

POLITECHNIKA ŚLĄSKA
WYDZIAŁ AUTOMATYKI, ELEKTRONIKI I INFORMATYKI
KIERUNEK INFORMATYKA

PRACA DYPLOMOWA INŻYNIERSKA

SYSTEM MONITOROWANIA I RAPORTOWANIA PRZEBIEGU MISJI
ROBOTA EKSPLORACYJNEGO

Promotor: dr inż. Dariusz Myszor

Konsultant: dr hab. inż. Piotr Przystałka, prof. PŚ

Autor: Sebastian Środecki

Cel pracy

Zaprojektowanie i implementacja modułu logowania danych w systemie robota eksploracyjnego – Łazika Silesian Phoenix II

Zakres pracy

- Zapoznanie się z projektem Łazika oraz jego systemem
- Zaprojektowanie modułu logowania danych diagnostycznych, z uwzględnieniem potrzeb grupy docelowej oraz specyfiki całego systemu
- Implementacja modułu z wykorzystaniem języka wspieranego przez Robotic Operating System (ROS) – Pythona.

Struktura modułu logowania pod kątem funkcjonalności


Aplikacja webowa

- Sterowanie logowaniem danych
- Przegląd informacji o systemie
- Konfiguracja logowania
- Przegląd logów

Logger

- Zbieranie danych pojawiających się w systemie
- Zapisywanie zebranych informacji do bazy danych

Widok interfejsu



Logger Service List of logs Logger state Logger configurator Device description

System Description

Topic	Type	Description
battery	int	battery level
rpi_cam_front	jpg	frontal camera mounted on manipulator
safety_button	bool	safety issues occurrences
velocity	real	rover velocity

Wykorzystane narzędzia

Aplikacja webowa

- Python3
- MariaDB
- Flask oraz powiązane: Jinja2, Werkzeug
- Bootstrap
- Matplotlib

Logger

- Python3
- MariaDB
- ROS

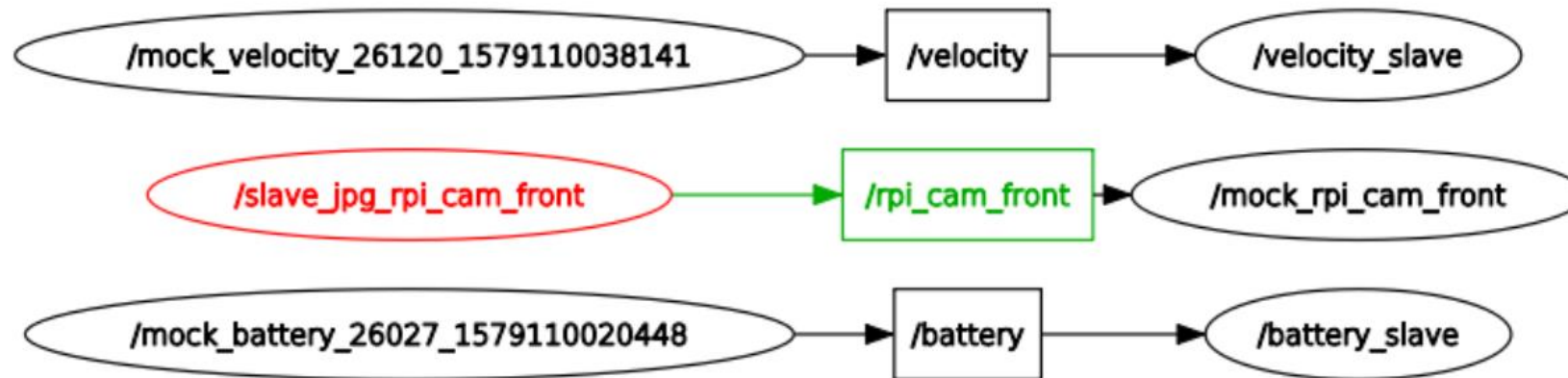
Podstawowe środowisko - ROS

- System umożliwiający komunikację programów działających na tym samym bądź różnych komputerach
- Niskopoziomowo, przesył komunikatów jest oparty o TCP lub UDP oraz rosmastera – program o działaniu podobnym do serwera DNS
- Na wyższych warstwach – abstrakcja tzw. topicu, na którym publikowane lub subskrybowane są wiadomości o określonym typie (proste lub złożone)

Architektura modułu – Logger

- Komunikacja między loggerem a aplikacją webową odbywa się za pośrednictwem bazy danych
- Logger składa się z procesu głównego (mastera) oraz procesów pobocznych (slave'ów)
- Master odpowiada na żądania operatora i zarządza logowaniem danych
- Każdy slave zbiera informacje z jednego topicu i zapisuje je do odpowiedniej tabeli w bazie danych
- Tabele danych używane przez
- Slave'y istnieją tylko wtedy, gdy zbierane są dane

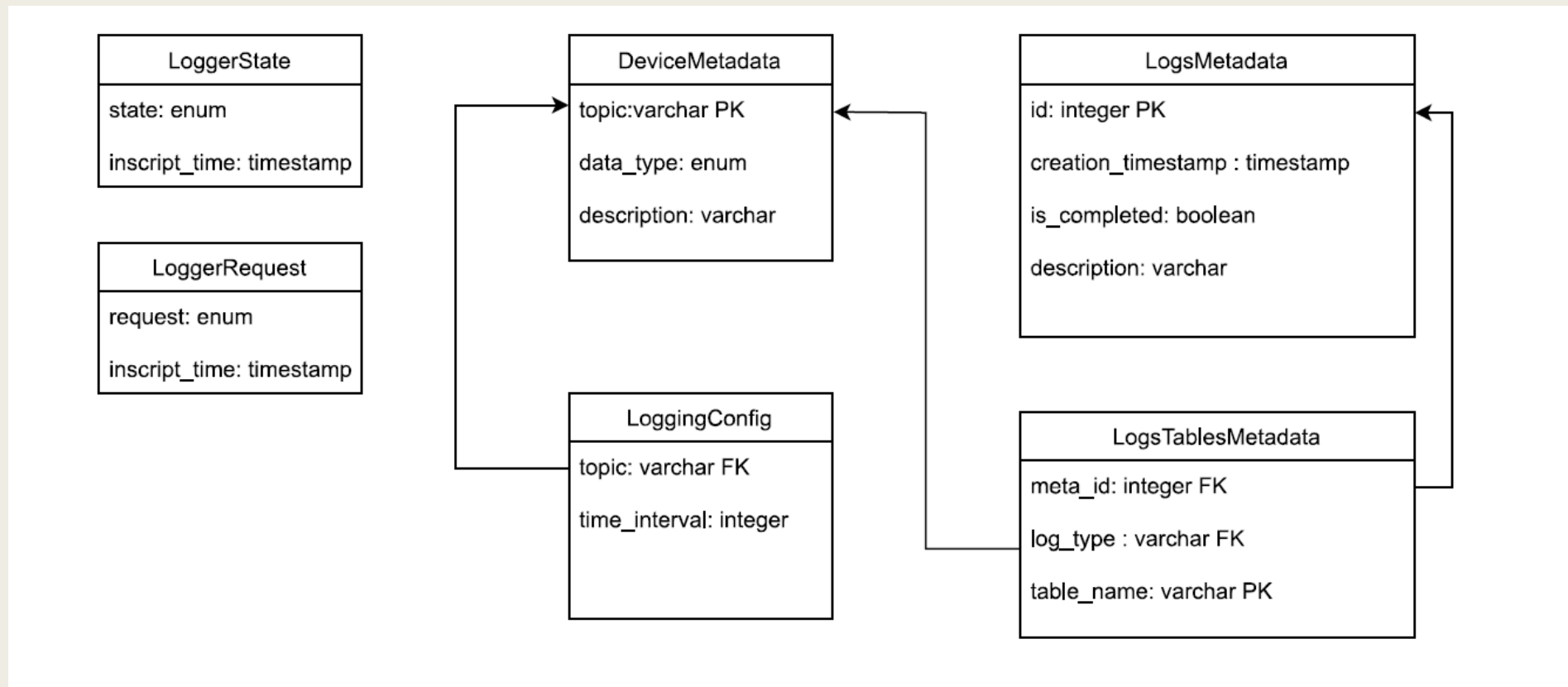
ROS – przykładowa struktura procesów



Architektura modułu – aplikacja webowa

- Aplikacja bazuje na frameworku Flask
- Implementacja aplikacji webowej we Flasku opiera się o dekoratory (element składniowy Pythona) poprzedzające funkcje obsługujące podstrony
- W dekoratorze zawiera się informacje o adresie oraz jakie metody HTTP są dozwolone (np. Post, Get)
- W omawianej aplikacji zastosowano głównie podejście proceduralne, w niewielkim stopniu obiektowe
- Główna klasa wykorzystywana w implementacji to LogsDbConnector – do obsługi zapytań bazodanowych

Podstawowa struktura bazy danych



Podsumowanie

- Zaprojektowano i zaimplementowano moduł zgodnie z założeniami funkcjonalnymi i нефункциональными
- Moduł jest otwarty na rozszerzenia, np. dodanie nowych typów logowania czy analizę danych
- Moduł będzie rozwijany, aby ostatecznie stać się częścią systemu Łazika