Assigument 2 Design sketch

Zhao Yanbo

system components:

① Aggregation Server: This is the central server, this Server responsible for handling it TTP GET and Put requests, managing weather data, and maineaining synchronization. using campout clocks

② Content Servers: These servers are responsible for collecting weather data, coverting it to JSON format, and uploading it to the aggregation server using at the put requests.

③ Client Applications: This part that will shown to the user to retrieve weather data from the aggregation Server using HTTP GET requests

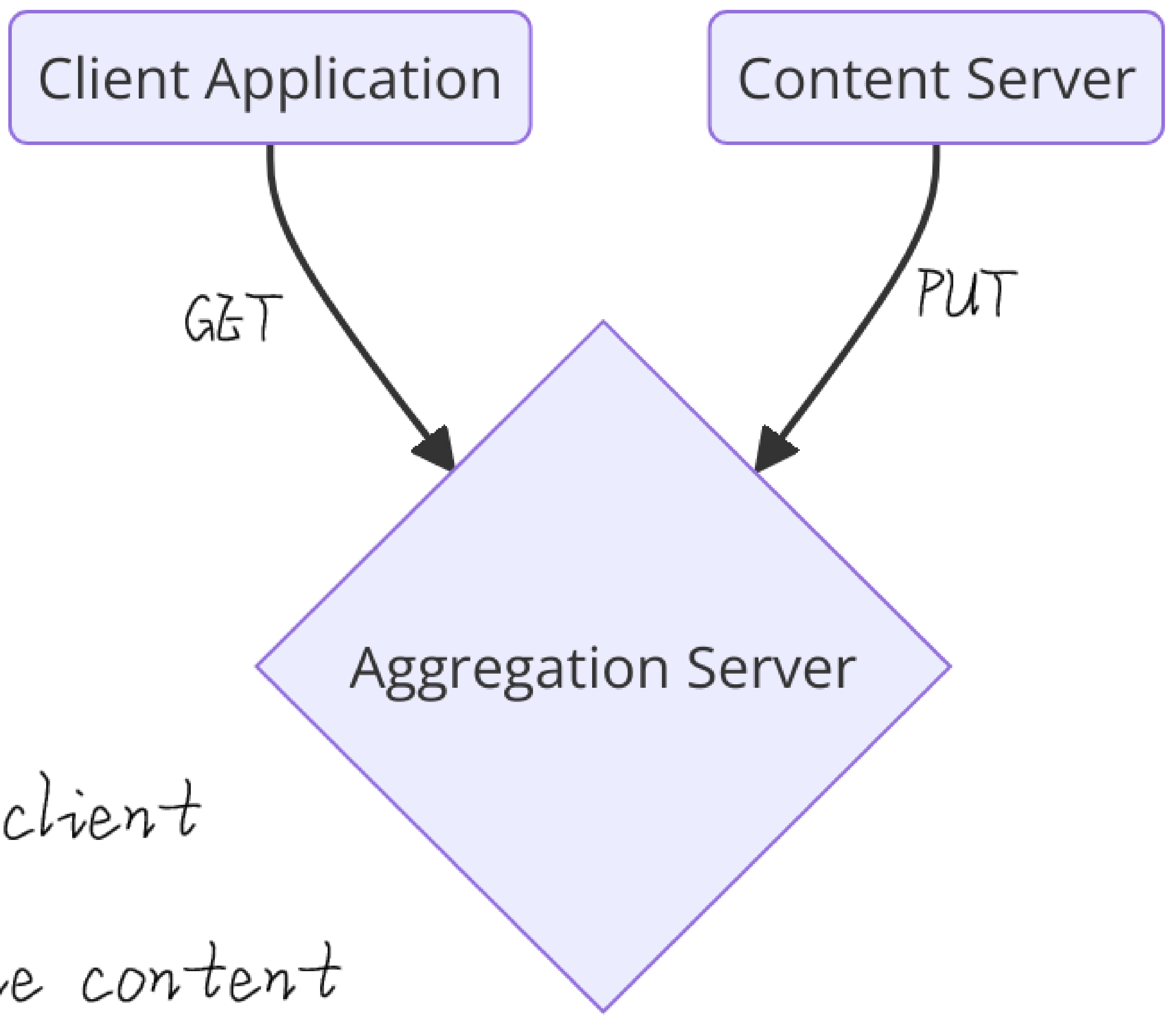
Communication and Data flow:

① From Content Servers to Aggregation sever:
The content servers will read local weather data, then rouvert it in to JSON, and send it to the aggregation server using PUT request.

② From client applications to Aggregation server:
The clients will seert Get requests to retrieve the wenther data, and the aggregation server will return the newest weather data.

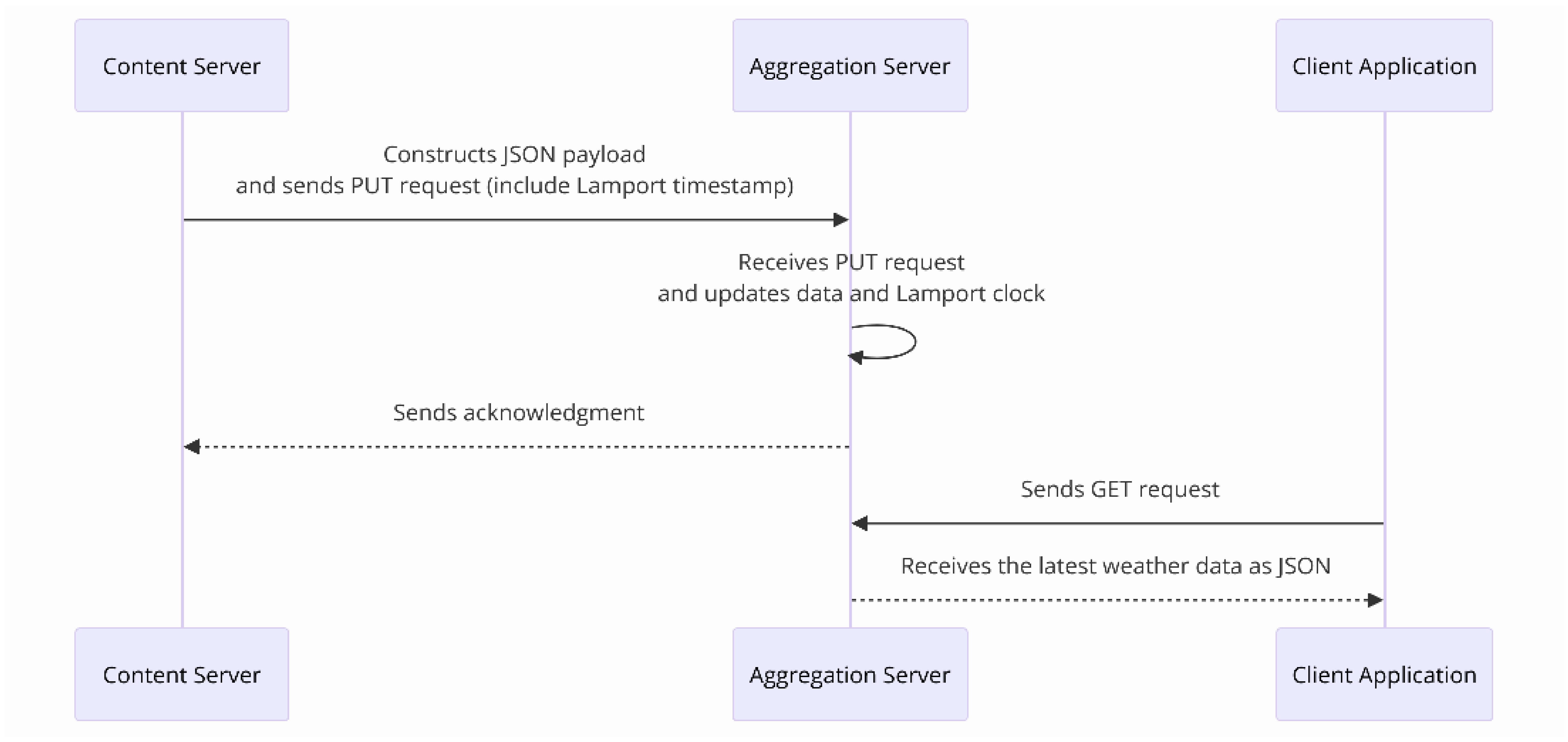
③ Aggregation server:
The aggregation server will handling all Put requests by stored the weatter data and if there are multiple concurrent put requests it will adjust based an Lamport time stamps to serialize the multiple concurvent Put requests

Diagram for components



The awars of the client application and the content server point to the aggration server to indicate the direcation of their communication.

Diagram for sequence



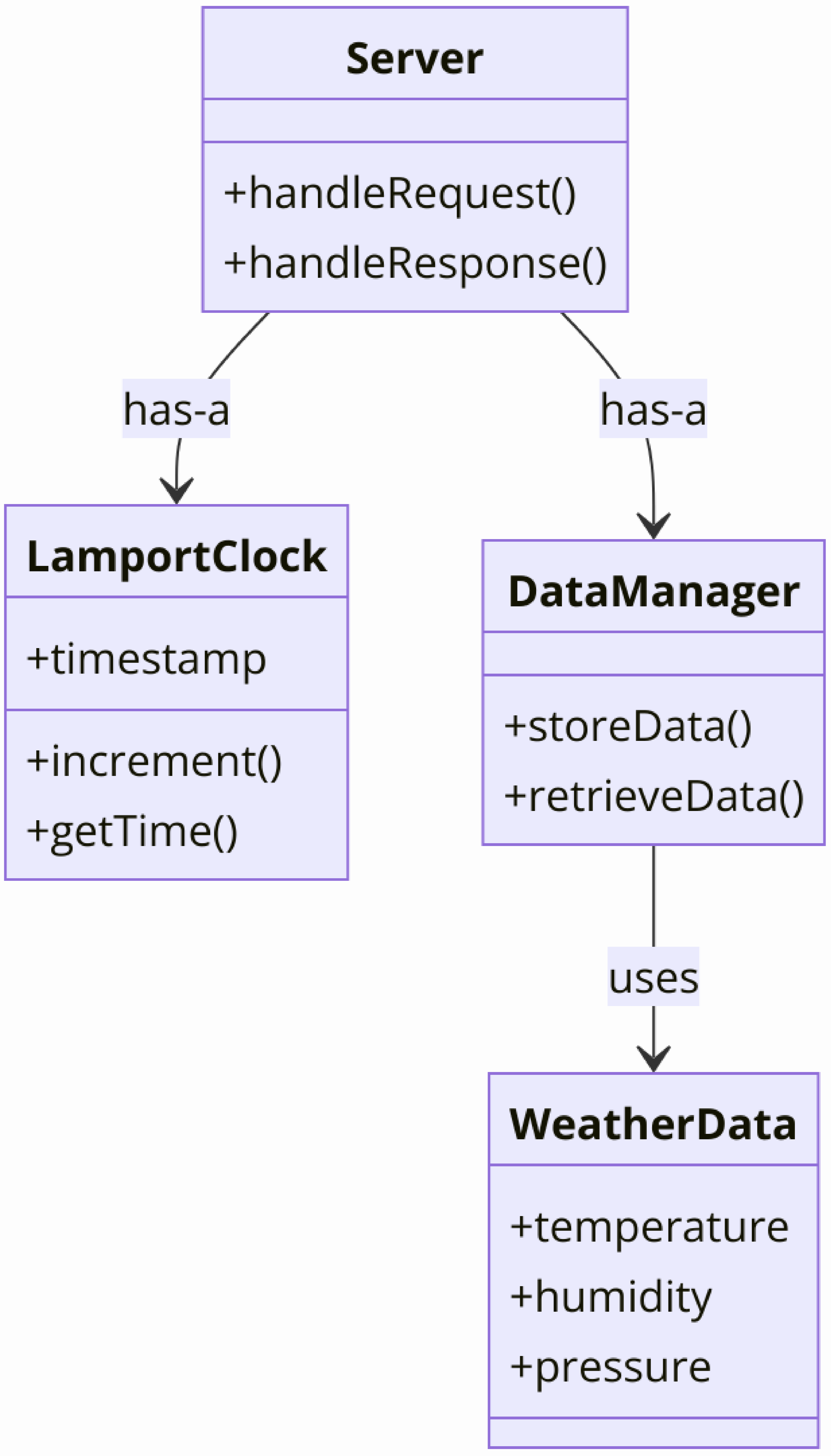
Thread safety and synch wouization problems:

For the multi-thread system, I will be deploying separate threads to hanlde the inflow of neework nequest efficently. I think I will utilize Syachronization mechanisms to ensure thread safety especially for operations like Put and Get that interact with shared weather data.

These will include locks and synchronized blocks to prevent any race conditions or unsafe mutations.

The preliminary implementation design of aggregation sever

Class diagram



These dasses are main dasses Within the aggregation server.

classes:

server: Main class to handle It the requests and responses.

Lamport claķe thisclass is to manage the logic of Lamport time stamps

Datumanager: This class is resposible tor data retrieval

Weather Data :this class is to store the weather data

· The implementation design of Lamport clock

I think each server and client should will maintain a Lamport clock, and the aggregation server should proress the sperathers order by a time stamp. Which every Put and Get request will have one.