

# ULM

AGGREGATION SERVER
- REQUEST HANDLER
- DATA STORAGE
- LAMPART LOCK
- EXPIRY HANDLER
- HANDLE REQUEST()
- STORE DATA()
- UPDATE LAMPART LOCK()
- REMOVE STALE DATA()

PUT  
REQUEST

CONTENT SERVER
- FILE READER
- DATA VALIDATOR
- PUT REQUEST SENDER
- READ FILE()
- VALIDATE DATA()
- SEND PUT REQUEST()

GET  
REQUEST

CLIENT
- REQUEST SENDER
- RESPONSE PARSER
- DATA DISPLAY
- SEND GET REQUEST()
- PARSE RESPONSE()
- DISPLAY DATA()

## FUNCTIONS/COMPONENTS

### CLIENT:

- DISPLAY: weather data
- REQUEST: get → AS
- ~~REQUEST~~ PARSE: JSON
- HANDLE: errors

MODULES: - REQUEST SENDER  
- RESPONSE PARSER  
- DATA DISPLAYER

### CONTENT SERVER:

- SEND: weather data: put → AS
- READ: weather data files
- VALIDATE: ensure format & make correct before send

MODULES: - FILE READER  
- VALIDATOR  
- REQUEST SENDER

### SYNCHRONISATION

- Use reentrant locks in sections with shared resource access
- Use thread pool to manage client requests
- Respect lampart clock ordering for PUT requests

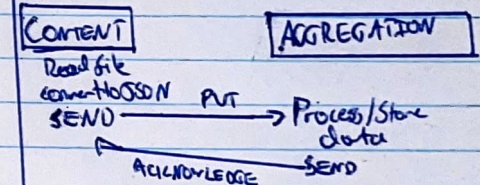
### AGGREGATION SERVER

- RELIEVE: Accept weather data in JSON format (Put)
- SEND: Provide weather data (GET)
- REMOVE: outdated data
- ENSURE: consistent clock handling (Lampart)
- MANAGE: invalid requests, server crashes, client disconnections

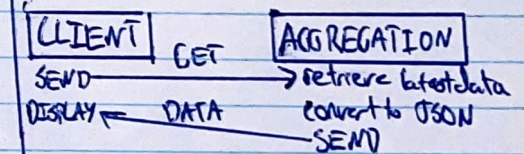
MODULES: - DATA STORE  
- REQUEST HANDLER  
- SYNCHRONISATION MODULE  
- DATA EXPIRY HANDLER

### DATA FLOWS

PUT:



GET:



### SERVER REPLICAS:

ACTIVE - PASSIVE REPLICATION:

- Synchronized backup replicas
- Take over if primary fails
- Lampart clocks synchronize data across replicas