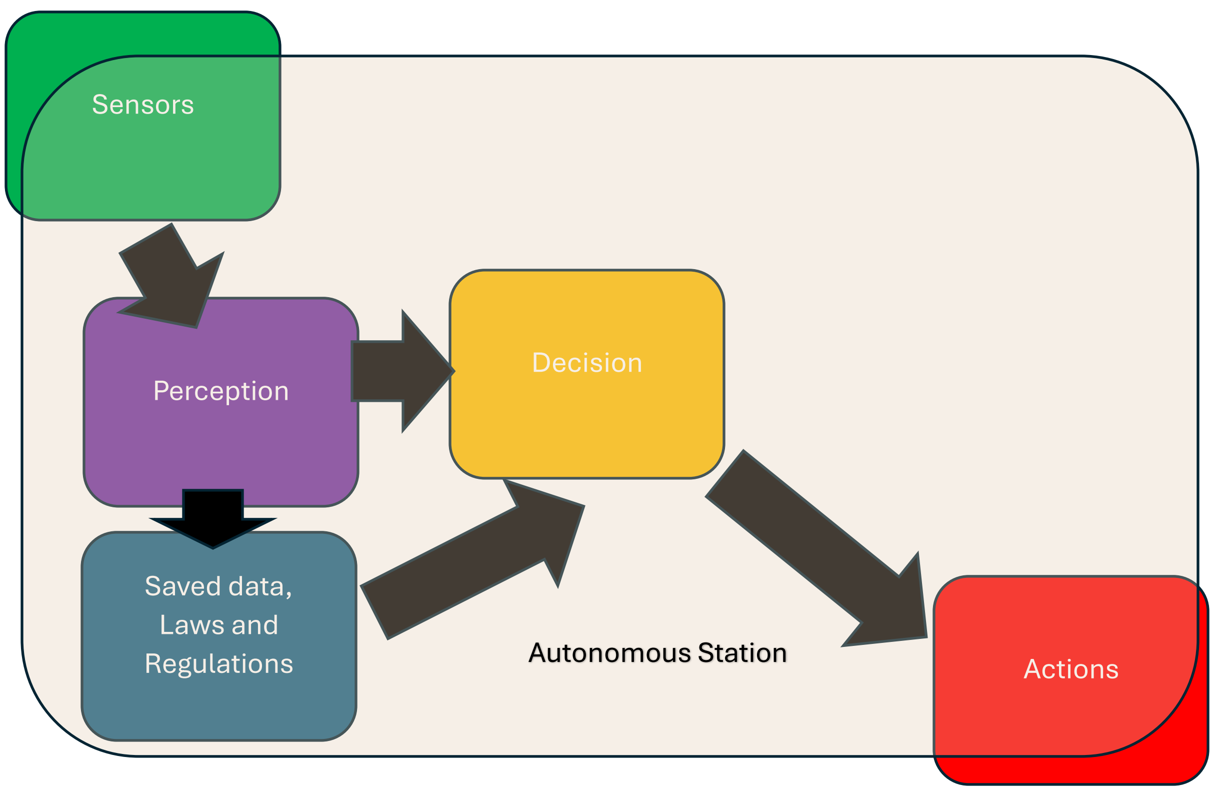
3.1 and 3.2 System Diagram

The chosen system is an autonomous fuel extraction and refilling station in space, meant to allow the rocket Starship to get refuelled in orbit around the Earth. Since Starship uses liquid rocket fuel and has too small tank to both lift off from Earth and travel to Mars, it needs to be refuelled in orbit before leaving. This is inconvenient to do directly by pumping fuel from one Starship rocket to another, as this would allow for a much smaller amount of fuel to be carried for each rocket trip. Thus, a station is needed. This station should be able to dock with a rocket, extract its fuel, move to another rocket and empty the fuel in the new rockets tank. Since space is a hostile place, and astronauts are few and very expensive, the station should not be manned. This means the station needs to handle environmental challenges, move and make different decisions on its own, which is why it needs to be an autonomous system.



The system consists of sensors sending info to a perception block where the info is processed. Then, the processed info is given to a decision block, together with relevant rules and regulations, and a decision of what to do is evaluated. This decision is given to the actions block which makes acts upon the decision. Some of the information gathered by the sensors should also be saved for later use as history of environment could be useful for making decisions.

The sensors will need to gather visual data on the surroundings of the system in 3D, which will in the perception block be used to get info on threats, objects and rockets to interact with. The sensors for the tank will also need to register piston positions used when transporting fuel from tanks. The laws and regulations are mostly related to safety of rockets, people in space around the system, and possibly anyone in distress in the future. But there are also some restricted orbits around Earth to look out for. The decision block needs to figure out movement, objectives and when to start and stop fuel transportation. Finally, the actions block controls the movement of the system and the start and stop of fuel transportation and the docking and undocking.

Though it is not needed at this point in time, it could be useful to add some communication with the ground, to potentially allow for instructions on when rockets will arrive, as well as status reports and future updates to software. This would have been a communication part giving data to the saved data and the decision part. However, this also introduces a larger risk of hacking as the system is open to outside influences.

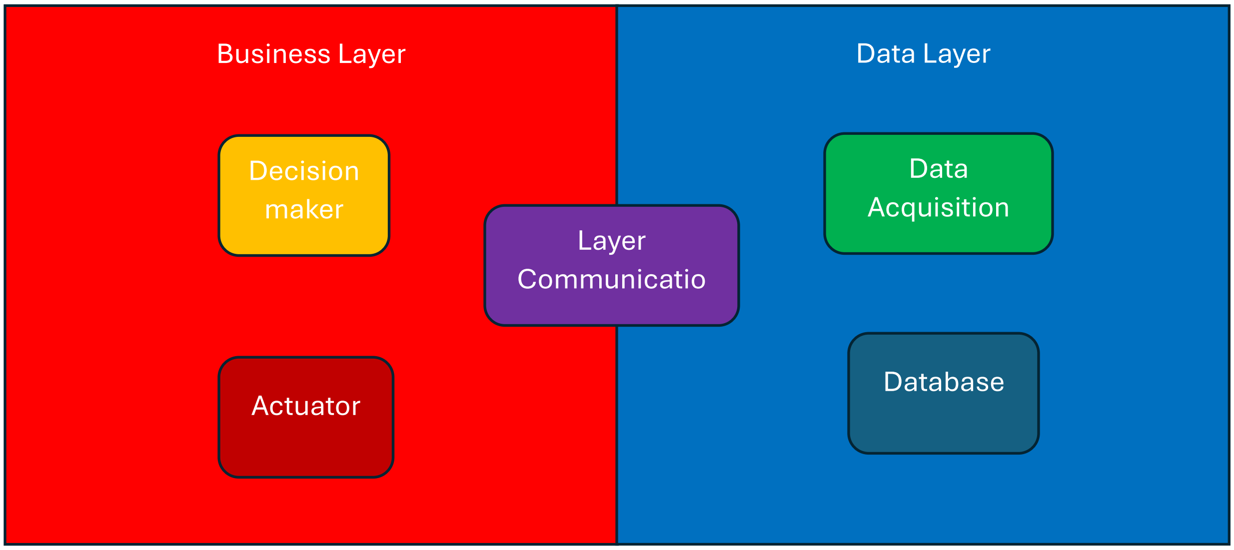
3.3 Organizing Software

The software for the autonomous extraction and refuelling station will be organized into two layers. The business layer and the data layer. The data layer will perform the gathering of information and storage of the important data to be kept for later. The business layer will be in charge of processing the information, making decisions based on the information and acting upon the decisions.

For the data layer, the system needs a database to store data needed for later and the map of its surroundings. This could be an SQL database or any other database, but it should have options for storing laws and regulations, map of surroundings with history of changes, status of internal tank and history of previous movements and goals. This layer also needs some sort of data acquisition devices for gathering from sensors and sending data to database. The types of data acquisition devices would be decided from the types of sensors used but should all be capable of communication over the same protocol to simplify the system.

The Business layer must be able to request data from the data layer and give this on to the decision maker. Then it needs a logical part which will make decisions based upon the information in the data layer. This part could be a program with all problems possible during operation coded into it. More useful however is some AI which can be trained to solve the problems it faces on the fly. Finally, the layer needs a program which can act upon the decisions. This program needs to control the motion of the station, the docking sequence, and tank pump. This program could be a PLC program, or some other high- or low-level language program capable of running on the hardware in the station. How the actual hardware for the different actuators can be controlled could limit what types of software tool can be used for this part.

A constraint for future models of the station is that the software and data should be unreadable by someone trying to hack into the station directly by plugging into the system or remotely if communication equipment are installed in the future. This is not that relevant yet, since hacking attempts would be hard now, but if space travel becomes more common, this will be something to consider. However, sniffer satellites which move up to other satellites and try to spy on them are already a thing, so access to the software should at least be protected.



3.4 Information Collection

The autonomous extraction and refueling station will collect information on the environment around it. This can be done with cameras taking optical images, or by a radar or lidar. The most useful option is optical imaging, as this has a long range and allows for orientation by using the Moon, stars and the Earth. This method also allows for identification of important objects, like rockets or identifying tags on rockets. However, the distance to the objects identified are harder to determine in this way. However, since there are very few of them, this can be done by taking multiple images and processing the change in size of the object. Then the objects can be compared to other similar objects in the databank and their distance and speed can be calculated. This is an ideal task for an AI to perform, as not all objects will be the same size and learning based on experience will be important. Sensors for checking the status of the tank in the station and potentially when a rockets tank is full or empty are also important, but these decisions are more trivial. A short-range proximity sensor can be used for the internal tank sensors, and a sensor similar to fuel stations refill sensors can be used for rockets tanks.

The planning and decision making for the system could be made by a normal program, acting in a specific way based on the situation. This would allow for a predictable station which can handle problems just as we would want. This would also lead to problems when unforeseen events occur, and thus put a lot of stress on the developer’s ability to predict problems. However, using an AI to plan what to do would be a more robust planner. An AI could adapt choices based on unforeseen events and learn from active experience in the field. Since the system is the first of its kind, this would be especially useful in dealing with new problems. On the other hand, this could lead to wrong decisions being made without anyone being directly responsible for the actions of the station. However, with careful use of simulations beforehand, most if not all these cases could potentially be avoided.

Finally, the actions of the system are either to move somewhere using small rockets, activating or deactivating docking locks, or pushing or pulling on a pump inside its tank. The rockets can use small parts of the same fuel they transport, to allow less refueling problems. The most optimal way would have been if it could have run on solar power, but electric rocket engines are not very good yet. The pump and docking mechanism can however work using electric power, and since the system needs electricity to run software anyways, this is ideal. This also has the added benefit of limiting the use of the fuel, which is of course the main point of this station.

There is no sensitive data stored which would be a direct problem if it was leaked, but the vulnerability of fuel being stolen, or the station being sabotaged is significant. This means that the decisions made and how the actual software works needs to be protected. Hackers should not be able to predict how the system will make its decisions, and not be able to influence these decisions in a way which is not ideal for the system. Since the system doesn't contain any communication devices at this point, remote hacking is not a problem, but this should be considered if relevant. However, physical hacking or spying from either sniffer satellites or other rockets should not be possible and if a physical service port is to be installed, the code and most of the data should be encrypted to hinder tampering. The likelihood of this being a major issue in the near future is, however, limited as such attacks would be of limited use and very expensive.

3.5 Learning

In an autonomous system, the most normal way to use machine learning is for its movement and decision making. Either by creating an AI which can find the best route to a place or by making one that can choose what to do at each step to go in a specific direction. Machine learning can also be used to recognise objects of interest around the autonomous system. This is more like information processing than decision making, however it is still just as useful in this context. Trained AIs can also be used for prioritising problems and decision making when handling multiple problems at the same time. This can be trained in simulations beforehand and run continuously during operation to allow for better model with experience from almost all problems.

In the case of the automatic extraction and refuelling station, it is possible to use machine learning for most of these cases. Image recognition and decision-making is perhaps the most important, as the station will have to be able to recognise rockets and could have to deal with both low amount of fuel, energy, dangerous objects on collision course and a goal to reach at the same time. However, the general path planning is normally not that big of a problem once a goal is chosen, as most of space is empty, and the station must be able to move in three dimensions. This is still a situation where it can be used and would allow for possible extra doublechecking of decisions.