ATLAS

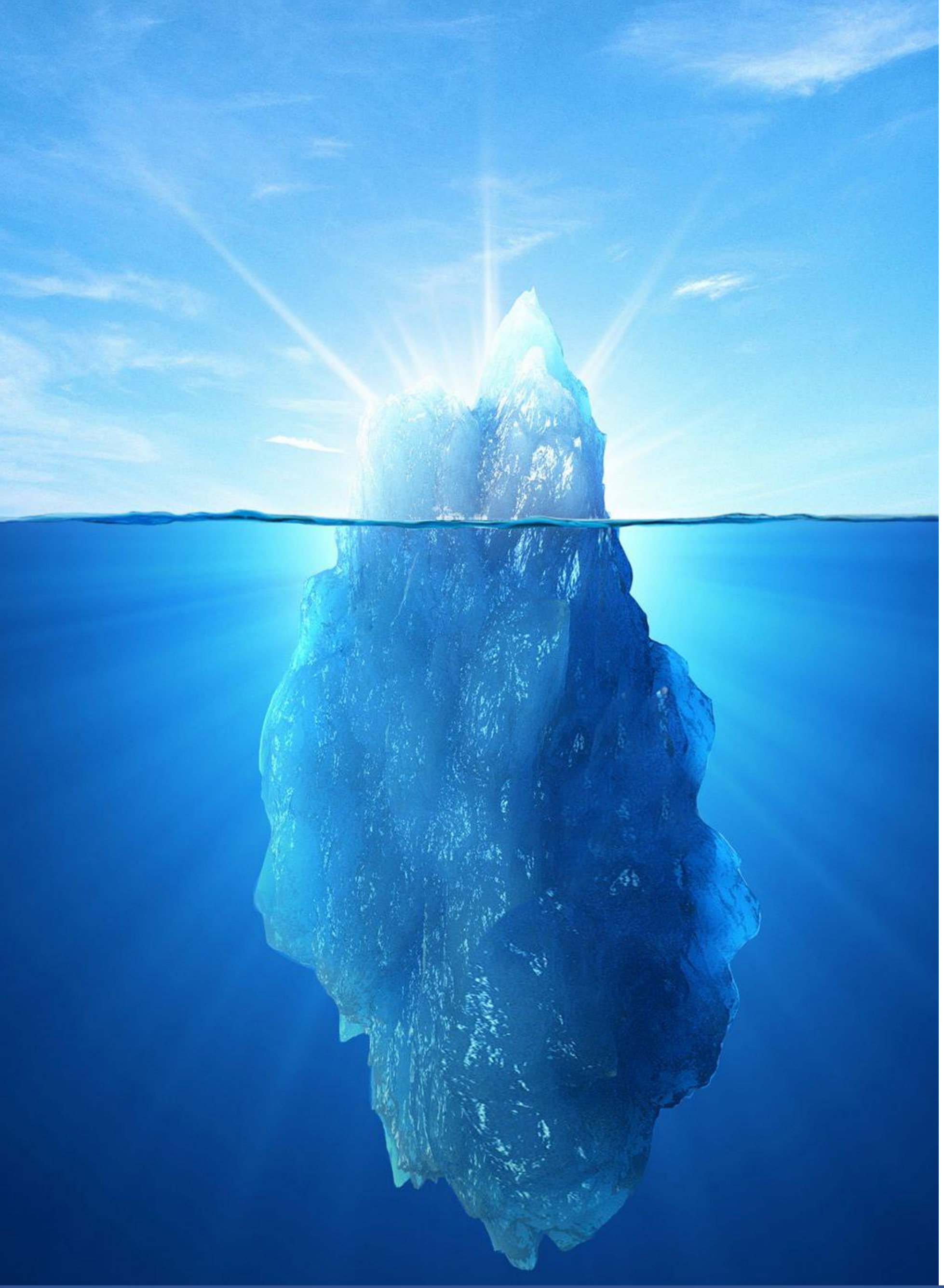
ROS on multiple workstations

Diego Dall’Alba
UNIVR - Altair Robotics Lab

Integration Meeting @ KU Leuven 20 -24 Sept 2021

UNIVERSITÀ di VERONA
Dipartimento di INFORMATICA

ALTAIR robotics lab

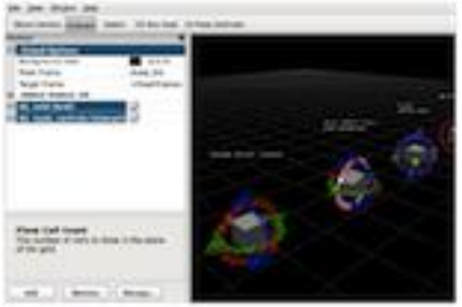


Overview

- Quick Recap from NTA3
  - ROS architecture & philosophy
  - ROS master, nodes, and topics
- Rules for setting up ROS on multiple machine
  - Network configuration
  - Example use-case
  - Extras

ROS Characteristics

+



+

+

Plumbing

- Process management
- Inter-process communication
- Device drivers

Tools

- Simulation
- Visualization
- Graphical user interface
- Data logging

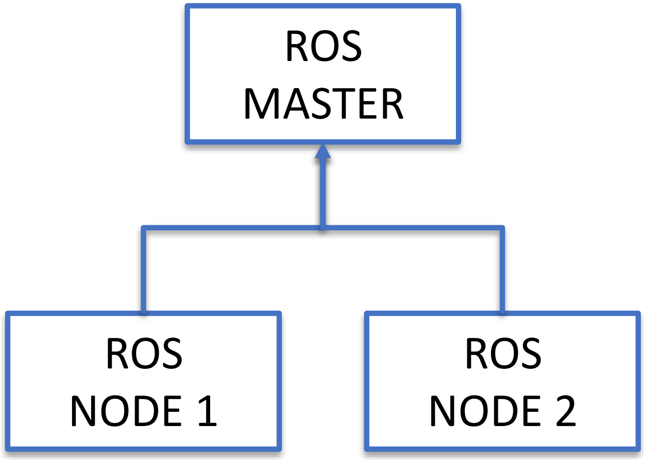
Capabilities

- Control
- Planning
- Perception
- Mapping
- Manipulation

Ecosystem

- Package organization
- Software distribution
- Documentation
- Tutorials

ROS Architecture: Basics

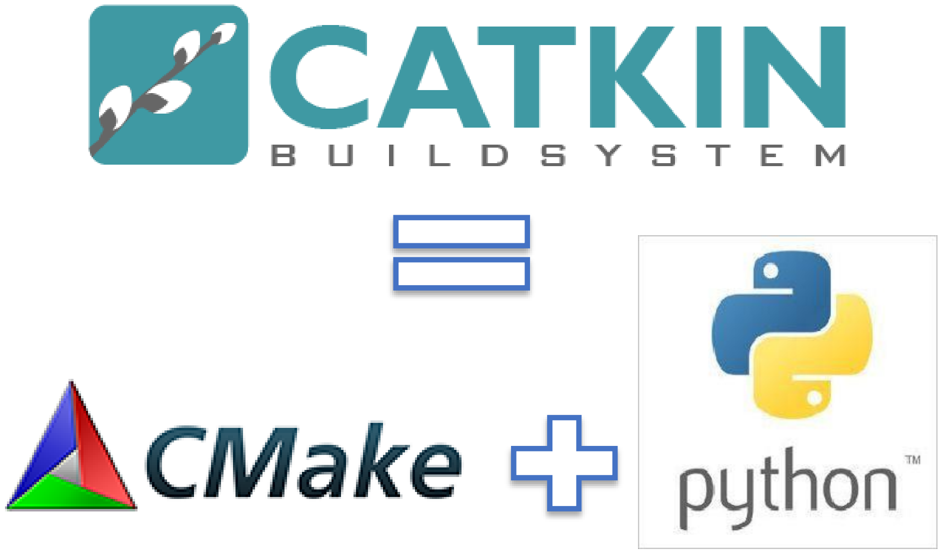


ROS MASTER

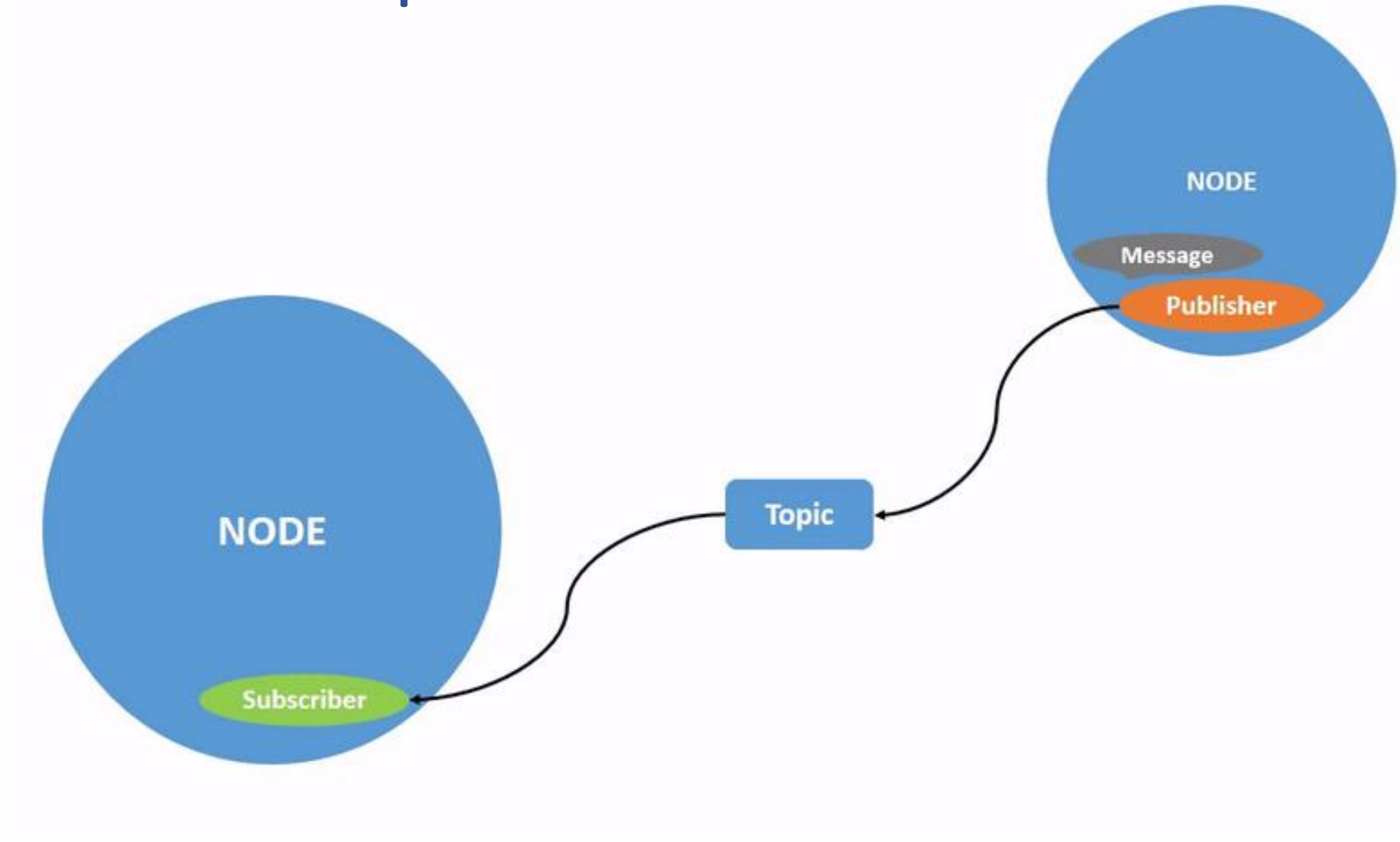
- Manages the communication between nodes (XML-RPC server + naming and communication services)
- Every node registers at start-up with the master
- Nodes can run on different workstation and communicate through network (transparent to user)

ROS NODE

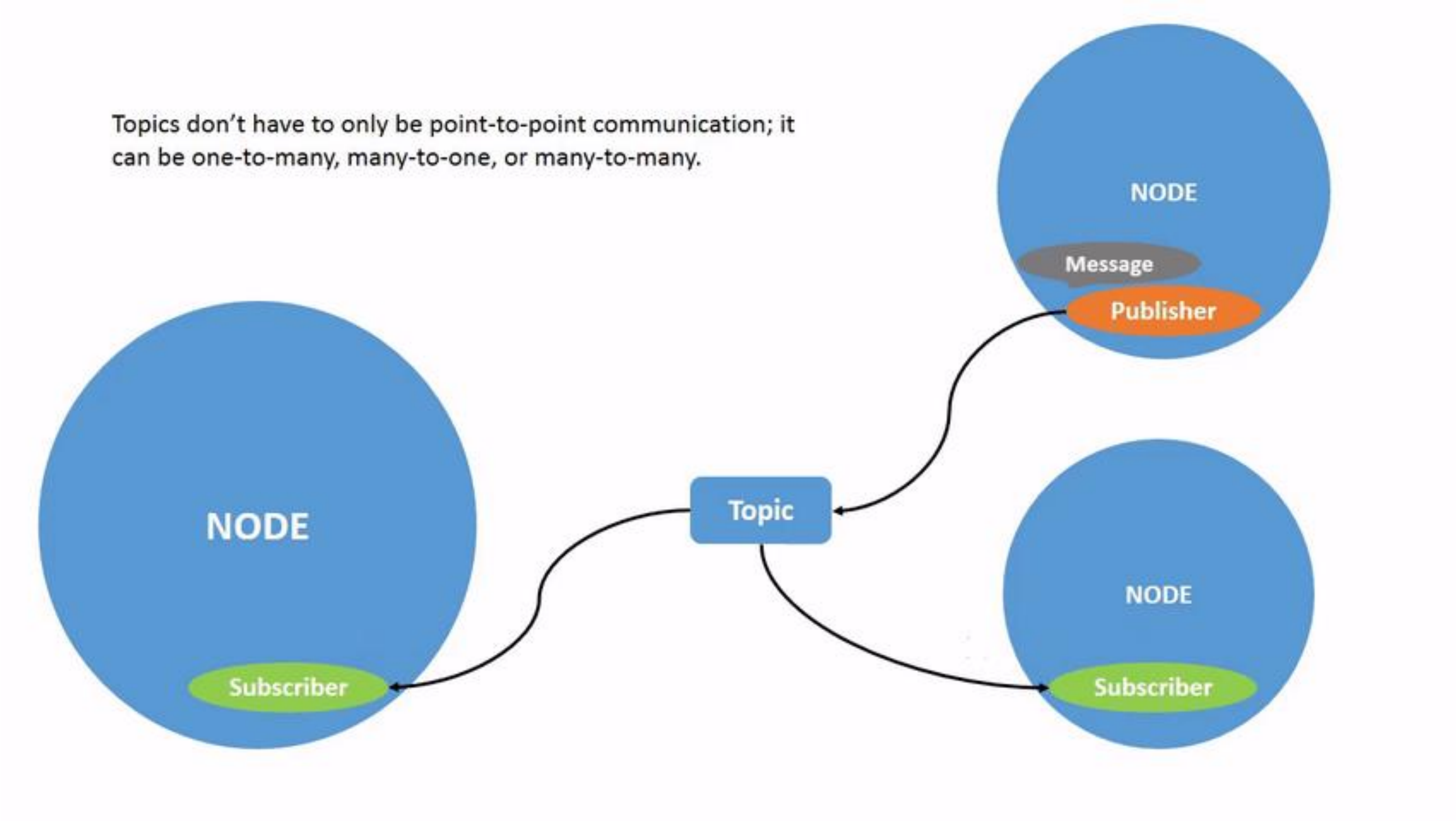
- Single-purpose, executable program
- Individually compiled, executed, and managed
- Organized in *packages*



ROS Nodes simplest communication



ROS Nodes general communication



Topics don't have to only be point-to-point communication; it can be one-to-many, many-to-one, or many-to-many.

ROS Topics

- Nodes communicate over *topics*
  - Nodes can *publish* or *subscribe* to a topic
  - Typically, 1 publisher and *n* subscribers
- Topic is a name for a stream of *messages*

List active topics with

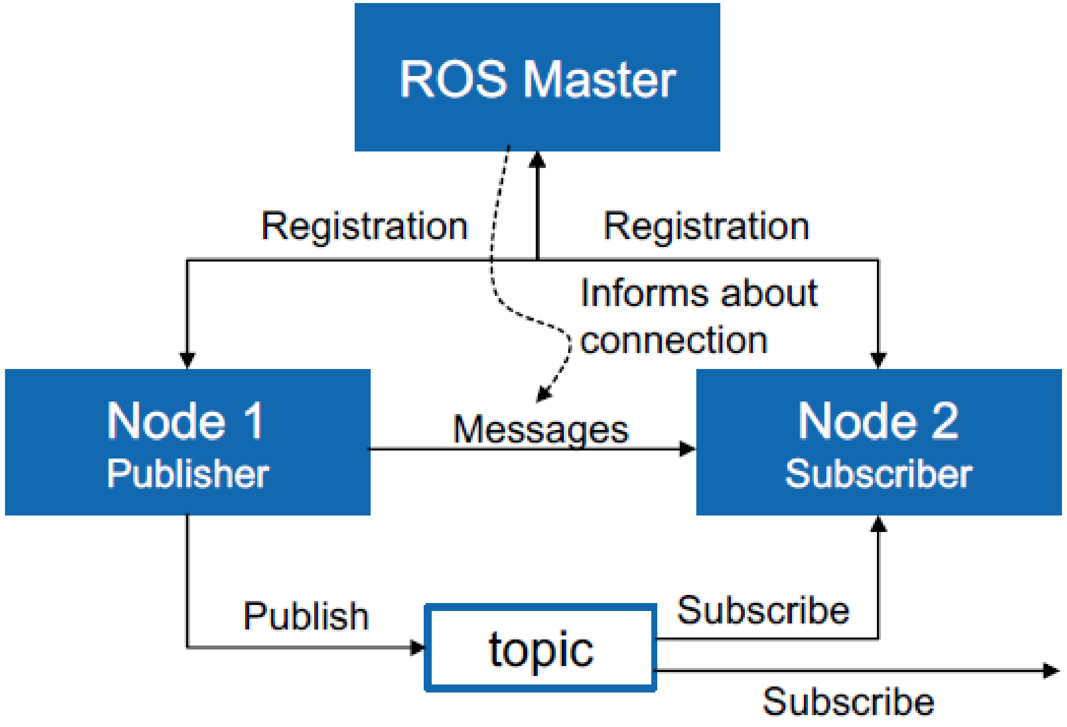
```
> rostopic list
```

Subscribe and print the contents of a topic with

```
> rostopic echo /topic
```

Show information about a topic with

```
> rostopic info /topic
```



More info
http://wiki.ros.org/rostopic

ROS Messages

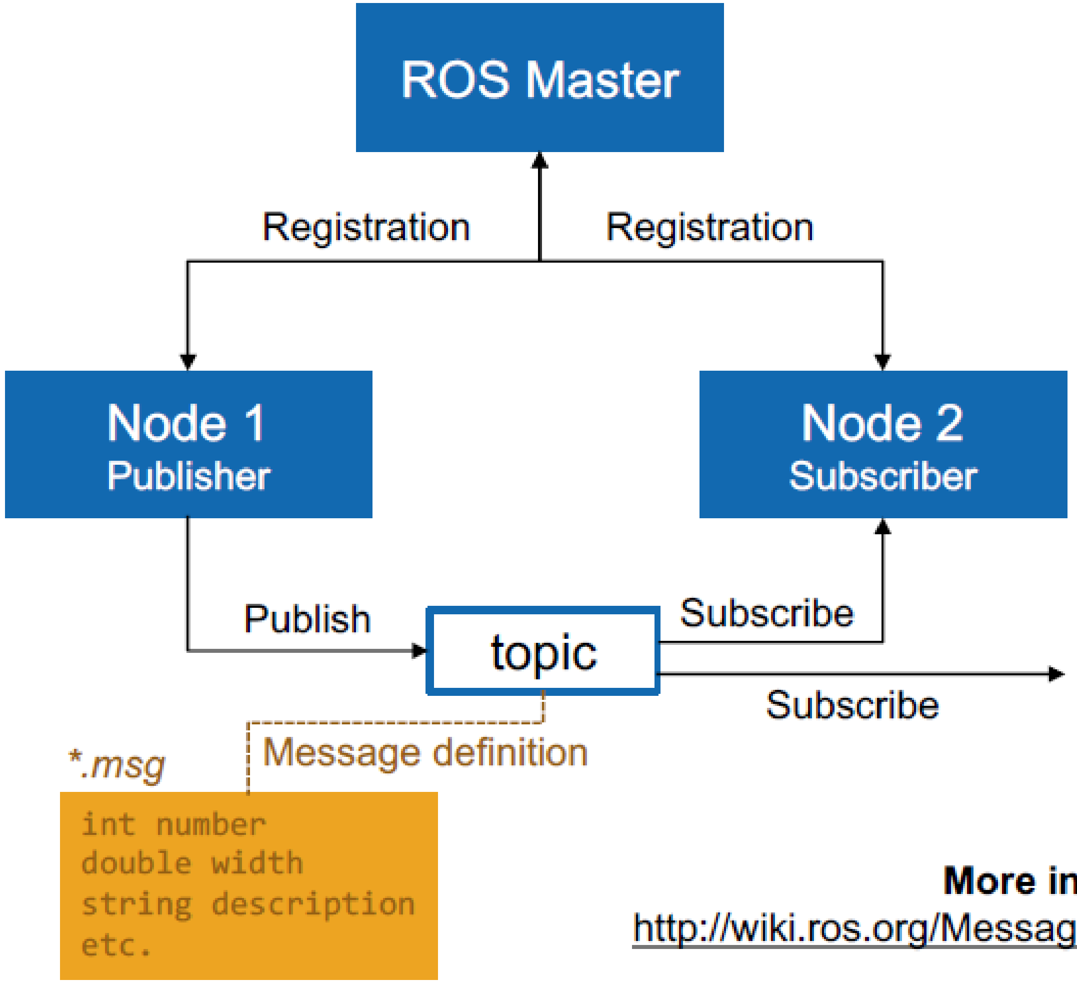
- Data structure defining the *type* of a topic
- Compromised of a nested structure of integers, floats, booleans, strings etc. and arrays of objects
- Defined in **.msg* files

See the type of a topic

```
> rostopic type /topic
```

Publish a message to a topic

```
> rostopic pub /topic type args
```



```
int number
double width
string description
etc.
```

More info
http://wiki.ros.org/Messages

Basic RULES when working with multiple machine

- All the machines must be on the same network.
- There will be only one master (server) as many clients as you need.
- All the workstation have to “see” the same master
- Each workstation can run multiple nodes
- Pay attention to network configuration, since many problem are often related to this part!

Basic STEPS when working with multiple machine

- Configure the network on each machine
- Check network communication on each machine

Required only when you are setting up your ROS demo

- Configure ROS on each machine
- Check ROS communication on each machine
- Start working with ROS ☺

You have to repeat these steps each time you start working in ROS, on each machine and on each terminal

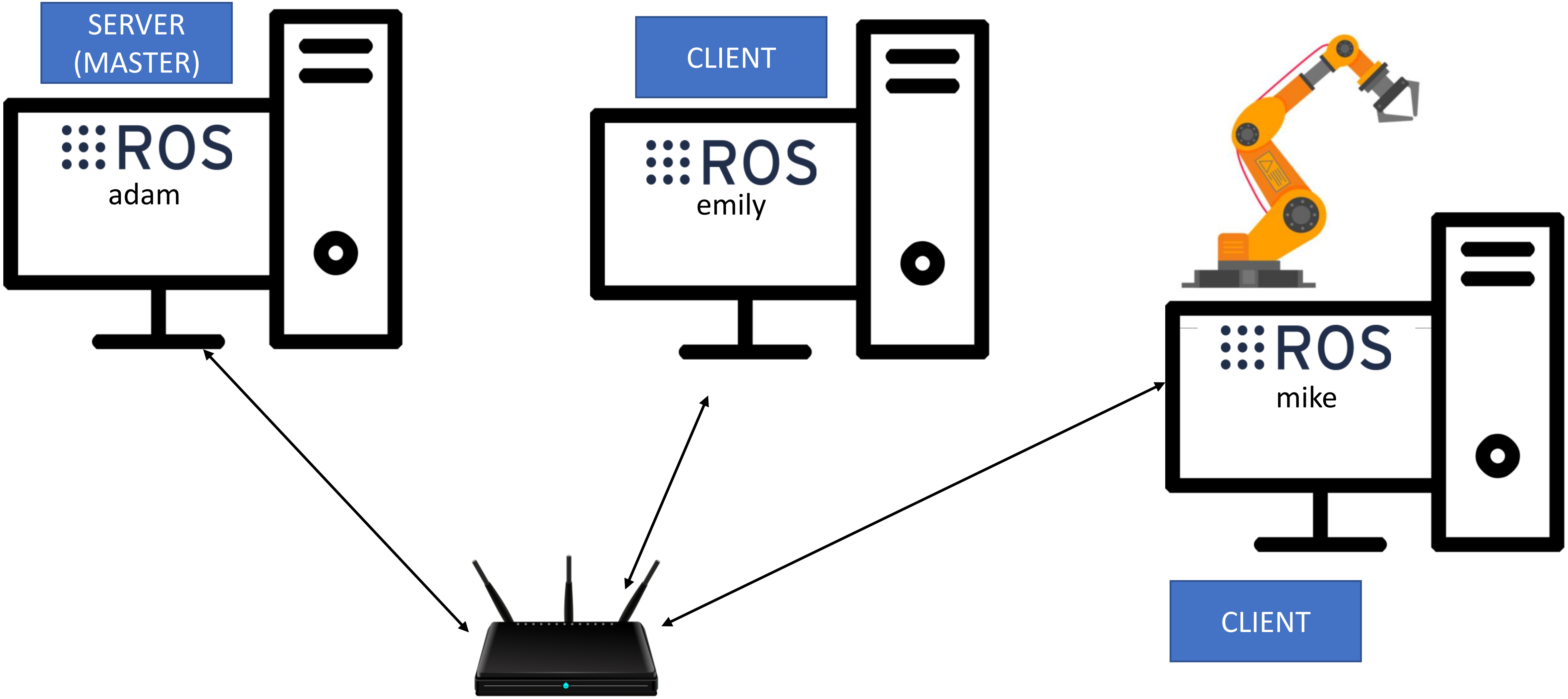
If you have problem with the second part, always re-check the first one!

Example use case (1)

- We have 3 workstations:
  - The master will be running on **adam** (192.168.203.1).
  - **emily** (192.168.203.2) and **mike** (192.168.203.3) will be running other nodes.
  - **adam** could also run some other ROS nodes

  - This is just an example, try using meaningful names and also avoid using “ws” (e.g. control_ws) since it is often used for referring to your catking workspace

Example use case (2)



Setup hostnames

- Add IP addresses and hostnames of all the other machines in the network, in /etc/hosts file. Remember to set static IP addresses.
- This is an example of /etc/hosts file on adam’s machine:
  - 127.0.0.1 localhost
  - …
  - 192.168.203.2 emily
  - 192.168.203.3 mike
- The same goes for emily and mike machines.
- This step is otpional, but it lets you type the hostname instead of the IP address each time.

Check connection between machines

- There must be complete, bi-directional connectivity between all pairs of machines -> Ping the workstations to check connectivity.

- From **adam**, try to ping **emily** and **mike**
  - ping emily
  - ping mike

- Then, ping from **emily** to **adam** and **mike**, and so on…

ROS Commands

[1] Start the server (master) on **adam**’s terminal

```
› roscore
```

[2] Setup the environment in client terminals (**emily** and **mike**)

```
› export ROS_MASTER_URI=http://adam:11311
› …
```

- **Remember to perform [2] on each terminal you open.**

- **Also, don’t forget to configure the ROS environment before everything else, i.e. source the desired catkin workspace.**

Configuring the ROS environment

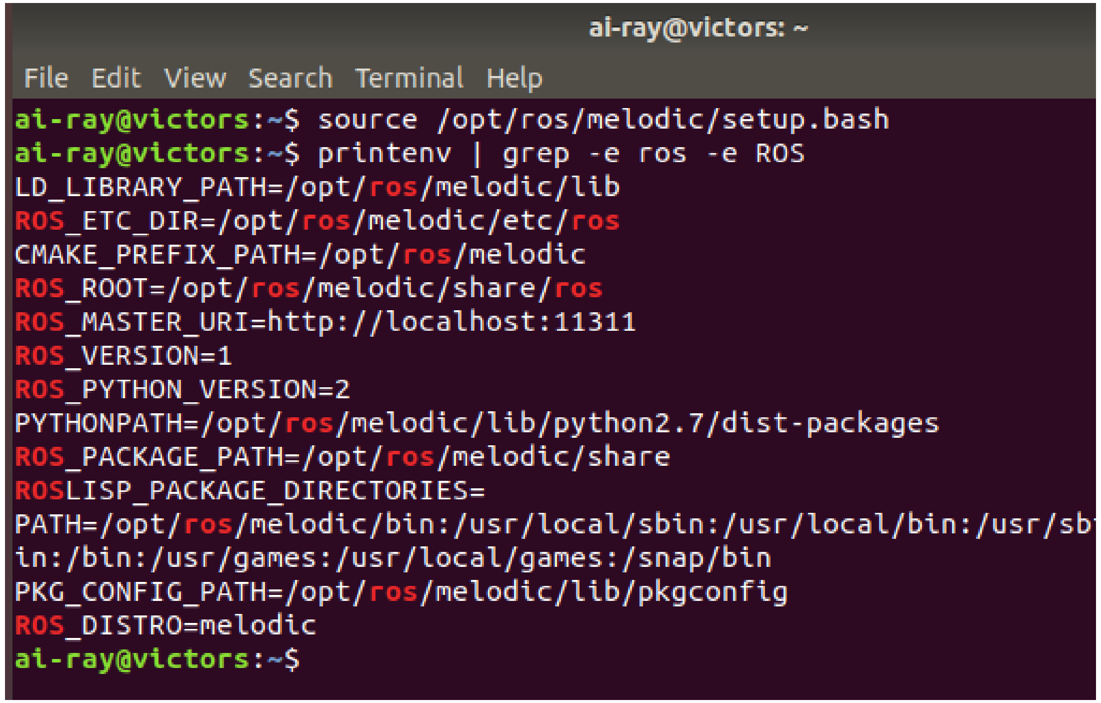
```
source /opt/ros/kinetic/setup.bash
```

This command is fundamental for correctly configuring all environment variables

Most of the problem with ROS are related to problems with this config…

Essential variables are:

- ROS_ROOT sets the location where the ROS core packages are installed.
- ROS_MASTER_URI is a required setting that tells nodes where they can locate the master.



```
ai-ray@victors:~$ source /opt/ros/melodic/setup.bash
ai-ray@victors:~$ printenv | grep -e ros -e ROS
LD_LIBRARY_PATH=/opt/ros/melodic/lib
ROS_ETC_DIR=/opt/ros/melodic/etc/ros
CMAKE_PREFIX_PATH=/opt/ros/melodic
ROS_ROOT=/opt/ros/melodic/share/ros
ROS_MASTER_URI=http://localhost:11311
ROS_VERSION=1
ROS_PYTHON_VERSION=2
PYTHONPATH=/opt/ros/melodic/lib/python2.7/dist-packages
ROS_PACKAGE_PATH=/opt/ros/melodic/share
ROSLISP_PACKAGE_DIRECTORIES=
PATH=/opt/ros/melodic/bin:/usr/local/sbin:/usr/local/bin:/usr/sb
in:/bin:/usr/games:/usr/local/games:/snap/bin
PKG_CONFIG_PATH=/opt/ros/melodic/lib/pkgconfig
ROS_DISTRO=melodic
ai-ray@victors:~$
```

Example: /etc/hosts on different machines

Adam

```
/etc/hosts

127.0.0.1            localhost
.
.
.
192.168.203.2        emily
192.168.203.3        mike
```

Mike

```
/etc/hosts

127.0.0.1            localhost
.
.
.
192.168.203.1        adam
192.168.203.2        emily
```

Emily

```
/etc/hosts

127.0.0.1            localhost
.
.
.
192.168.203.1        adam
192.168.203.3        mike
```

Example: check network comunication

Adam

```
> ping emily
PING emily (192.168.203.2) 56(84) bytes of data.
64 bytes from 127.0.0.1: icmp_seq=1 ttl=128 time=0.123 ms
64 bytes from 127.0.0.1: icmp_seq=2 ttl=128 time=0.094 ms
…
> ping mike
PING mike (192.168.203.3) 56(84) bytes of data.
64 bytes from 127.0.0.1: icmp_seq=1 ttl=128 time=0.123 ms
64 bytes from 127.0.0.1: icmp_seq=2 ttl=128 time=0.094 ms
…
```

Mike

```
> ping adam
PING adam (192.168.203.1) 56(84) bytes of data.
64 bytes from 127.0.0.1: icmp_seq=1 ttl=128 time=0.123 ms
64 bytes from 127.0.0.1: icmp_seq=2 ttl=128 time=0.094 ms
…
> ping emily
PING emily (192.168.203.2) 56(84) bytes of data.
64 bytes from 127.0.0.1: icmp_seq=1 ttl=128 time=0.123 ms
64 bytes from 127.0.0.1: icmp_seq=2 ttl=128 time=0.094 ms
…
```

Emily

```
> ping adam
PING adam (192.168.203.1) 56(84) bytes of data.
64 bytes from 127.0.0.1: icmp_seq=1 ttl=128 time=0.123 ms
64 bytes from 127.0.0.1: icmp_seq=2 ttl=128 time=0.094 ms
…
> ping mike
PING mike (192.168.203.3) 56(84) bytes of data.
64 bytes from 127.0.0.1: icmp_seq=1 ttl=128 time=0.123 ms
64 bytes from 127.0.0.1: icmp_seq=2 ttl=128 time=0.094 ms
…
```

Example: ROS configuration

Adam

```
> roscore

SUMMARY
========

PARAMETERS
 * /rosdistro: melodic
 * /rosversion: 1.14.11

NODES

auto-starting new master
process[master]: started with pid [25916]
ROS_MASTER_URI=http://adam:11311/

setting /run_id to e73610ea-1886-11ec-81f0-04d9f5d41c6f
process[rosout-1]: started with pid [25944]
started core service [/rosout]

…
```

Mike

```
> export ROS_MASTER_URI=http://adam:11311
> rostopic list

/rosout
/rosout_agg
/tf

> …
```

Emily

```
> export ROS_MASTER_URI=http://adam:11311
> rostopic list

/rosout
/rosout_agg
/tf

> …
```

Extra (1)

- To solve problems related to timestamps, it could be useful to synchronize the clocks with a shared server.
- I recommend to do this this step at the beginning of your setup to avoid facing communication/synchronization problems (complex to find and solve)
- Run the following command on each terminal

```
› sudo ntpdate ntp.ubuntu.com
```

- If **ntpdate** is not available:

```
› sudo apt-get install ntpdate
```

- A cleaner and automatic solution can be obtained with chrony.

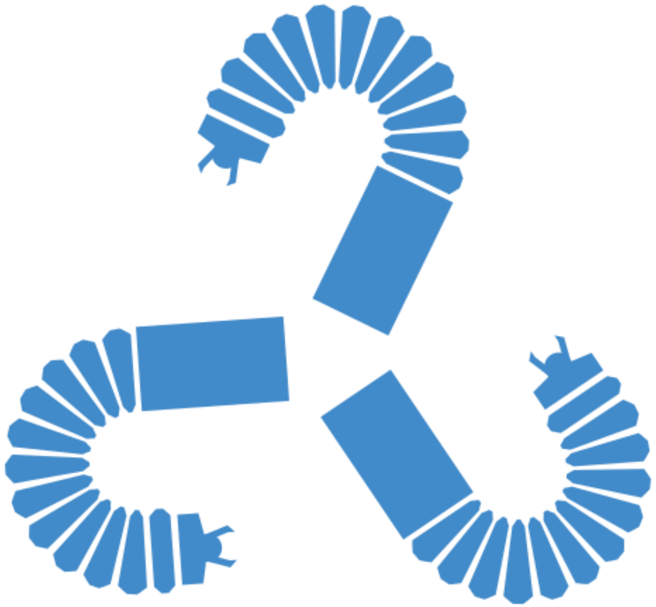
Extras (2)

- The standard roscore port is 11311, but it can be changed to a different one.

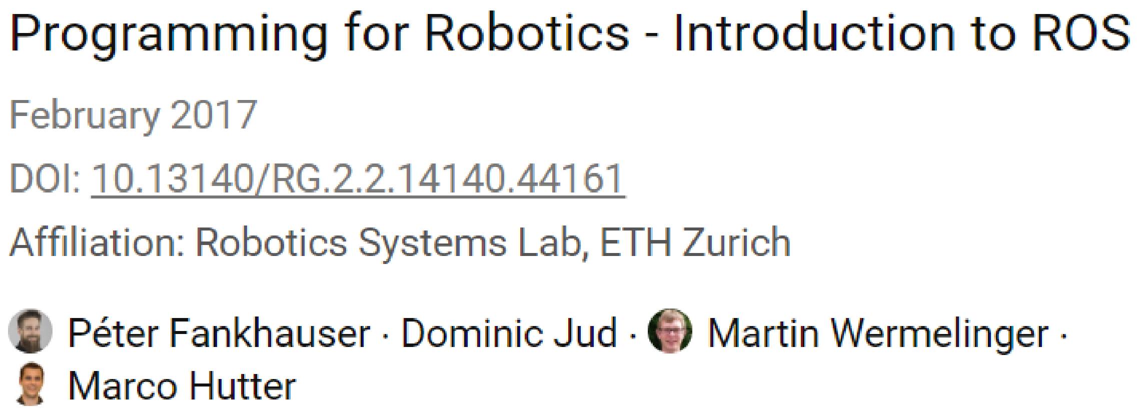
- ROS is well integrated in other software:
  - MATLAB
    https://www.mathworks.com/products/ros.html
  - C# (Unity3D, Windows)
    https://github.com/siemens/ros-sharp
    https://github.com/Unity-Technologies/Unity-Robotics-Hub
  - rosbridge protocol
    https://github.com/RobotWebTools/rosbridge_suite

Sources

1. http://wiki.ros.org/ROS/NetworkSetup
2. http://wiki.ros.org/ROS/Tutorials/MultipleMachines



The contents of these slides are partially based on:



Programming for Robotics - Introduction to ROS

February 2017

DOI: 10.13140/RG.2.2.14140.44161

Affiliation: Robotics Systems Lab, ETH Zurich

Péter Fankhauser · Dominic Jud · Martin Wermelinger · Marco Hutter