**RSE2107A - Engineered System Context**

Introduction

* This document outlines the engineered system context which the focus is on understanding the system is not in isolation but rather within a wider operational context. Through application of system approach, this document has identified the Narrower SoI (NSoI), Wider SoI (WSoI) to outline the system boundaries and their relevant interactions.

System Context

* This project operates as a real-life scenario context where learners are expected to demonstrate their competencies to future industry employers. This project is evaluated by our university professors. They act as stakeholders for real-world clients, reinforcing the expectation of standards and delivery.
  + The project allow learners to demonstrate competency in two areas:

1. Technical competency – implementing mapping, localization, recovery behaviours, costmap layers.
2. Problem solving competency – Applying system approach to identify problems, analyse the system context, define requirements, and develop solutions within constraints.

* With these competencies, the project is approached as if learners were building a solution for a supervisor or client, rather than simply meeting academic requirements. The arena is built around the assumption that the robot may one day function in operational roles.

In this way, the robot is treated not just as a product, but expanded into a potential service enabler. For example, this context can be applied to a patrol robot operating in a real environment like an airport terminal where it must perform reliably within indoor conditions. This real-life scenario context prepares learners to build industry trust in their ability to deliver foundational robotic systems that could be scaled for operational deployment.

System-of-Interest (SoI) Definition

* + Refer to [**SEP1**](https://drive.google.com/file/d/1A7wBbzr6U-m1ovK3SBeWB1-Q2Gzw8l7R/view?usp=sharing)in artefacts folder for full diagram

Narrower SoI (NSoI):

* Team 8’s Terminal 3 plot (1.5 m × 1.33 m) plus the Agilex LIMO Bot and its ROS1 navigation stack.
* Reasoning:
  + Our team selected these as it offers a suitable setting to explore autonomous robotic functions. The combination of our allocated plot and the LIMO platform provides a well-defined space where key behaviours such as mapping, localization and navigation can be developed and tested. The plot size allows us to design, build, and place elements that reflect spatial constraints typical of indoor environments like airport terminals, enabling us to tackle navigational challenges.

Wider SoI (WSoI):

* The fully assembled eight plot arena.
* Reasoning:
  + Our team selected these because our demonstration must prove that the robot can map, localize, and navigate continuously across every plot, with recovery behaviors to handle any issues. Treating the entire arena and all bots as one system lets us verify mapping continuity, navigation tuning, and recovery routines exactly as they will run in the live demo.

Environment:

* ROS and Ubuntu OS
* Reasoning:
  + Our team selected these as part of the environment as they represent software infrastructure that enable our NSoI’s performance. ROS provides core functionalities for executing navigation functions and middleware services. Ubuntu provides system level support required for running the ROS packages. These components are part of the environment as they directly influence the performance of our robot.

Wider Environment:

* Scope of SEP1– Allocated $600 budget, theme, arena requirements, criteria and grading, timeline schedule
* Reasoning:
  + Our team identified these elements as they influence our decisions even though they fall outside the boundaries of the SoI. Budget constraints influenced our material selection and fabrication approach. Criteria and grading shaped how we build our arena design.
  + Although we cannot modify these parameters, acknowledging the wider environment ensures we recognise the limits and expectations within our NSoI operates.