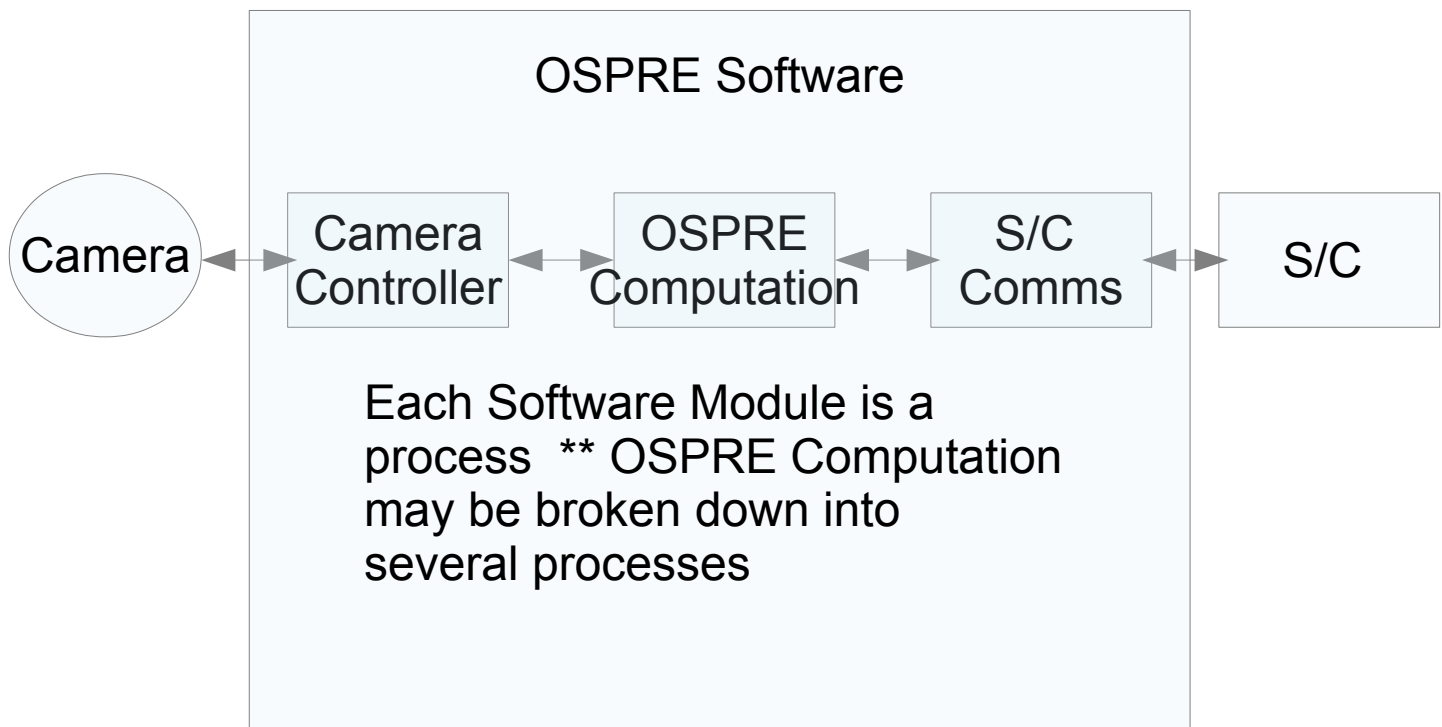


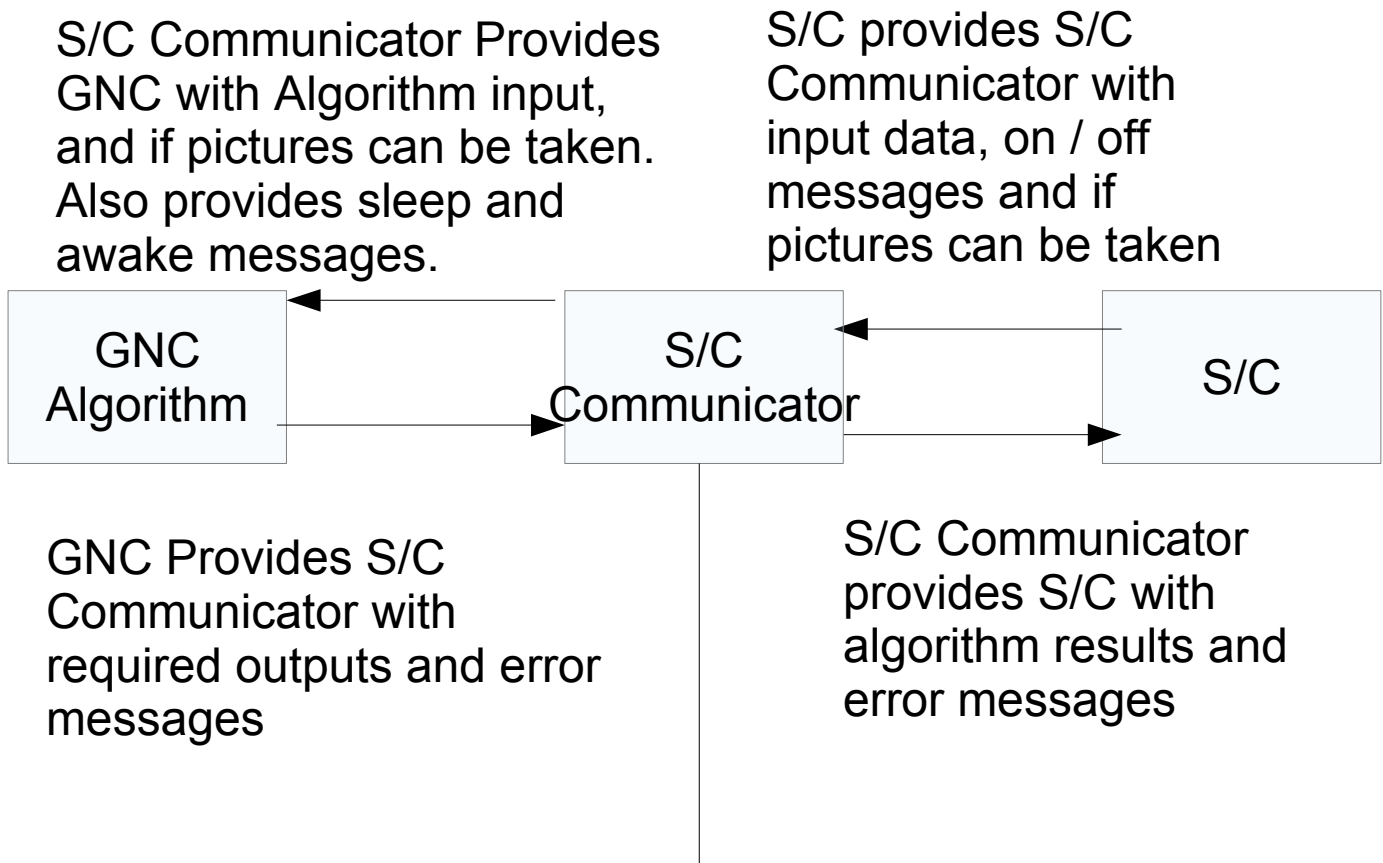
By using multiple processes, the OSPRE system maintains encapsulation from the outside systems and allows for implementation in either serial or parallel configurations.



** I am expecting to use TCP/IP sockets for intra process communication. All communications will utilize non-blocking sockets. The “I am alive and functioning” aka heartbeat messages will be sent over separate sockets. Error messages will be sent back directly to the S/C communicator to a dedicated unicast socket on the S/C communicator for error messages. **

S/C Communicator

Each box is a process



The S/C Communicator will be broken down into three sections:

1. Run overall operations of OSPRE
2. Monitors the health, and status of OSPRE
3. Communicates with the external S/C

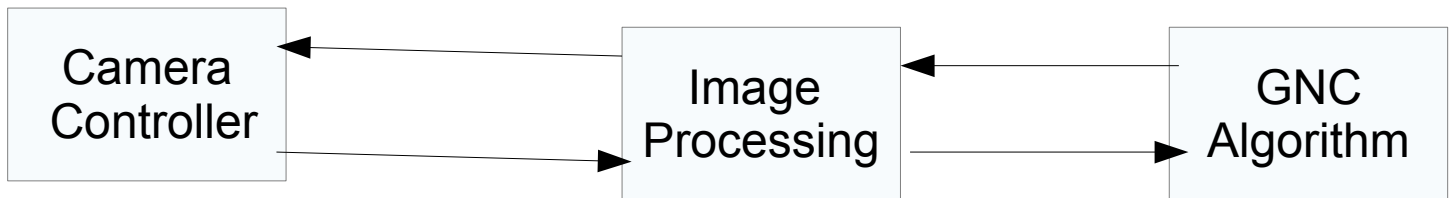
The S/C Communicator will be the only software module communicating with the S/C. Therefore, the S/C Communicator module insulates the OSPRE system from the S/C. If the S/C changes, the S/C communicator is the only thing that needs to change. The S/C Communicator will also act as the “controller” of the operation. It will receive with all error messages and decide what to do. The S/C communicator will be blocking in a selector with multiple timeouts. Every process keeps track of how long it takes the processes it has made a request of to fulfill its request. If the request times out, an error is raised and sent back to S/C Communicator.

OSPPE Computation

Each box is potentially a process

Image Processing will command the Camera Controller to take a picture and pass new camera settings based off previous pictures

The GNC Algorithm will ask Image Processing to take a picture, tell it which object is in its field of view, and the estimated range from that object



The Camera Controller will send Image Processing the image and a camera health message

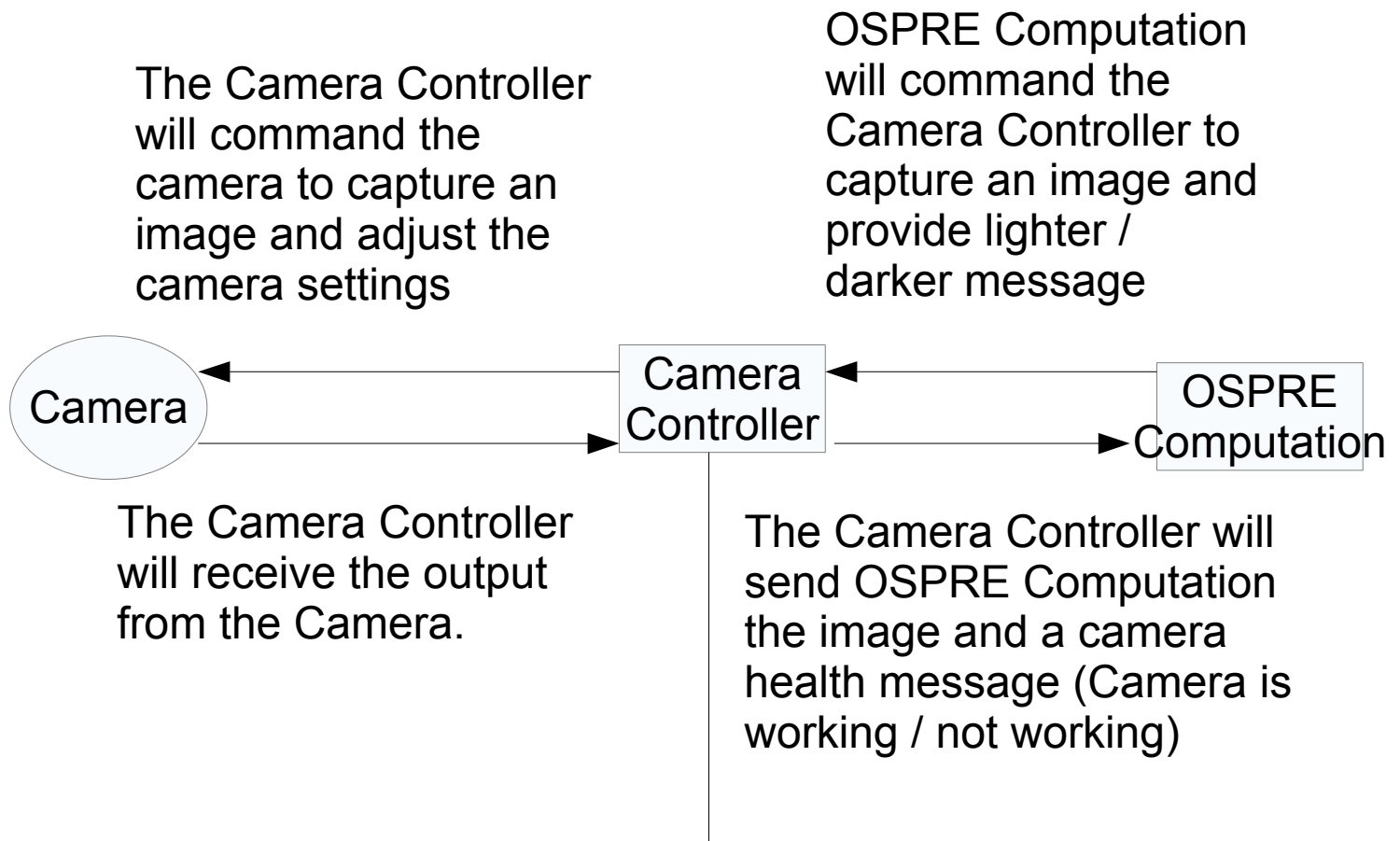
The Image Processing will provide the angular diameter and Unit Vector to the body. Image Processing will provide a pixel error associated with the provided measurements. It will also provide camera health, and Image Processing Error Messages.

** Image Processing has the capability to run in Parallel with Camera Controller and GNC Algorithm if desired **

Image Processor and GNC need to periodically check for messages while it is in the process of doing expensive calculations.

CAMERA CONTROLLER

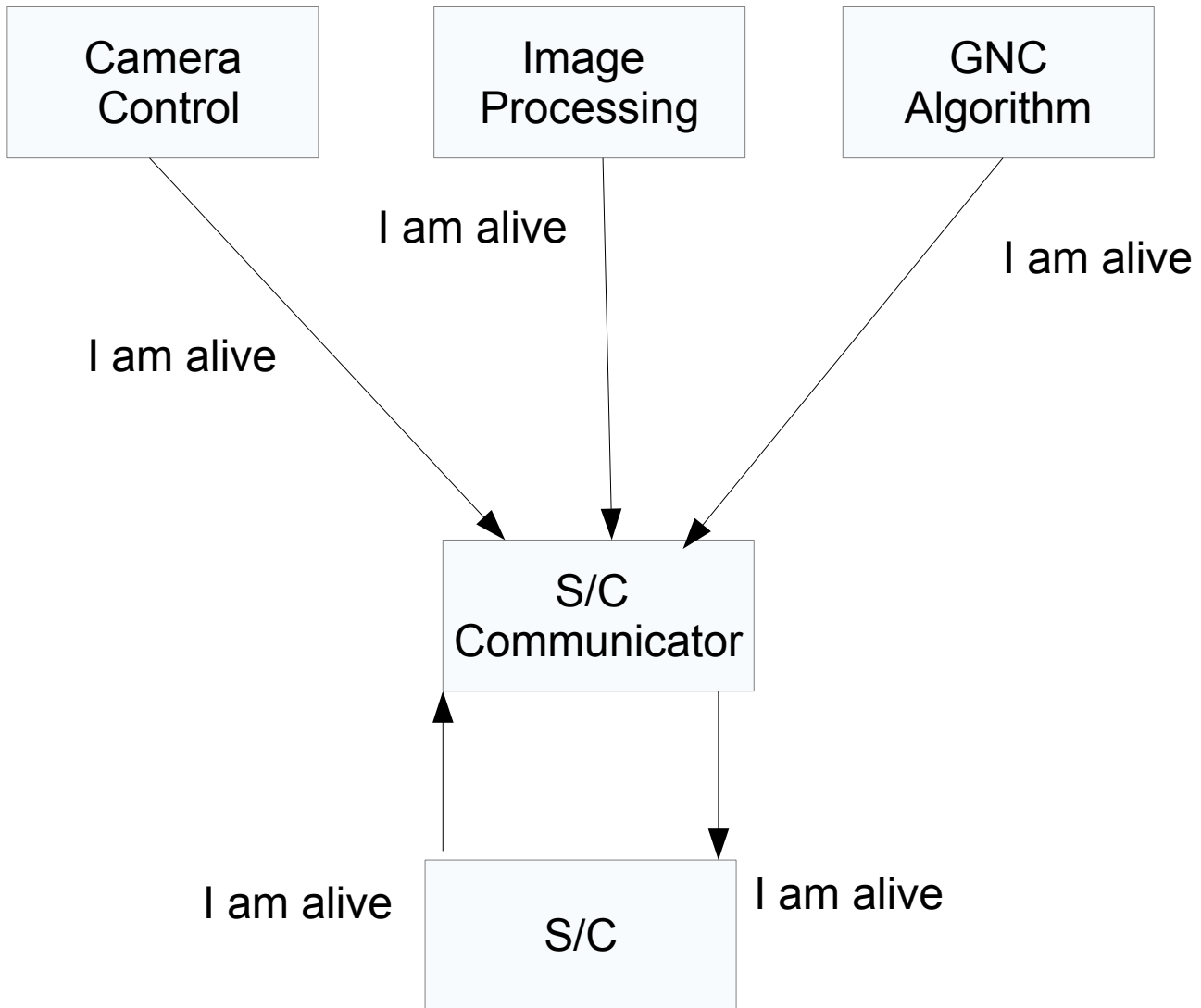
Hardware are ellipses
Processes are squares



The Camera Controller will be the only software module communicating with the camera. Therefore, the Camera Controller module insulates the OSPRE system from the specific protocols and implementation details of the specific camera utilized. If the camera gets changed, only the Camera Controller Module will need to be changed. The Camera Controller will have the logic embedded inside to determine how best to adjust the camera settings to achieve Image Processing request.

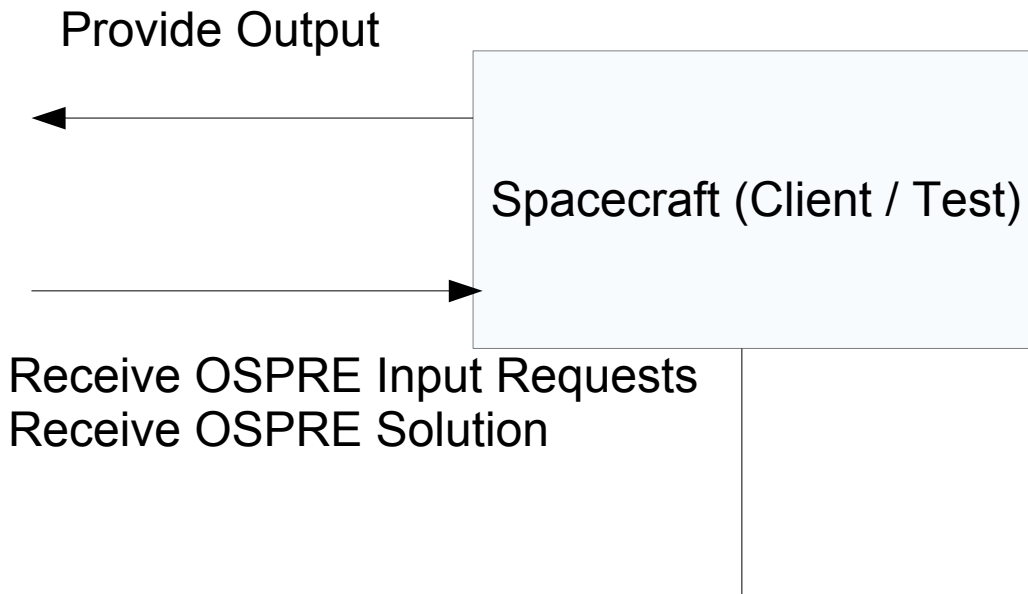
“I am alive and functioning”

Each box is a process



If connections fail, they will be attempted to be reestablished. After a timeout, the software module will be restarted. The S/C Communicator is responsible for restarting and restoring state to any OSPRE software module that fails. If the S/C Communicator fails, S/C is responsible for restarting it. The S/C communicator will save important information to non-volatile memory to recover its state on startup. If file is empty / corrupt, the OSPRE system will start fresh.

Spacecraft (Client / Test)



The Test / Client software interface will provide the least possible amount of information for the OSPRE system to run.

Client Must Provide:

1. Start OSPRE system
2. Command OSPRE to go into Passive Standby State and Active State
3. Inform OSPRE that it can / cannot take pictures
4. Receive Input requests and respond with necessary information
5. Receive OSPRE Solution
6. Request a current Solution despite error

#4 means that the Spacecraft Software Module will be capable of simulating the quaternion, time, ephemeris and sun angle