

Robotic Operating System Podstawy

Marek Piechula

Pytech summit

08-12-22

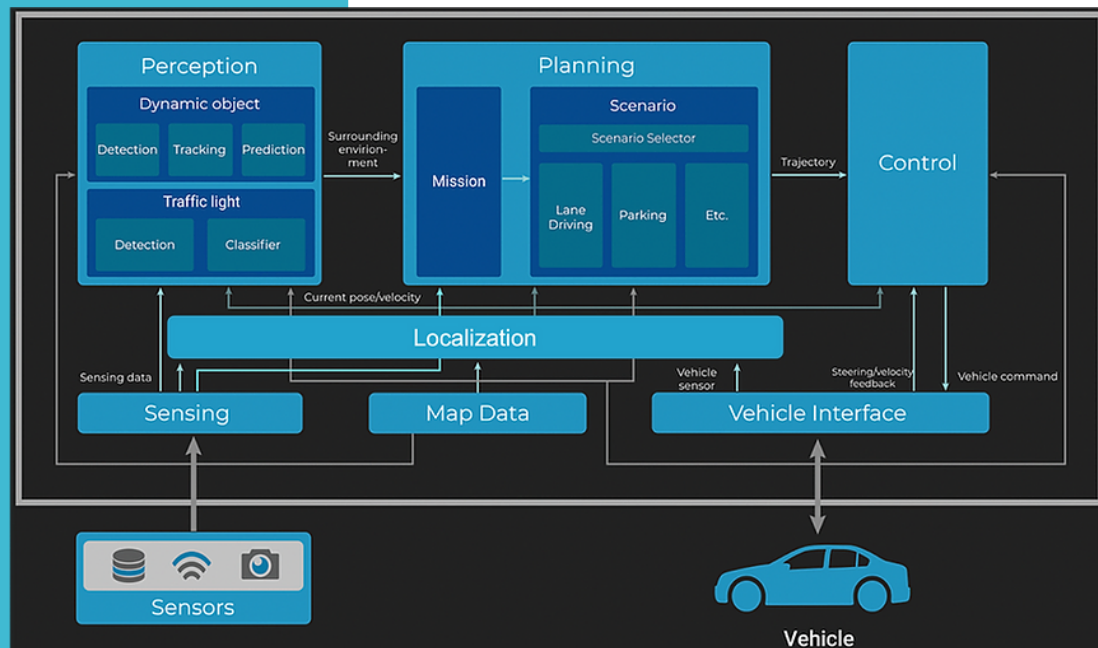
Agenda

1. Kim jestem?
2. Co to jest ROS
3. Dlaczego ROS
4. Z czego się składa ROS
 1. Node
 2. Topic
 3. Services
 4. Parameters
 5. TF
5. Final Demo
6. Q/A

Kim jestem?



- 10 lat doświadczenia z IT
 - głównie python oraz web
 - od 2 lat C++ oraz automatyczne pojazdy
- Moje doświadczenie z ROS:
- Pracuję w firmie Blee
 - Współpracuję z Autonomous Systems
 - Znam projekt Autoware.Auto – opensourcowy projekt na ROS2 dla samochodów autonomicznych



Co to jest ROS?

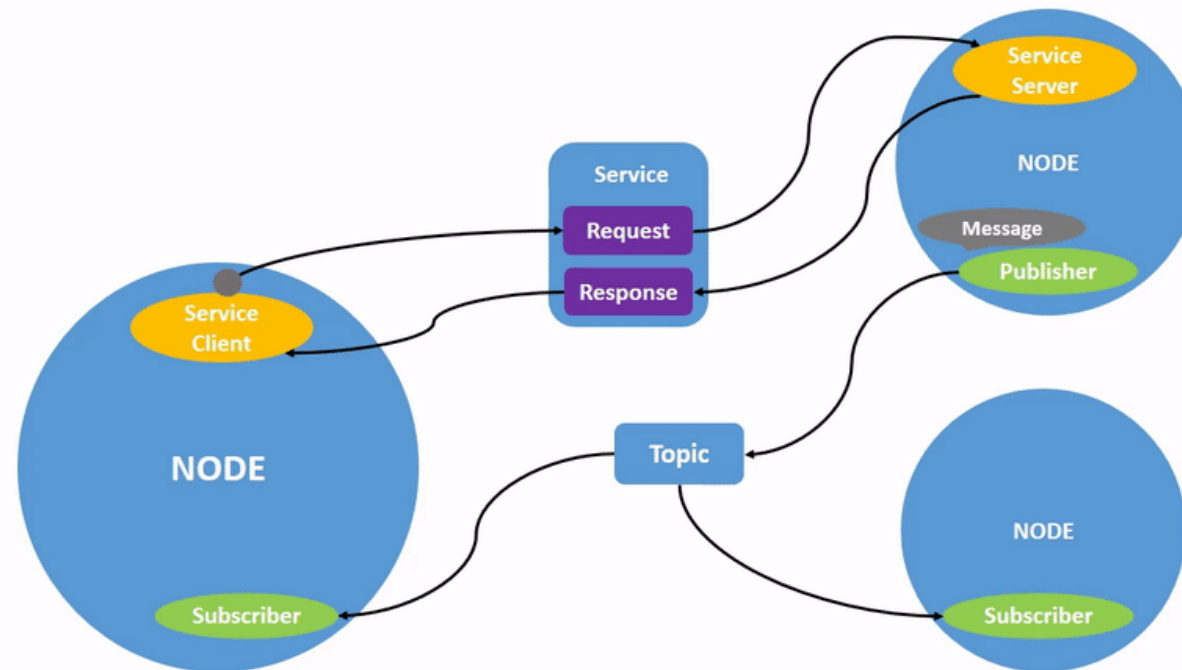
- **ROS** = **R**obot **O**perating **S**ystem
- Jest to "framework" z zestawem bibliotek oraz narzędzi stosowanych w robotyce
- W praktyce ROS jest jak świnka morska – ani Robot, ani Operating System

Dlaczego ROS?

- Od ponad 10 lat posiada spore community oraz jest wciąż aktywnie rozwijany
- Ma sporą liczbę narzędzi oraz sterowników do obsługi sensorów
- Wspiera C++ oraz Python (<3)
- Open Source <3
- Firmy wspierające ROS: Intel, Bosch, Microsoft, Samsung I wiele innych
- Interfejs pozwala testować swoje oprogramowanie w symulatorach (np. Carla) praktycznie jeden do jednego

Node

- Jest to niezależny moduł ("cegiełka") w systemie ROS który odbiera oraz przesyła do innych węzłów dane.
- Przykładem węzła mogą być sensory jak kamery, lidary; controller sterujący szybkością pojazdu czy planowanie trasy



Topic

- Jest to "adres" gdzie węzeł może odbierać dane lub przesyłać
- Np. Węzeł GNSS wysyła lokalizację do topicu `"/gnss"`, który jest odbierany przez Węzeł decydujący o przyspieszeniu pojazdu
- Sam Topic jest kolejką - jeżeli dojdą nowe wiadomości do topicu, a stare nie są przetworzone, to wiadomość zostanie wrzucona do kolejki przez sam system ROS.
- Sama obsługa kolejki jest definiowana przez QoS (Quality of Service)
- Komunikacja w ROS2 jest zapewniona przez DDS (Data Distribution Service)

Service

- Jest bardzo podobny do Topic
- Jego różnica jest taka, że po wysłaniu wiadomości, wysyłający węzeł będzie czekać na odpowiedź
- Np. Chcemy sprawdzić czy wywołanie serwisu "zamknij drzwi" doprowadziło do wysłania akcji do sieci CAN

Parameters

- Każdy węzeł może mieć dodatkowe parametry
- Najczęściej parametry są wykorzystywane do ustawienia węzła przed jego uruchomieniem (np. Maksymalna prędkość pojazdu)
- Mogą być parametry ustawiane dynamicznie

TF

- TF to rozproszony system do budowania drzewa transformacji poszczególnych połączeń w czasie
- Np. Pojazd ma transformatę względem mapy oraz koła mają transformatę względem pojazdu
- Z TF jest powiązany URDF który opisuje model "robota"
- Działa to naprawdę dobrze :-)

Final Demo

- Kod w pythonie z "AI" która zatrzymuje się przed przeszkodami
- Wykorzystuje odbieranie danych o przeszkodach, transformację względem pojazdu a przeszkód oraz wysyłanie o szybkości

Czy ROS jest gotowy dla autono mii?

- Myślę że **NIE**
- W tej chwili jest rozjazd pomiędzy starszą wersją ROS1 a nowszą ROS2 (nie wszystkie biblioteki zostały przeportowane w 100%)
- Nie ma jeszcze obsługi realtime
- Pomimo dużego community, liczby tutoriali i dokumentacji, czasami jest bardzo trudno znaleźć do poszczególnych bibliotek dokumentacji i wymaga to szukania w kodzie źródłowym
- Nie ma dobrego środowiska do testów jednostkowych

Sumarium

- Węzły to niezależne komponenty które są komunikują z innymi węzłami
- Węzły mogą wysyłać wiadomości do topiców oraz odbierać
- Węzły też posiadają serwisy do których wysyła się żądanie (request) I jest wysyłana odpowiedź (response)
- Węzły posiadają też parametry które mogą być ustalane dynamicznie
- TF to rozproszony system połączeń w czasie

Linki

- <http://wiki.ros.org/rviz>
- <https://www.plotjuggler.io>
- ???

Q/A