Dispatch Service Take Home

Dispatches are core to our work at Voltus. Energy Markets ask us to dispatch our assets and we then ask those assets (our customers) to reduce their energy use. We have a web service at Voltus that handles this task. It holds metadata about our customers and the various Demand Response programs they participate in. When a dispatch is triggered (either through a web form or an api request) we send out emails and text messages, and alert any of our devices (Voltlets). Our dispatch service must meet a few requirements:

1. It must be reliable. Incorrectly ignoring or missing a notification from a market could be detrimental to our business.
2. It must be flexible. We receive many different types of signals and alerts through various mechanisms, such as emails, api calls and webhooks.
3. It must be auditable. We want to be able to know "what happened" and provide information about an event to whoever needs it.

We would like you to create a simple dispatch service. It should have an input (api, or web form, or other) that takes dispatch information, and it should have an output (send an email, sms, alert a webhook). The server should have metadata about various "Programs" and should be able to alert different groups of people depending on which "Program" was dispatched.

As an example:

You might make a web request like so (we have to support open-ended events so the event might not have an end time) :

```
curl -X POST -H 'Content-type: application/json' --data \
    '{"start_time":"08/27/2019 21:04", "end_time":"08/27/2019 22:04", \
    "program_id":1, "event_type":"market_dispatch"}'
http://localhost:8080/
```

And as a result of that web request I might receive an email with the following content:

```
Dear Voltan,

You have been dispatched as part of the Program "Voltus Interview".

Please have your full curtailment plan in effect between the hours
of 9:04pm and 10:04pm
```

You will be judged based on the performance of your server in the context of our previously mentioned requirements. Eg:

1. Is it reliable? This is just a take home, so it doesn't need to be incredibly resilient, but have you described how it might fail and how it's going to still perform reliably in those cases? What does the service do if it alerts on slack and slack is unavailable?
2. Is it flexible? Would it be easy to extend it to support more inputs and outputs?
3. Is it auditable? Can I easily see what's happening/what happened?

You’re welcome to add more than one input or more than one output, you are welcome to add UX/UI niceties, you’re welcome to build it up into a docker container or deploy plan, but you will mostly be judged on the requirements above and the overall quality of the project and your code.