Glue Code Pseudo Code:

- Start thread of Authentication Manager
- Start thread of Routing Manager
- Start thread of Communication Manager
- Start thread of Indexing Manager
- Start thread of Web Server
- Start thread of VoIP
- Start thread of Mailing System
- Start thread of Messaging System
- Start thread of DFS Storage
- Start thread of UFS Storage

- Initialize a status_flag corresponding to each input buffer of the Glue Code with 1. (This status_flag will contain values in powers of 2 such as 1, 2, 4, etc., upto some specific number limit such as 4 or 8.) (The default value for the status_flag for each module at starting will be 1.)
- Initialize a current_flag corresponding to each input buffer of the Glue Code with 1. (This current_flag will contain the current value; ie., after how many rotations the input buffer corresponding to that respective module will be checked.) (The default value for the current_flag for each module at starting will be 1.)
- The input buffers of the Glue Code will be checked periodically (only if the current_flag's value is 1. More about this in the next point.) for all the modules in Round Robin fashion. A thread will be running for each module in parallel. Each input buffer will be checked for 5 seconds or until all the data/query are extracted, whichever time is less.
- If the current_flag value is 1, then only the corresponding module's input buffer is checked. Otherwise, the current_flag's value is decremented by 1.
- If data/query is found in the input buffer, then the status_flag and the current_flag is set to 1.
- If no data/query is found, then the status_flag value is multiplied by 2, if it is less than the number limit which is set before (something like 8 or 16), else, it is kept the same as the number limit (8 or 16). Also, the current_flag is made equal to the status_flag.
- If any query is found in any input buffer, read it, analyse it, and then do the required operation.

- The required operation may be to give information (publish) to any module or to get some information (query) from any module.
- In such a case, the Glue Code will call the respective module directly via an API call.
- The modules will respond to such calls instantly with the required data/information (in case of a query). (As no response is needed in case of a publish).
- After the Glue Code receives the required information, it will send back the information to the module which initially demanded for it.
- In this case the information is directly put in the input buffer of that respective module which initially demanded for that information.
- All these while, the Glue Code will also update all the required statistics and data to a web browser using the web-server module. (This will be the user interface; to be done using React JS).

Algo has a problem.

→ how the status flag is set?
It seems to be hard coded.

when current flag is 2,
next time → no read and but
it will be decremented.

If in next iteration again, no message found → then it is now 2 by default

Current flag can take only 1 and 2 → two possible values.

Probably → You were looking for setting 1→2, if after two read again message not found → making it 4, again message not found then making it 8. Ideally if after two, you get message less than 5 msec worth, keep same, if more than reduce