



# Politechnika Wrocławska



Katedra Cybernetyki i Robotyki

## Struktura ramowa ROS

Mariusz Janiak  
p. 331 C-3, 71 320 26 44

© 2015 Mariusz Janiak  
All Rights Reserved



# Zawartość wykład

1 Wprowadzenie

2 Podstawy



# Wprowadzenie

## ROS w kilku słowach

- *Robot Operating System* (a nie system operacyjny)
- Narzędzia i biblioteki do rozwijania aplikacji robtycznych
- Abstrakcja sprzętu, sterowniki urządzeń, narzędzia do debugowania, przekazywanie wiadomości, zarządzanie paczkami, itd.
- Licencja wolnego oprogramowania BSD

## Strona domowa

- [www.ros.org](http://www.ros.org)



# Wprowadzenie

## Historia

- Pracę rozpoczęto w 2007 w Stanford Artificial Intelligence Laboratory następnie kontynuowano w Willow Garage
- Od 2013 jest zarządzany przez OSRF (Open Source Robotics Foundation)
- Dotychczas jedenaście głównych wydań, ostatnie *Lunar Loggerhead*
- Nadchodzi ROS2 (czas rzeczywisty, DDS, urządzenia wbudowane równorzędne komponentom programowym)



# Wprowadzenie

## Cele

- Model programistyczny zorientowany na agenty
- Peer to peer
- Oprogramowania rozwijanie w oparciu o zestaw narzędzi
- Wsparcie dla wielu języków programowania (C++/Python/Java)
- Lekki – działa na krańcach modułów
- Darmowy, licencja wolnego oprogramowania
- Odpowiedni do prowadzenia na dużą skalę badań i zastosowania w przemyśle



# Wprowadzenie

## Zestaw narzędzi

- Tworzenie paczek ROS
- Budowanie węzłów ROS
- Uruchamianie węzłów ROS
- Podgląd topologii sieci
- Monitorowanie ruchu w sieci
- Nie pojedynczy, monolityczny program, w zamian wiele małych procesów



# Wprowadzenie

## Wspierane roboty

- Manipulatory mobilne
- Platformy mobilne
- Manipulatory
- Autonomiczne samochody
- Roboty społeczne
- Humanoidy
- UAVs
- AUVs
- UUVs
- Inne





# Podstawy

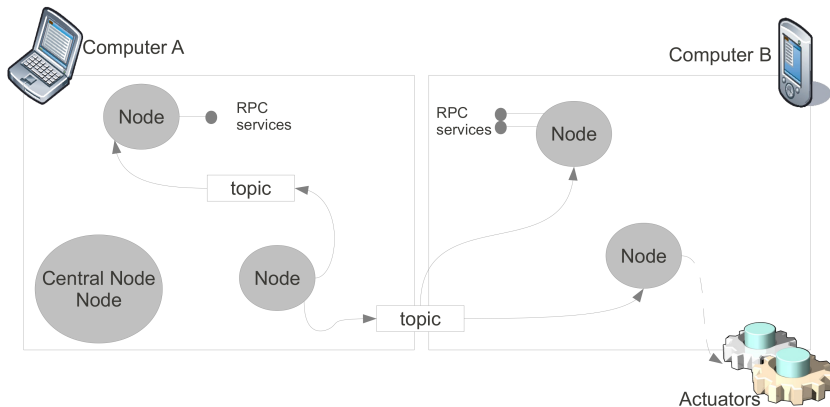
## Architektura rozproszona

- Architektura hybrydowa P2P
- Rozproszone komponenty
- Kilka mechanizmów komunikacji pomiędzy węzłami (bazujące na przekazywaniu wiadomości)
- Koncentruje się na mechanizmach komunikacji pomiędzy węzłami
- Dowolność w budowie wewnętrznej węzłów



# Podstawy

## Architektura rozproszona





# Podstawy

## Rdzeń ROS (minimalny, działający zestaw programów)

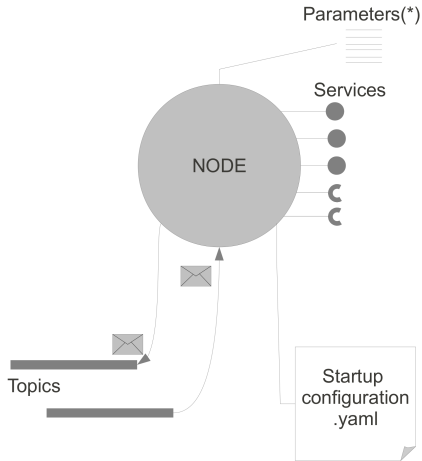
- ROS Master
  - Scentralizowany serwer XML-RPC
  - Katalog nadawców, odbiorców i usług
  - Pośredniczy w nawiązywaniu połączeń
- Serwer parametrów
  - Scentralizowane repozytorium parametrów
  - Umożliwia dostęp do parametrów wszystkim węzłom
  - typy danych XML-RPC
  - Brak automatycznej aktualizuje parametrów wewnątrz węzłów
- roscout
  - Sieciowy stdout dla wiadomości czytelnych dla człowieka
  - Subskrybuje temat /out
  - Składowe wyjście na dysku



# Podstawy

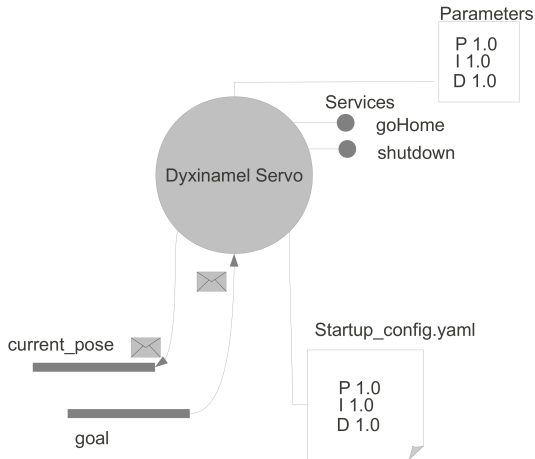
## Węzeł (*Node*)

- Minimalny element systemu
- Własny przepływ sterowania
- Pojedynczy program wykonywalny
- Dowolny, wspierany język
- Mechanizmy komunikacji: tematy (*topic*), usługi (*service*), parametry (*parameters*)
- Konfigurowalny (pliki YAML)





# Podstawy

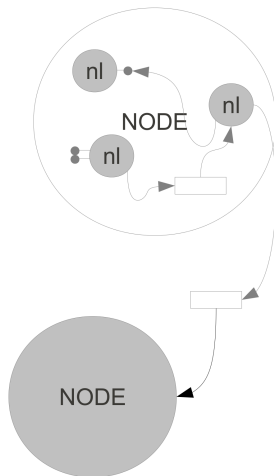




# Podstawy

## Nodelet

- Wątek
- Kompatybilny w węzłami mechanizm komunikacji
- Komunikacja bez kopiowania pomiędzy nodletami
- Dzielą wspólna maszynę
- Wyłącznie C++

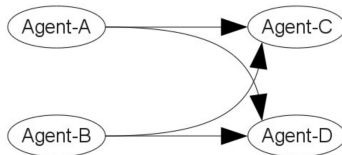
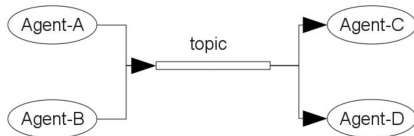




# Podstawy

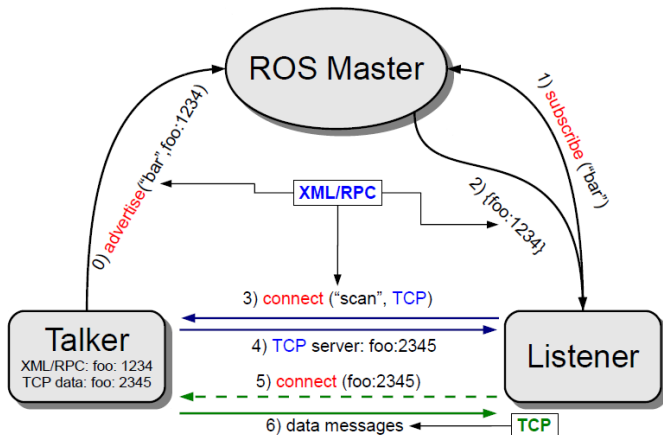
## Temat ( *Topic* )

- Model Publikuj/Subskrybuj
- Wielu do wielu
- Zorientowany na przekazywanie wiadomości
- Silna typizacja (plik .msg)
- Warstwa transportowa
  - TCP, UDP, pamięć dzielona
  - roserial, ethercat





# Podstawy





# Podstawy

## Wiadomość (*Message*)

- Struktura danych
- Opis interfejsu zawarty w pliku
- Generacja kodu serializacji: C++, Python, ...
- Definiowane statycznie
- Standardowe paczki wiadomości
  - geometry\_msgs
  - sensor\_msgs
  - navigation\_msgs
  - ...

my\_package/msg/example.msg

```
string field1
int8 field2
bool field3
other_pkg_msg/custom_msg field4
```



.h

.py

.java



# Podstawy

## Usługa (Service)

- RPC
- Jeden do jednego (żądanie-odpowiedź)
- Silna typizacja (plik .srv)
- Generacja kodu serializacji: C++, Python, ...
- Definiowane statycznie
- Warstwa transportowa: TCP/IP lub UDP

my\_package/srv/example.srv

```
string request_field1
int8 request_field2
---
string response_field1
other_pkg_msg/custom_msg response_field2
```



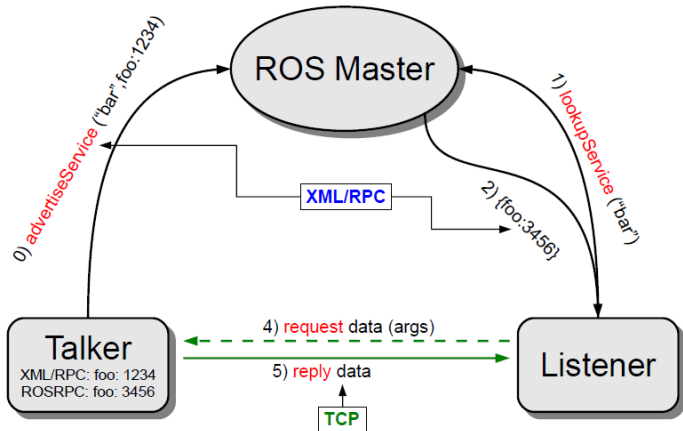
.h

.py

.java



# Podstawy





# Podstawy

## Paczka ROS

- Niepodzielna jednostka podlegająca budowie
- Może zawierać cokolwiek
  - węzły
  - wiadomości
  - narzędzia
  - skrypty
  - pliki *launch*
- Minimalny zestaw
  - `package_name/CMakeLists.txt` – plik cmake paczki
  - `package_name/package.xml` – plik catkin manifest



# Koniec

Dziękuję za uwagę.