we have provided the predictions in json files (bid_price_predictions.json, offer_price_predictions.json). Each model predicts prices for the next 48 periods. The top level key is the timestamp at which the predictions would be made available . Each prediction object is a key-value pair of settlement period start time and predicted price.

Task:

You are asked to provide a solution that will simulate the following aspects. You can treat each section to be a different service or feel free to architect as you see fit.

1.1 State of the battery:

- State of charge in MWh
- We need to keep track of the total import and export of energy:
  - On a given day, and;
  - From go-live to current simulation time.

1.2 Optimiser

- This is the core service responsible for deciding when to charge (buy energy through **bids**) and when to deplete (sell energy through **offers**) the battery.
- **It should consume the ML predictions of bid/offer prices and aim to maximise the revenue we generate each given day (00:00 to 23:59), by respecting the battery asset specifications, i.e. try to buy (bid) when the price is predicted to be the lowest and sell energy (offer) when the price is the highest, doing a max of 2 cycles/day while the battery never goes above 10MWh and lower than 0MWh.**
- For simplicity, submit the prices of the ML predictions for the periods you want to bid/offer for.
- At each simulation interval, you need to submit bids/offer pairs of the following format:

```
# submitted bid/offer pair; intention to offer in this example
```

```
{
   "submissionTime": "2021-10-04T16:00:00",
   "settlementPeriodStartTime": "2021-10-04T17:00:00",
   # submitted offer
   "offerPrice": 100.00, # use 9999 if you don't want to offer
   "offerVolume": 10.00, # use 0 if you don't want to offer
   #submitted bid; we don't want to bid for this period
   "bidPrice": -9999,
   "bidVolume": 0.00,
}
```

- Each simulation computation should have a step of 30 minutes.
- For each execution the following should be logged in json format (**terminal and to a file**):
  - Simulation timestamp
  - Battery state of charge (in MWh)
  - Total energy throughput from start to date (i.e. the sum of import and export in MWh)
  - Total energy throughput for the day (i.e. the sum of import and export in MWh)
  - ML bid and offer price predictions
  - Submitted bid/offer pairs
  - Bid result (same as submitted bid with a flag denoting if bid was accepted)
  - Offer result (same as submitted with a flag denoting if bid was accepted)

Deliverables:

- A zip folder containing the root directory of your solution
  - Module for each service
  - Dockerfiles
  - docker-compose file
  - Project README including instructions how to run the simulation

Part 2: Deploying services

For this task there is **no need to write additional code**.

Task:

Please provide a description of how you would deploy your solution on a cloud provider. Do feel free to use specific cloud offerings in your description, e.g. DynamoDB if using AWS.

Deliverables:

- Any document/ text file containing your solution.
- A diagram if appropriate.

Once we have reviewed your submission, we will schedule a follow-up call to discuss your proposed approach.

Good luck and we are looking forward to seeing your solution!