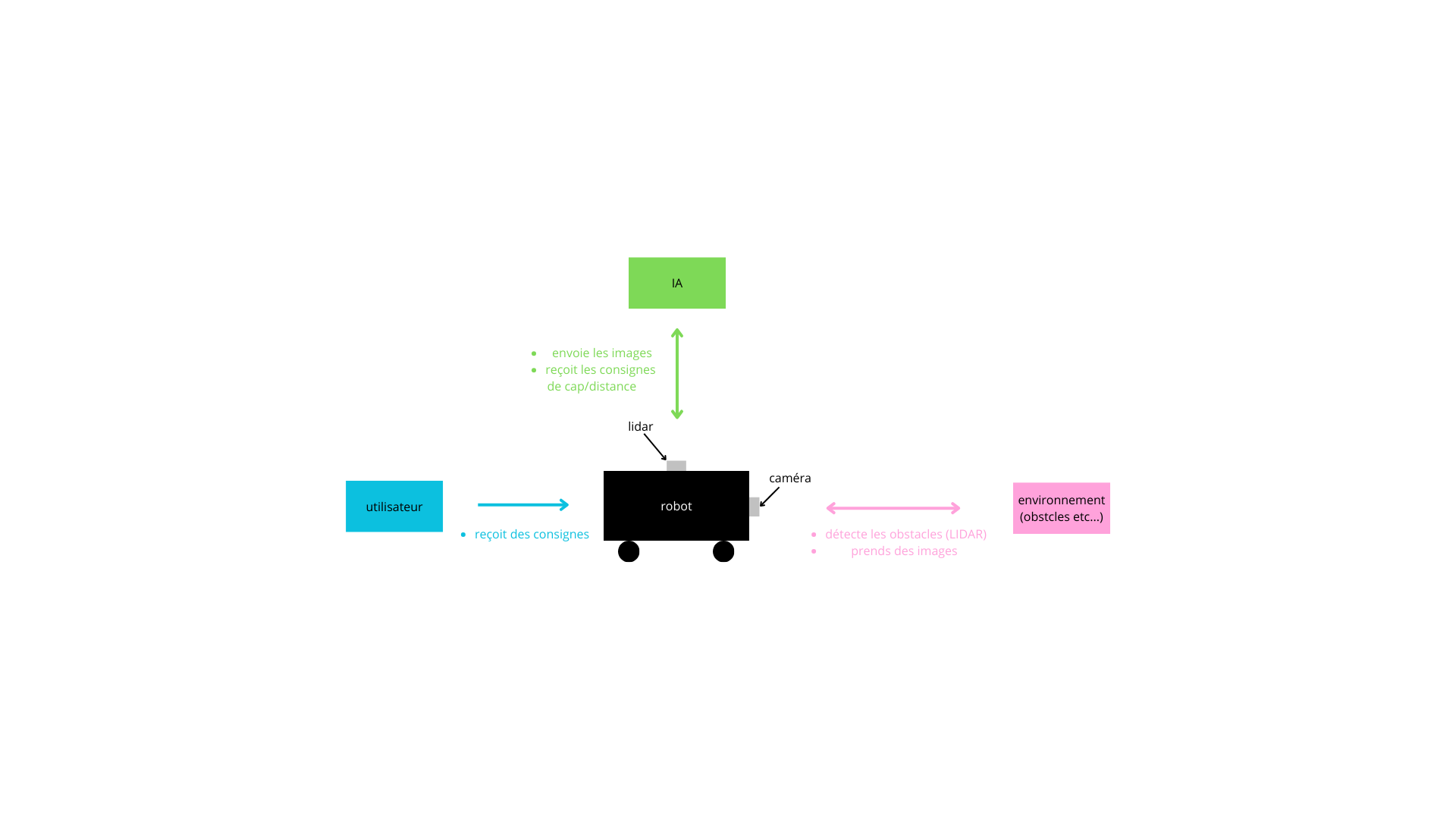
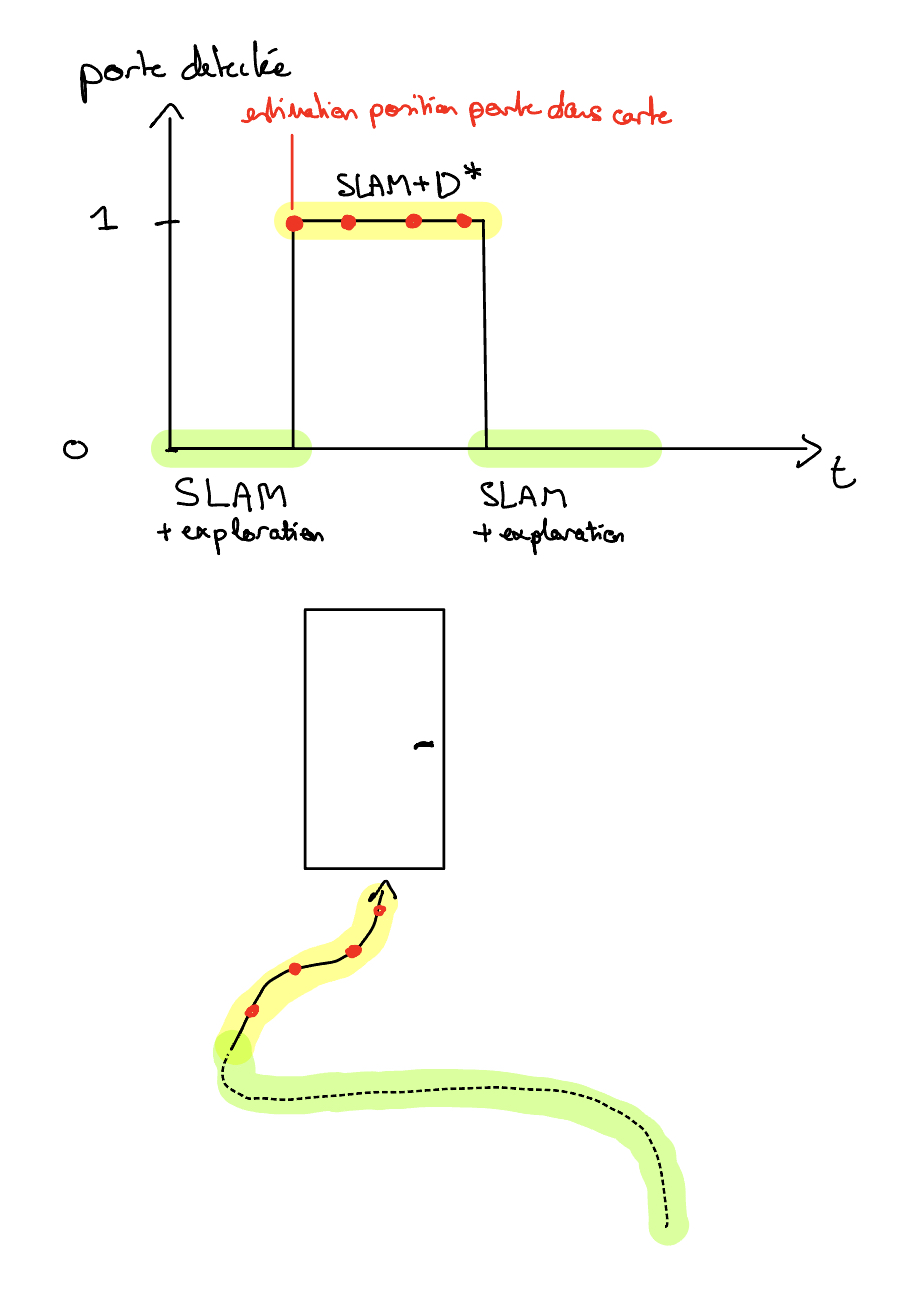
**TP1 : use-cases & architecture fonctionnelle**

1. **use-cases et exigences fonctionnelles :**

2) Description du système en une phrase (exigence source) :

Le système est un robot autonome capable de suivre des consignes de cap et de distance, et d’éviter des obstacles en communiquant en temps réel avec une IA qui analyse les images de ce qui l’entoure.

3) Exemple de use-case :

****

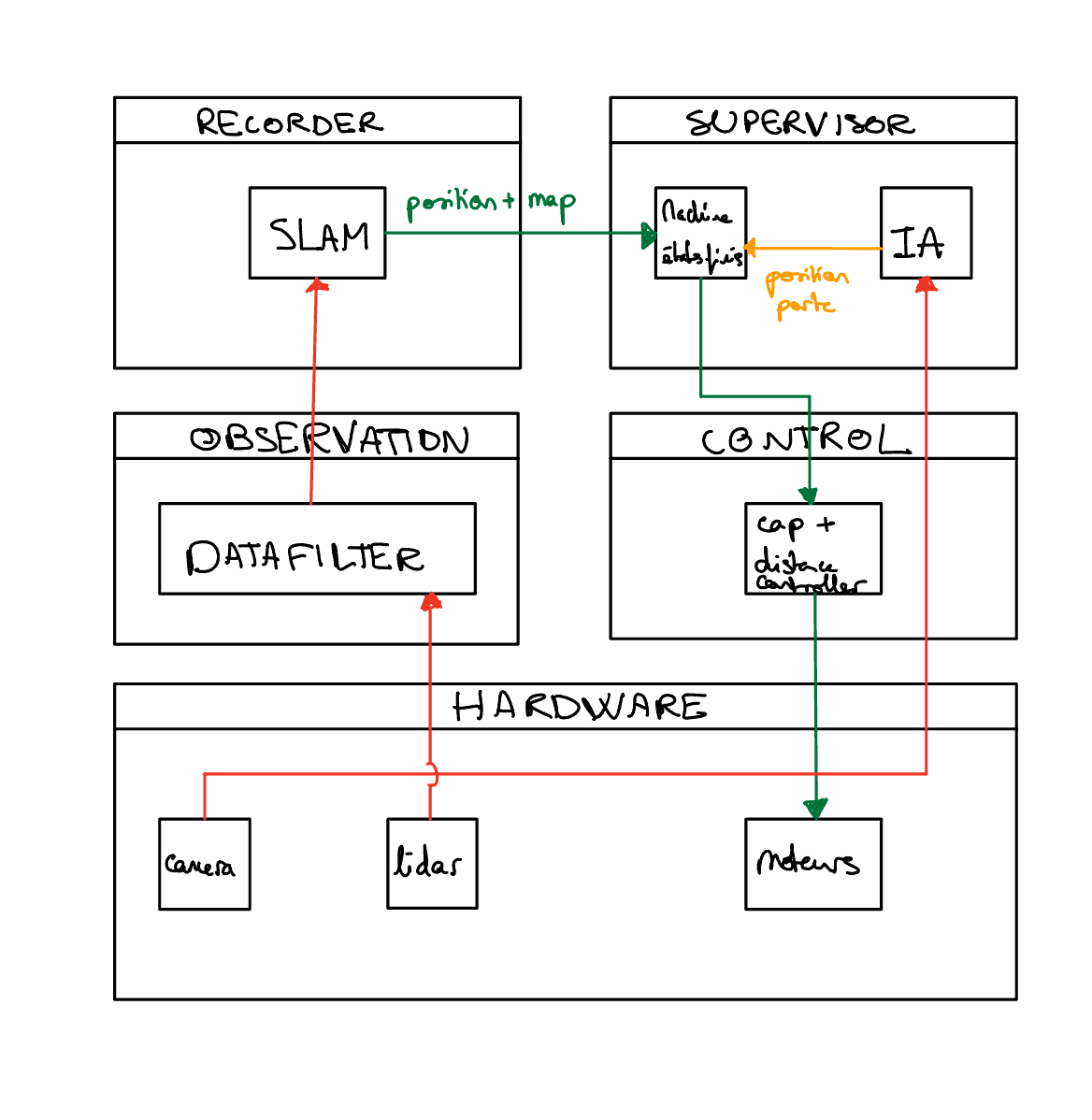
4) Liste des exigences fonctionnelles/ non fonctionnelles :

(exigences fonctionnelles = ce que le système doit faire

exigences non fonctionnelles = comment le système doit fonctionner (qualité, performance, sécurité, fiabilité etc…)

| **exigences fonctionnelles** | **exigences non fonctionnelles** |
| --- | --- |
| suivre les consignes de cap et de distance | réagir rapidement aux instructions données par l’utilisateur (ex : en moi d’une seconde) |
| détecter les obstacles et les éviter | être capable de se déplacer de façon autonome pendant 1h |
| communiquer avec l’IA en temps réel | fonctionner aussi bien en intérieur qu’en extérieur (être capable de détecter tout type de portes) |
| analyser les images en temps réel |  |
| trouver et suivre un itinéraire de façon autonome |  |
| être capable de suivre des instructions données par l’utilisateur (par exemple commandes vocales) |  |

1. **architecture fonctionnelle**

5) architecture fonctionnelle : 

6) identification des chaînes fonctionnelles :

* camera > data filter > IA > machine d’états finis > cap + distance controller > moteurs
* lidar > data filter > machine d’états finis > cap + distance controller > moteurs
* camera > data filter > log > communication > utilisateur

7) Communication Hardware // Observation :

RPLIDAR A1 <-> data filter:

* fréquence de balayage par défaut (environ 5.5Hz) (360° toutes les 5.5 sec : 1450 points de mesure par tour)

Camera <-> IA :

* fréquence de 10Hz

8) Description d’un algorithme d’un bloc fonctionnel à haut niveau :