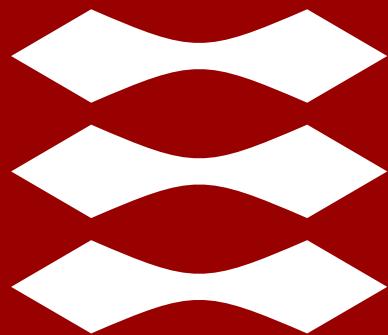


DTU



Slides: Evangelos Boukas
Lecturer: Lazaros Nalpantidis

Middleware for Autonomous Systems

Outline

- Introduction
- Middleware Description
 - HAL
 - Most Important Middleware
 - Main Taxonomy
- Intro to ROS!
- Summary

- Introduction
- Middleware Description
 - HAL
 - Most Important Middleware
 - Main Taxonomy
- Intro to ROS!
- Summary

Introduction

- What is a "middleware"?

Introduction

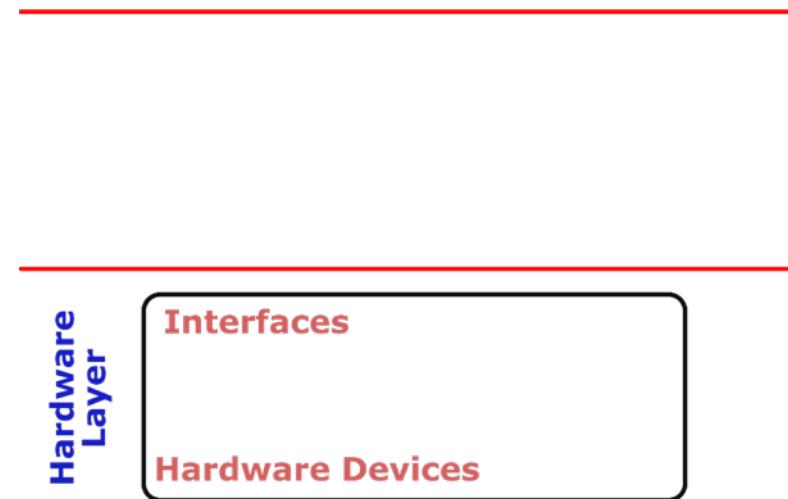
- What is a "middleware"?
- Why do we need it?

- Introduction
- Middleware Description
 - HAL
 - Most Important Middleware
 - Main Taxonomy
- Intro to ROS!
- Summary

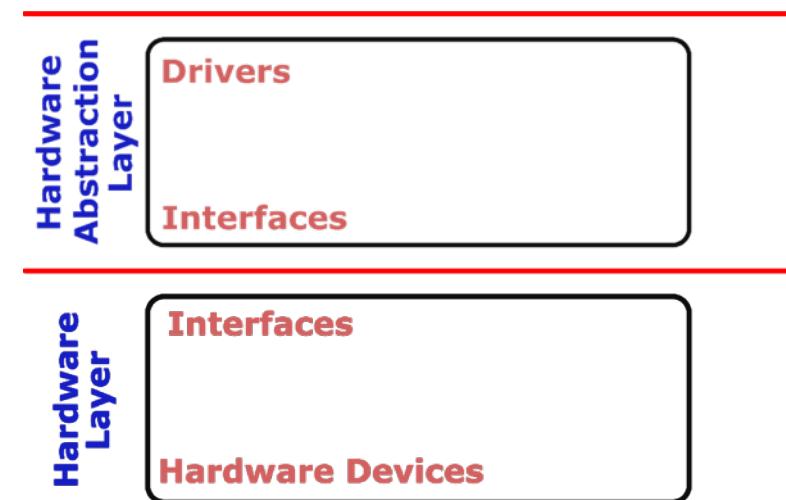
- Introduction
- Middleware Description
 - HAL
 - Most Important Middleware
 - Main Taxonomy
- Intro to ROS!
- Summary

Hardware Abstraction Layers #1

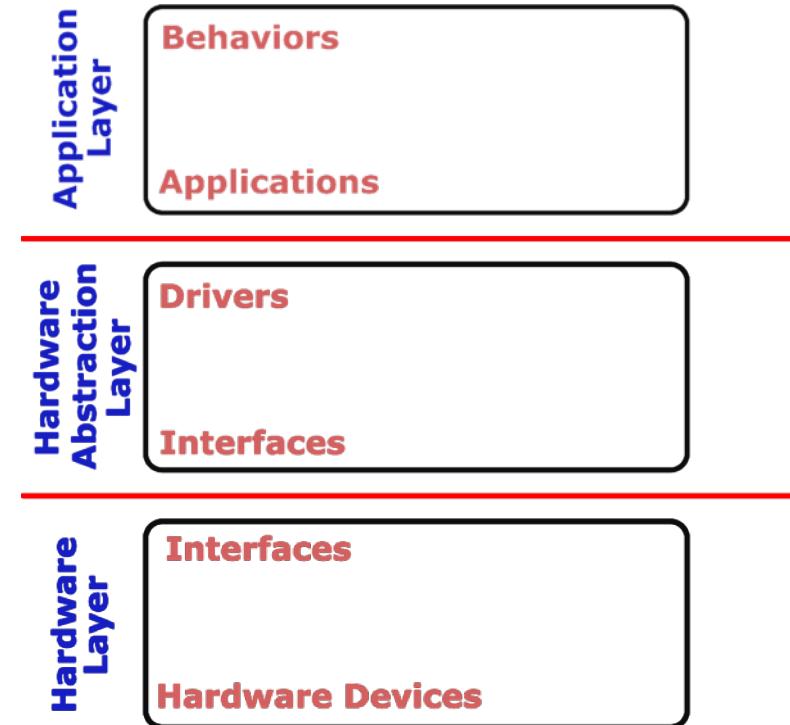
Hardware Abstraction Layers #1



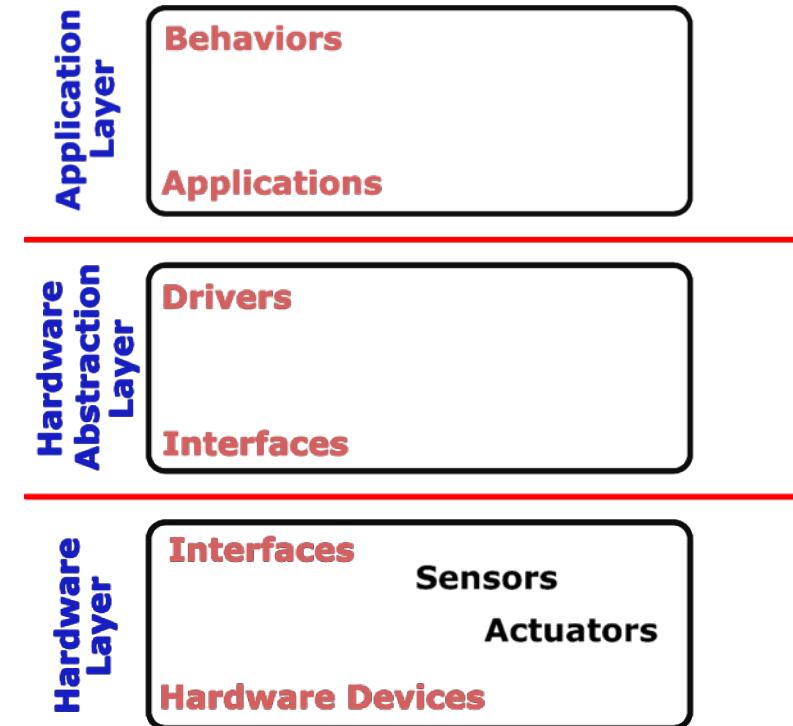
Hardware Abstraction Layers #1



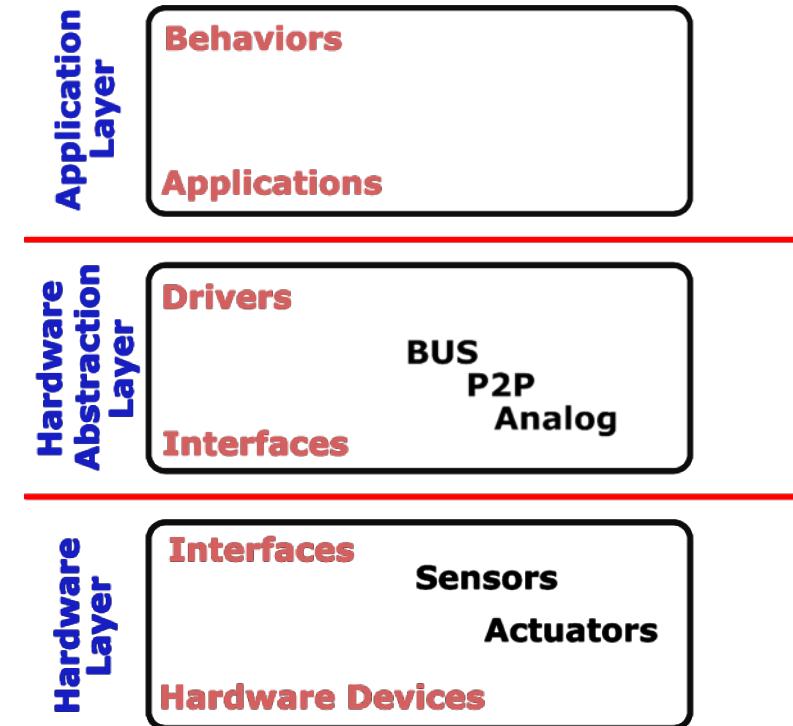
Hardware Abstraction Layers #1



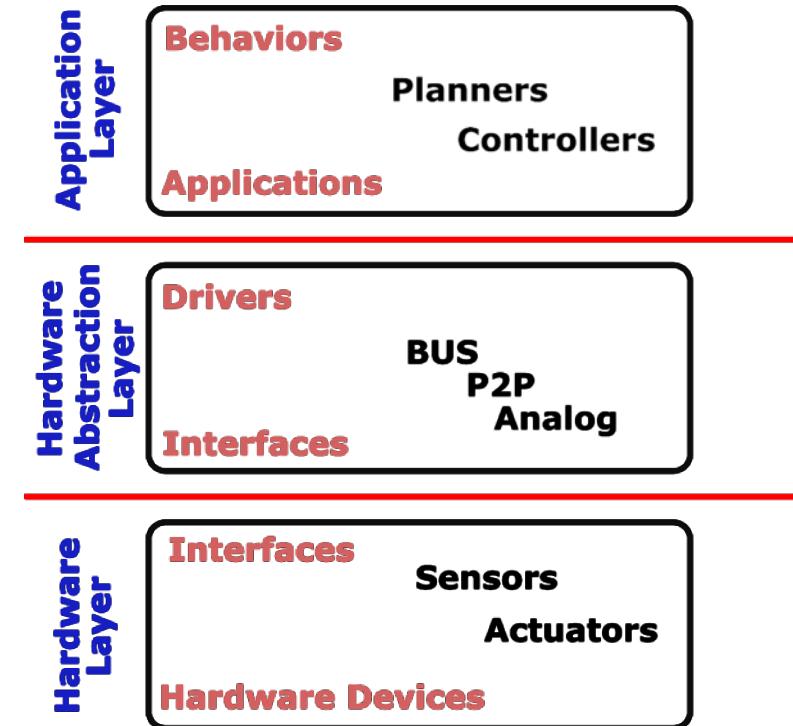
Hardware Abstraction Layers #1



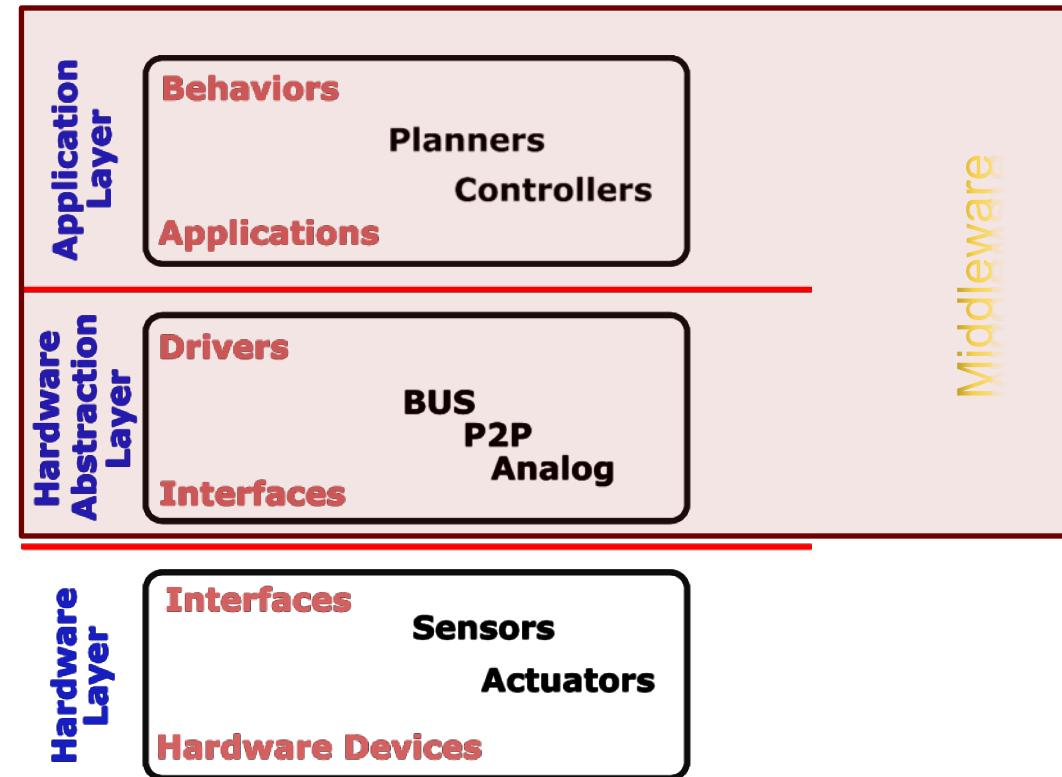
Hardware Abstraction Layers #1



Hardware Abstraction Layers #1



Hardware Abstraction Layers #1



Hardware Abstraction Layers #2

- Let's add a little detail..

Hardware Abstraction Layers #2

- Let's add a little detail..



Hardware

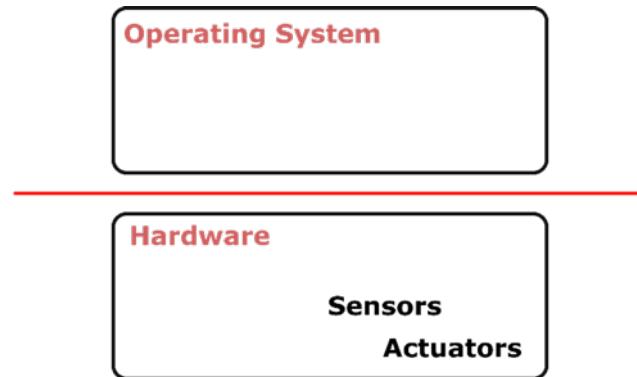
Hardware Abstraction Layers #2

- Let's add a little detail..



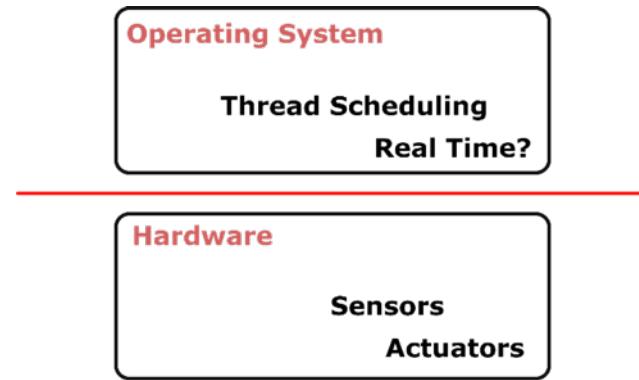
Hardware Abstraction Layers #2

- Let's add a little detail..



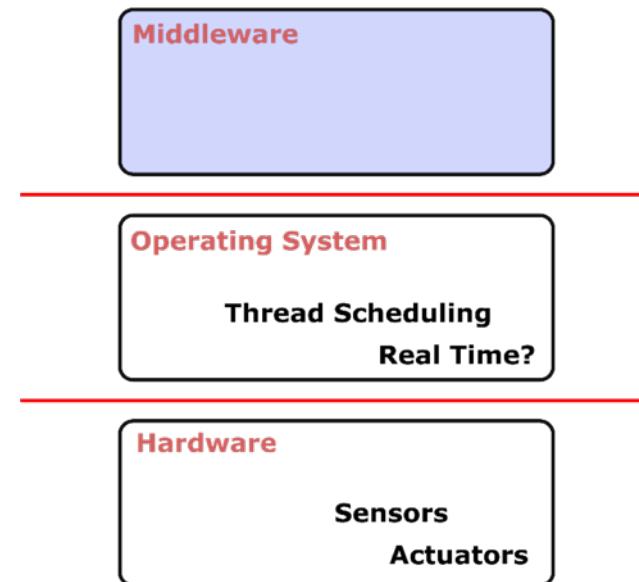
Hardware Abstraction Layers #2

- Let's add a little detail..



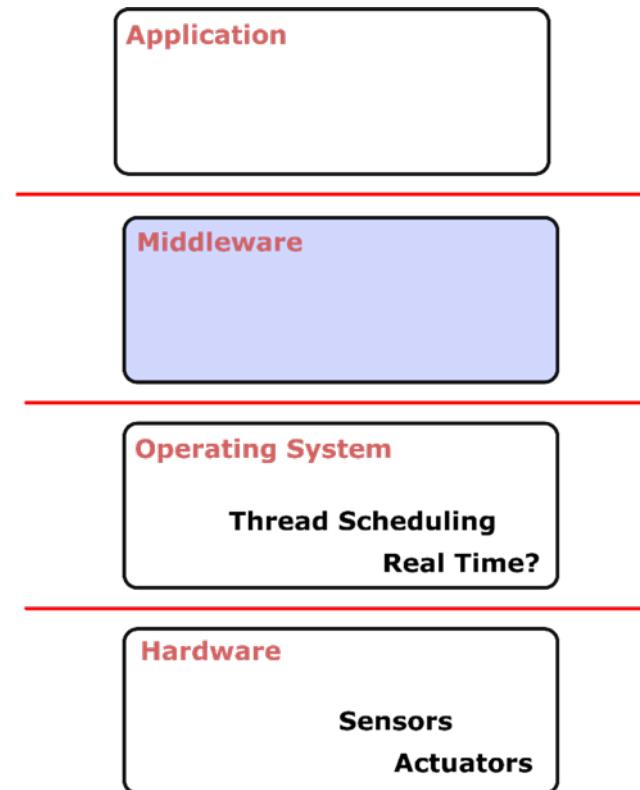
Hardware Abstraction Layers #2

- Let's add a little detail..



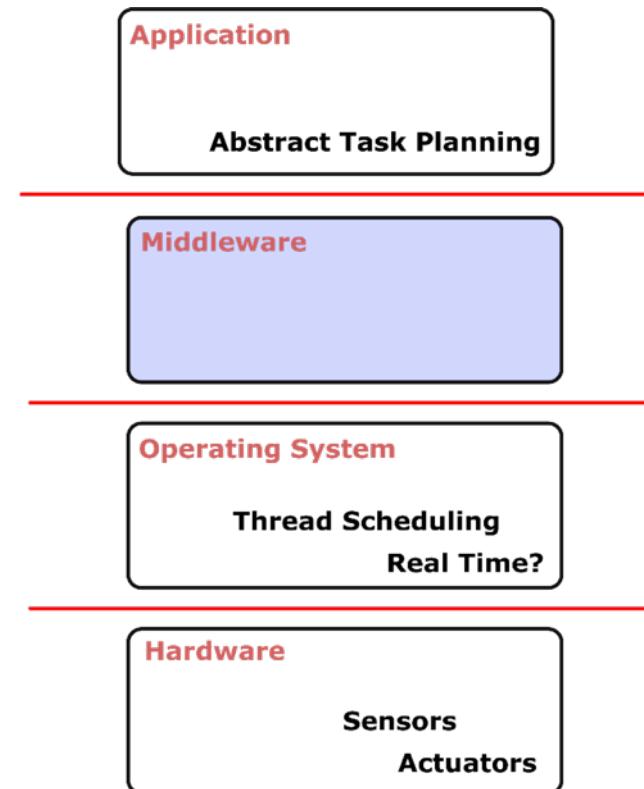
Hardware Abstraction Layers #2

- Let's add a little detail..



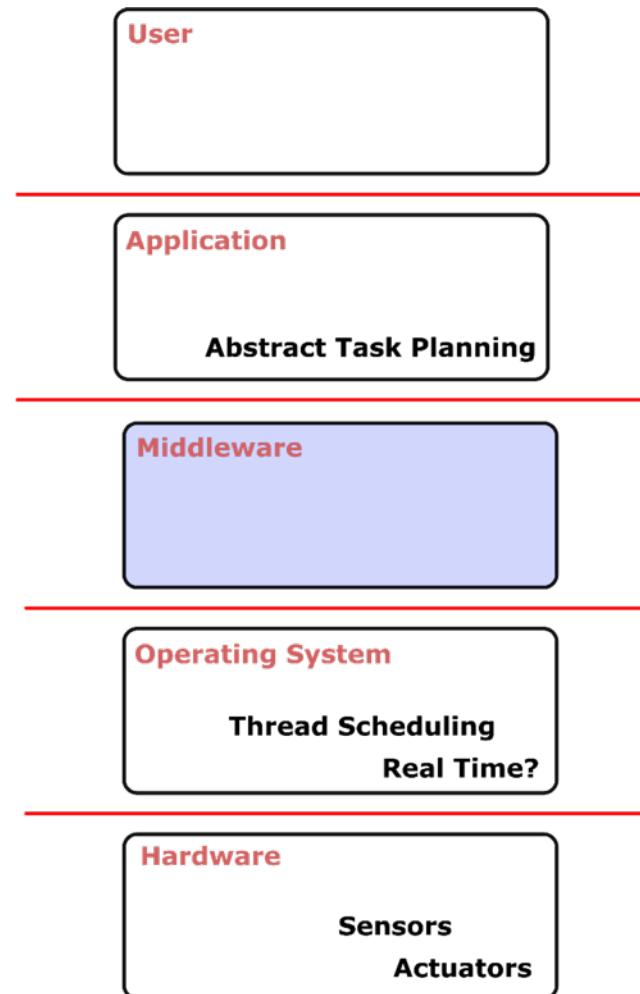
Hardware Abstraction Layers #2

- Let's add a little detail..



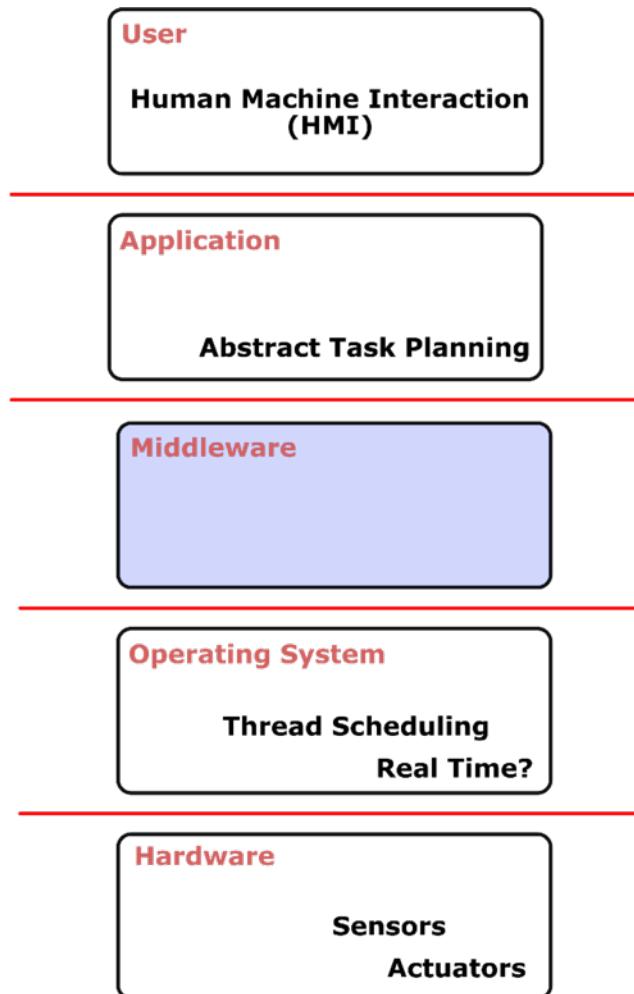
Hardware Abstraction Layers #2

- Let's add a little detail..



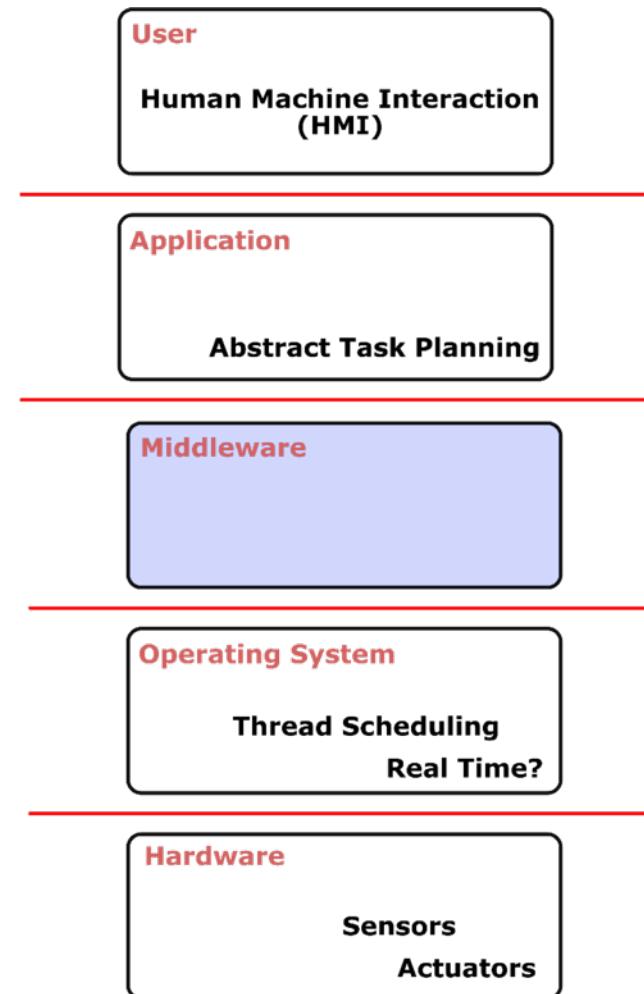
Hardware Abstraction Layers #2

- Let's add a little detail..



Hardware Abstraction Layers #2

- Let's add a little detail..
- A class of technologies in order to handle the complexity of distributed systems



What are some Middleware?

What are some Middleware?



USC University of
Southern California

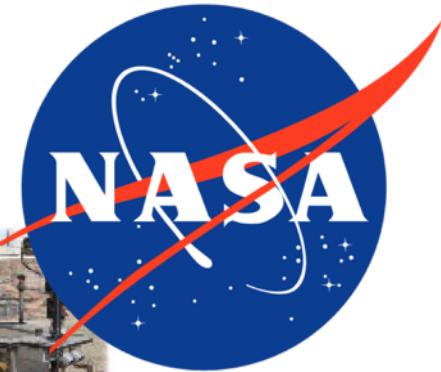
Player Stage

What are some Middleware?



USC University of
Southern California

Player Stage



What are some Middleware?



USC University of
Southern California

Player Stage



Microsoft®
Robotics
Developer
Studio
Microsoft

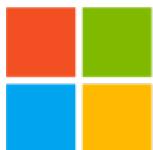


What are some Middleware?



USC University of
Southern California

Player Stage



Microsoft®
Robotics
Developer
Studio
Microsoft



ROS Willow Garage



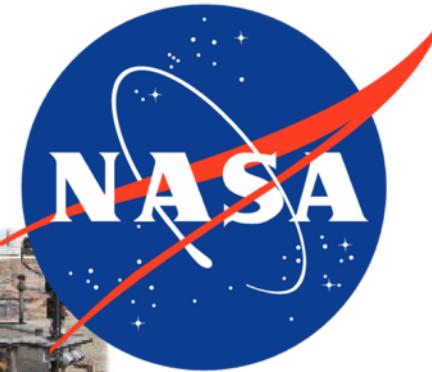
What are some Middleware?



USC University of
Southern California

Player Stage

Microsoft®
Robotics
Developer
Studio
Microsoft



ROS Willow Garage



KATHOLIEKE UNIVERSITEIT
LEUVEN

What are some Middleware?

- Orocос
- Pyro
- Player
- Orca
- Miro
- OpenRTMaist
- ASEBA
- MARIE
- RSCA
- MRDS
- OPROS
- CLARAty
- ROS
- SmartSoft
- ERSP
- Webots
- RoboFrame

What's UP with this?

What's UP with this?

- Is there any reason there're so many?

What's UP with this?

- Is there any reason there're so many?
- Different Scope

What's UP with this?

- Is there any reason there're so many?
- Different Scope
- Different Functional Architecture

What's UP with this?

- Is there any reason there're so many?
- Different Scope
- Different Functional Architecture
- Different Communication Architectures

Let's see the most important ones!

- CLARAty

Let's see the most important ones!

- CLARAty
 - NASA Jet Propulsion Laboratory

Let's see the most important ones!

- CLARAty
 - NASA Jet Propulsion Laboratory
 - Functional Layer:
 - Navigation, Mapping
 - Terrain evaluation, path planning
 - Estimation, simulation

Let's see the most important ones!

- CLARAty
 - NASA Jet Propulsion Laboratory
 - Functional Layer:
 - Navigation, Mapping
 - Terrain evaluation, path planning
 - Estimation, simulation
 - Decision Layer:
 - General Planners
 - Schedulers, Databases

Let's see the most important ones!

- CLARAty
 - NASA Jet Propulsion Laboratory
 - Functional Layer:
 - Navigation, Mapping
 - Terrain evaluation, path planning
 - Estimation, simulation
 - Decision Layer:
 - General Planners
 - Schedulers, Databases
 - Client ↔ Server Scheme

Let's see the most important ones!

- CLARAty
- Orocos

Let's see the most important ones!

- CLARAty
- Orocosp
 - RealTime

Let's see the most important ones!

- CLARAty
- Orocос
 - RealTime
 - Orocос Components Library (OCL)
 - Orocос Kinematics and Dynamics Library (KDL)
 - Orocос Bayesian Filtering Library (BFL)

Let's see the most important ones!

- CLARAty
- Orocос
 - RealTime
 - Orocос Components Library (OCL)
 - Orocос Kinematics and Dynamics Library (KDL)
 - Orocос Bayesian Filtering Library (BFL)
 - OMG's CORBA

Let's see the most important ones!

- CLARAty
- Orocос
- Player

Let's see the most important ones!

- CLARAty
- Orocosp
- Player



Kind of Opposite

Let's see the most important ones!

- CLARAty
- Orocос
- Player
 - Shared Libraries among devices



Kind of Opposite

Let's see the most important ones!

- CLARAty
- Orocosp
- Player
 - Shared Libraries among devices
 - Player Core
 - Drivers, Libraries
 - Configuration Parsing



Kind of Opposite

Let's see the most important ones!

- CLARAty
- Orocос
- Player
 - Shared Libraries among devices
 - Player Core
 - Drivers, Libraries
 - Configuration Parsing
 - Transport Layer
 - Independent of Drivers
 - TCP communication using web sockets



Kind of Opposite

Let's see the most important ones!

- CLARAty
- Orocос
- Player
- ROS

Let's see the most important ones!

- CLARAty
- Orocос
- Player
- ROS



Logical Extension

Let's see the most important ones!

- CLARAty
- Orocosp
- Player
- ROS
 - Master



Logical Extension

Let's see the most important ones!

- CLARAty
 - Orocос
 - Player
 - ROS
- Master
– Nodes

Logical Extension

Let's see the most important ones!

- CLARAty
 - Orocос
 - Player
 - ROS
- Logical Extension
- Master
 - Nodes
 - Topics
 - Messages

Let's see the most important ones!

- CLARAty
- Orocос
- Player
- ROS
 - Master
 - Nodes
 - Topics
 - Messages
 - Publish/Subscribe Scheme



Logical Extension

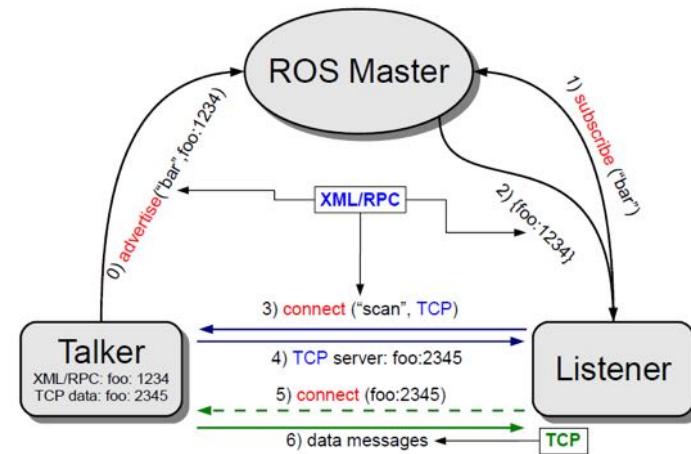
Main Differences to remember

Main Differences to remember

- Peer to Peer **Vs** “Open” Publish Subscribe Communication

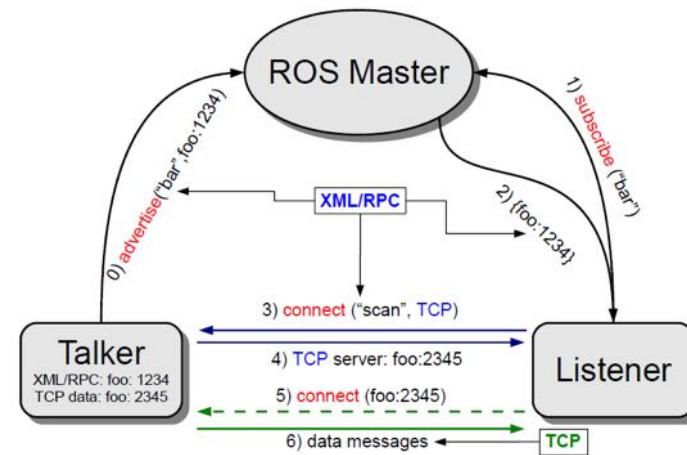
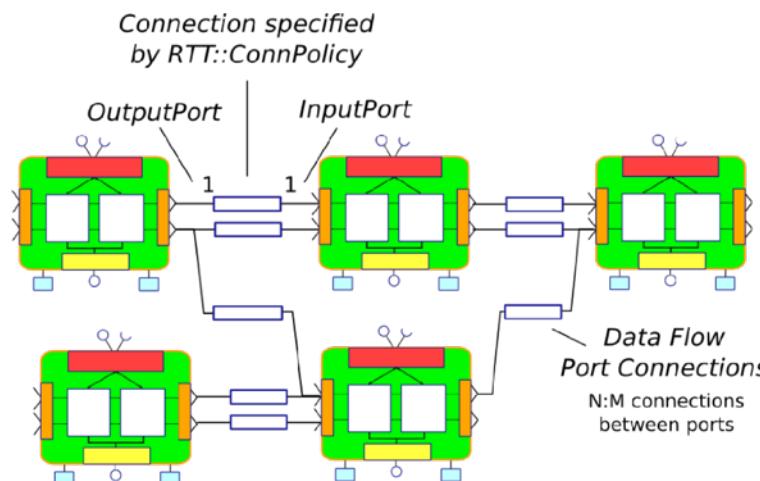
Main Differences to remember

- Peer to Peer Vs “Open” Publish Subscribe Communication



Main Differences to remember

- Peer to Peer Vs “Open” Publish Subscribe Communication

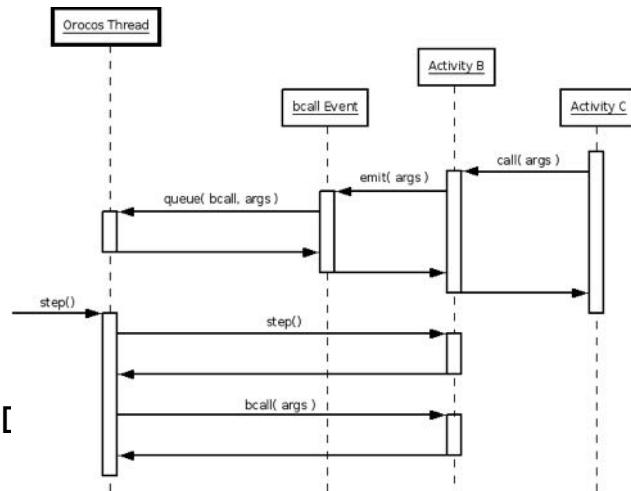


Main Differences to remember

- Peer to Peer **Vs** “Open” Publish Subscribe Communication
- State Driven **Vs** Message Driven

Main Differences to remember

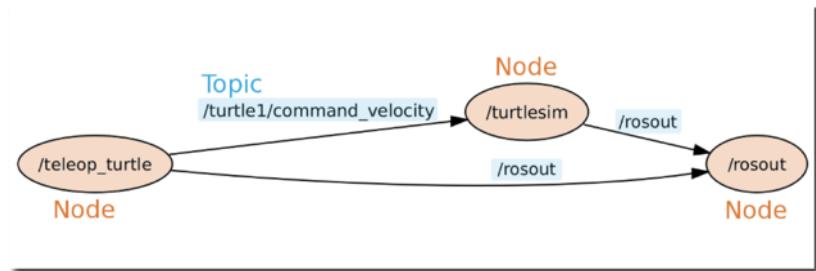
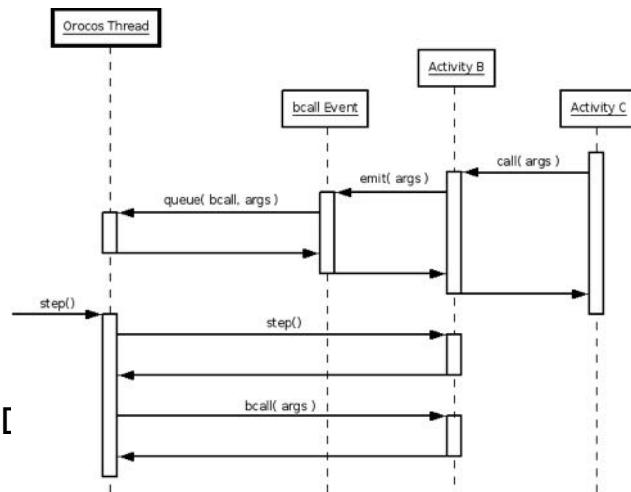
- Peer to Peer Vs “Open” Publish Subscribe Communication



- State [

Main Differences to remember

- Peer to Peer Vs “Open” Publish Subscribe Communication



- State [

Outline

- Introduction
- Middleware Description
 - HAL
 - Most Important Middleware
 - Main Taxonomy
- Intro to ROS!
- Summary

- Introduction
- Middleware Description
 - HAL
 - Most Important Middleware
 - Main Taxonomy
- Intro to ROS!
- Summary

Intro to ROS!

- What is ROS:

Intro to ROS!

- What is ROS:
 - Distributed Computation

Intro to ROS!

- What is ROS:
 - Distributed Computation
 - Software Reuse

Intro to ROS!

- What is ROS:
 - Distributed Computation
 - Software Reuse
 - Rapid Testing

Intro to ROS!

- What is ROS:
 - Distributed Computation
 - Software Reuse
 - Rapid Testing
- What it isn't:

Intro to ROS!

- What is ROS:
 - Distributed Computation
 - Software Reuse
 - Rapid Testing
- What it isn't:
 - A programming Language

Intro to ROS!

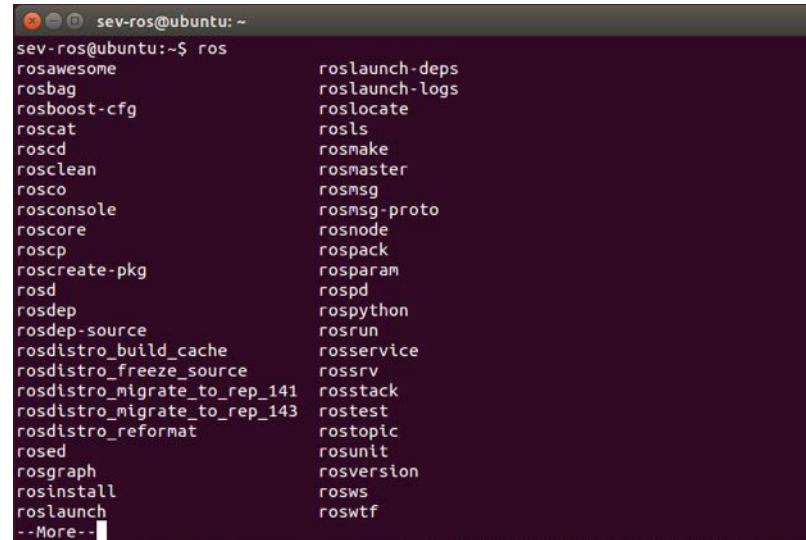
- What is ROS:
 - Distributed Computation
 - Software Reuse
 - Rapid Testing
- What it isn't:
 - A programming Language
 - A library

Intro to ROS!

- What is ROS:
 - Distributed Computation
 - Software Reuse
 - Rapid Testing
- What it isn't:
 - A programming Language
 - A library
 - An IDE

Intro to ROS - Installing

- Add ‘correct’ ROS package
- apt-get install
- rosdep
- Setup environment (terminal)



A terminal window titled "sev-ros@ubuntu: ~" displaying a list of ROS command-line tools. The tools are listed in two columns:

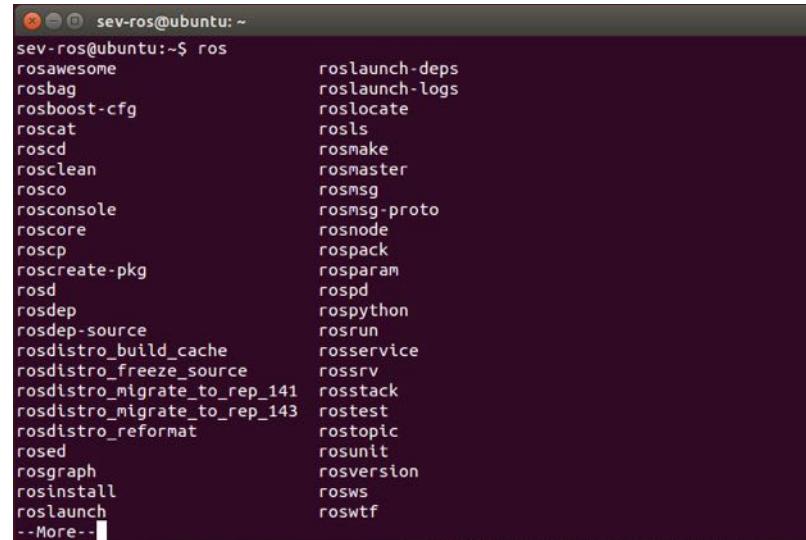
rosawsome	roslaunch-deps
rosbag	roslaunch-logs
rosboost-cfg	roslocate
roschat	rosls
roscd	rosmake
rosclean	rosmaster
rosco	rosmsg
rosconsole	rosmsg-proto
roscore	rosnode
rosclp	rospack
roscreate-pkg	rosparam
rosd	rospd
rosdep	rospython
rosdep-source	rosrun
rosdistro_build_cache	rosservice
rosdistro_freeze_source	rossrv
rosdistro_migrate_to_rep_141	rosstack
rosdistro_migrate_to_rep_143	rostest
rosdistro_reformat	rostopic
rosed	rosunit
rosgraph	rosversion
rosinstall	rosws
roslaunch	roswtf

--More-- █

Intro to ROS - Installing

- Add ‘correct’ ROS package
- apt-get install
- rosdep
- Setup environment (terminal)

Here is a guide



A terminal window titled "sev-ros@ubuntu: ~" displaying a list of ROS command-line tools. The tools are listed in two columns:

roswesome	roslaunch-deps
rosbag	roslaunch-logs
rosboost-cfg	roslocate
roschat	rosls
roscd	rosmake
rosclean	rosmaster
rosco	rosmsg
rosconsole	rosmsg-proto
roscore	rosnode
rosclp	rospack
roscreate-pkg	rosparam
rosd	rospd
rosdep	rospython
rosdep-source	rosrun
rosdistro_build_cache	rosservice
rosdistro_freeze_source	rossrv
rosdistro_migrate_to_rep_141	rosstack
rosdistro_migrate_to_rep_143	rostest
rosdistro_reformat	rostopic
rosed	rosunit
rosgraph	rosversion
rosinstall	rosws
roslaunch	roswtf

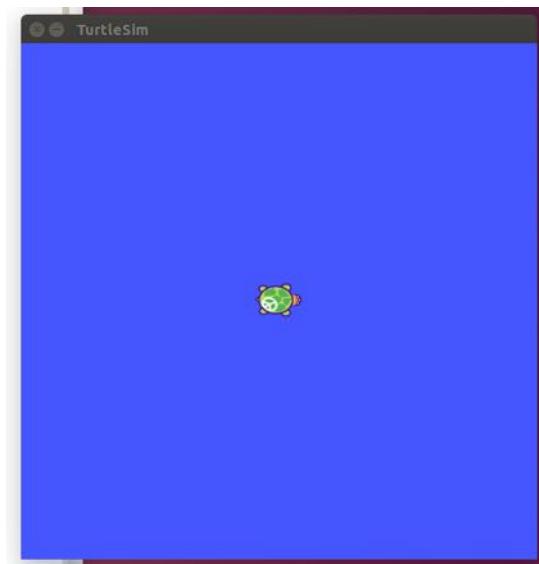
--More--

Intro to ROS: main components by example

- Master
 - “roscore”
- Let’s run something, quickly!
- “rosrun”
- Packages
 - “rospack”
- What did just happen?
- “rqt_graph”

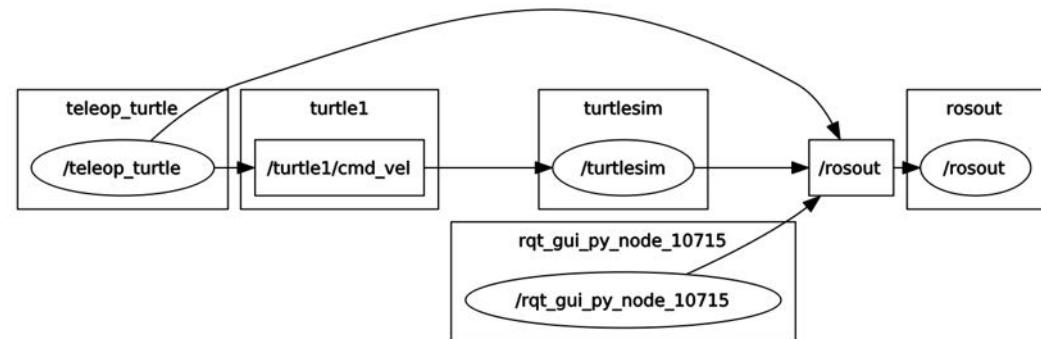
Intro to ROS: main components by example

- Master
 - “roscore”
- Let's run something, quickly!
- “rosrun”
- Packages
 - “rospack”
- What did just happen?
- “rqt_graph”



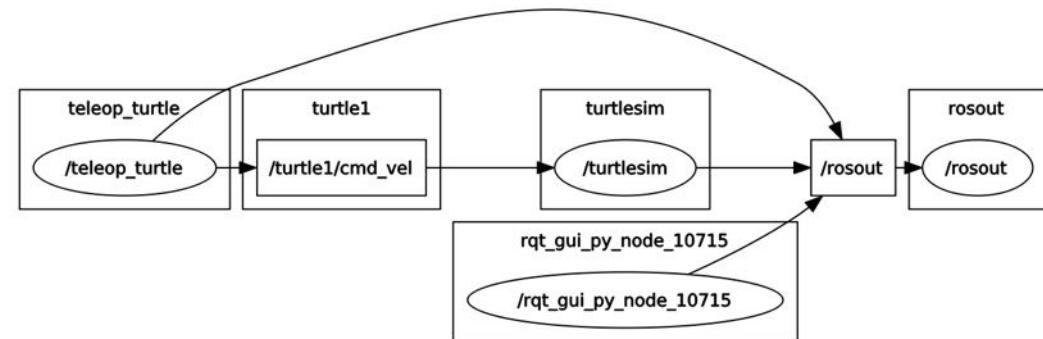
Intro to ROS: main components by example

- Master
 - “roscore”
- Let’s run something, quickly!
- “rosrun”
- Packages
 - “rospack”
- What did just happen?
- “rqt_graph”



Intro to ROS: main components by example

- Master
 - “roscore”
- Let’s run something, quickly!
- “rosrun”
- Packages
 - “rospack”
- What did just happen?
- “rqt_graph”



Intro to ROS: main components by example

- Nodes
 - “rosnode”
- Topics & Messages
 - “rostopic”
 - “rosmsg”

```
sev-ros@ubuntu:~$ roscore http://ubu... x sev-ros@ubuntu:~ x | sev-ros@ubuntu:~ x | sev-ros@ubuntu:~ x
sev-ros@ubuntu:~$ rostopic list
/rosvout
/rosvout_agg
/turtle1/cmd_vel
/turtle1/color_sensor
/turtle1/pose
sev-ros@ubuntu:~$ rostopic info /turtle1/cmd_vel
Type: geometry_msgs/Twist

Publishers:
* /teleop_turtle (http://ubuntu:35597/)

Subscribers:
* /turtlesim (http://ubuntu:34021/)

sev-ros@ubuntu:~$ rosmsg show geometry_msgs/Twist
geometry_msgs/Vector3 linear
  float64 x
  float64 y
  float64 z
geometry_msgs/Vector3 angular
  float64 x
  float64 y
  float64 z
sev-ros@ubuntu:~$ □
```

- Introduction
- Middleware Description
 - HAL
 - Most Important Middleware
 - Main Taxonomy
- Intro to ROS!
- Summary

- Introduction
- Middleware Description
 - HAL
 - Most Important Middleware
 - Main Taxonomy
- Intro to ROS!
- Summary

Summary

Summary

- We learned about the different middleware for autonomous robotic systems
- We had a first glimpse at ROS

Slides: Evangelos Boukas
Lecturer: Lazaros Nalpantidis

Middleware for Autonomous Systems