SOFTWARE ARCHITECTURE

Overview diagram of runtime components

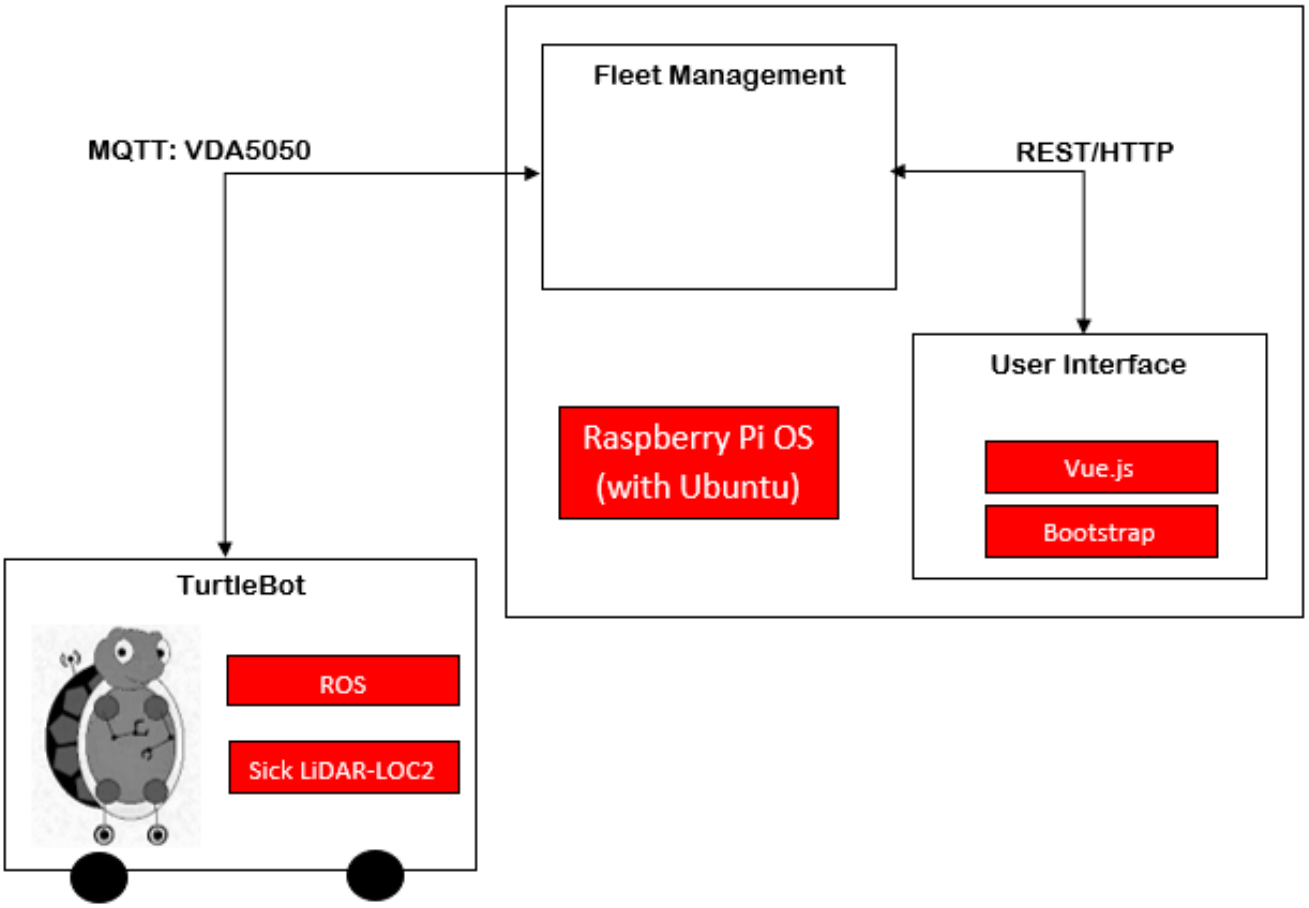

Fig. 1 Runtime components

Overview diagram of code components

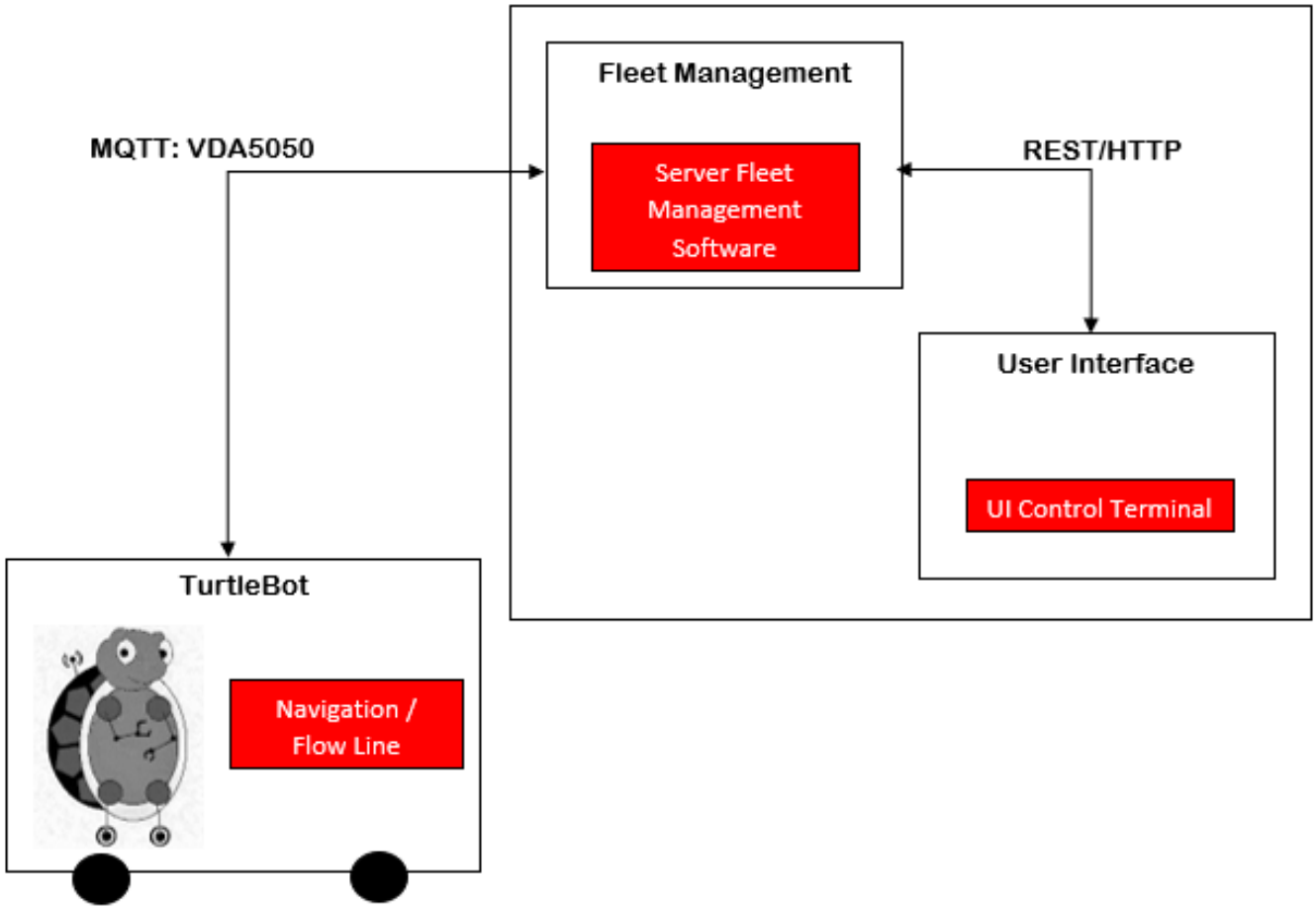

Fig. 2 Code components

Summary of the underlying technology stack

- **Robot (with LiDAR)**
  The robot TurtleBot is an open robotics platform designed for education and research on state of art robotics. It is also a powerful tool to teach and learn ROS and make the most of this cutting-edge technology. Equipped with a 3D sensor, it can map and navigate indoor environments. The 3D perception, together with the TurtleBot arm, enables manipulation tasks.

- **ROS** – Robot Operating System
  The Robot Operating System (ROS) is a framework that helps researchers and developers build and reuse code between robotics applications.

- **Wi-Fi/ MQTT/ VDA5050**
  - Wi-Fi is the wireless technology used to connect computers, tablets, smartphones, and other devices to the internet.
  - MQTT is a lightweight, publish-subscribe network protocol that transports messages between devices.
  - VDA 5050 is a standardized interface for AGV communication.

- **Raspberry Pi (Ubuntu version)**
  Raspberry Pi is a credit-card sized computer that plugs into a computer monitor or TV and uses a standard keyboard and mouse.

- **Python**
  As the programming language for the server fleet management software since it's most accessible among all the programming languages available and has simplified syntax

- **Docker**
  A software platform that allows you to build, test, and deploy applications quickly.

- **A web framework**
  - Vue.js – an open-source model–view–view model front end JavaScript framework
  - Bootstrap – Bootstrap is a free and open-source CSS framework

A textual explanation of the diagrams and choices

Runtime components (Fig.1) are existing technologies which can directly be used or implemented whereas, code components (Fig.2) are basically the software components which are to be developed from scratch to meet specific functionality requirements.

Runtime components

1. For the Robot

- ROS – The robot operating system quite simply enables researchers to build more complex robots in a much shorter amount of time. More interestingly, it is based on open-source code which offers higher flexibility at minimum cost.
- SICK LiDAR-LOC is a software for determining the position of automated guided vehicles (AGVs)

2. Together for the fleet management software & UI Control

- Raspberry Pi OS - It acts as a powerful combination to create smart robots, with a somehow low cost, and very small electronic board embedded in the robot.
- Ubuntu- The "default" operating system for Raspberry Pi is Raspbian. However, for using ROS, it is better to be served by using a Ubuntu version for the Pi. Installing ROS packages and managing them on Raspbian can be quite difficult, whereas on Ubuntu it works almost out of the box, just like on a standard computer or laptop.

3. For UI Control

- The web framework Vue.js can be used for building user interfaces and single-page applications.
- Bootstrap is directed at responsive, mobile-first front-end web development in short time.

Code components

1. **Server fleet management software for fleet management**
    - Coordinates a fleet of up to 5 TurtleBots
    - Receive requests from a user via UI Terminals
    - Schedule TurtleBots to fulfill tasks and recharge
    - Plan global route
    - Sent routes to TurtleBots
    - Keep track where the TurtleBots are
    - Convert a line layout to a graph
2. **Navigation or Follow line**
    - Implement a line follower module in ROS
    - Navigate the TurtleBot along the route given by the fleet manager
3. **User interface (UI) control terminal**
    - An interface to the fleet management

Interfaces between components

Wi-Fi/ MQTT/ VDA5050 are used for connectivity between robots and the fleet management software as specified by Sick.