While Aerostack can be viewed as a five layer architecture (Reactive layer, Executive layer, Deliberative layer, Reflective layer and Social layer), the code base is organized as processes and systems. A process is the basic component of the architecture. A process has two aspects – (i) what it represents and (ii) how it is implemented. The first aspect can be simplified as a series of actions taken in order to reach the goal. The second aspect can be viewed as a data processor that receives input data, processes the data and generates the output. Every process has a function associated with it. For instance, the "Obstacle distance calculator" process' function is to avoid obstacles.

Processes are further grouped into systems. A system comprises of interconnected processes that provide the same functionality/capability. The list of processes along with their respective systems and capabilities that are available in Aerostack are provided in this link https://github.com/Vision4UAV/Aerostack/wiki/Catalog-of-Available-Processes. As we look at the list of capabilities in the above link, they are very similar to the skills that we would expect in drones. These capabilities are more or less like on/off switches that can be declared when we need them. I also think that the question of whether it is better to have the user specify the mission, or having the user identify the skills required and manage them or the user having complete control over the drone depends upon the skillset of the user as well as the amount of autonomy that the user wants.

Having an architecture split up into five layers may seem unnecessary but I believe that in the long run as we improve the performance and energy efficiency of the drone, we can transfer the work load of some of the layers onto the UAV gradually, if not all. I also found that the Telekyb framework has just two interfaces (Human and Hardware/Sensor) and one control layer which comprises all the operations. The main issue with this approach is that there is no freedom to add new capabilities. There is a provision to exchange the modules present already as per the user's needs but that does not allow the user to add new functionalities. Because of this drawback, I think that the developers of Aerostack tried to incorporate the five layer architecture.