General Agreed Facts

- All services are registered to be automatically started when the PI starts up

- To reduce the general load on the RabbitMQ Backbone, just send messages if a value changes. If it is the same, avoid sending the same value over and over

# Services and their Behaviour

INIT:

1. Check if drone is connected

2. Check if arming the drone is possible

3: Send the status (good/bad) to the status module, where it is used for the \_\_px4\_running flag

STATUS:

The status module has dedicated flags to reflect the statuses of all the other modules. But those flags should only be used internally by the status module. To reflect the general status of the system, the \_\_system\_ok flag is used. The status module must contain logic to set the value according to the dedicated module statuses.

- \_\_sensor\_ok and \_\_memory\_ok flags can be removed

DATA\_PROCESSING:

1. Checks if sensor are connected

2. Constantly checks if received values are valid

3. According to the availability of the sensors and the values, the status is sent to the status module.

4. If the values are valid, they are forwarded to the logic module

LOGIC:

The exact logic needs to be defined by Oli. But generally, the logic modules only listens for the \_\_system\_ok flag. If it is good, it continues, if not it stops.

LOG:

The log module is used to persist messages. Every other module sends whatever logs they want to this module

Message Format

The message format consists of a string divided in 3 parts with the following format:

**SERVICE\_NAME:FLAG[OPTIONAL]:PAYLOAD**

Example= *init:\_\_px4\_running:True*

If there’s no flag to be sent, just payload, then the FLAG part is omitted=

Example = *data\_processing:{front = 201, side = 221, ground = 500}*