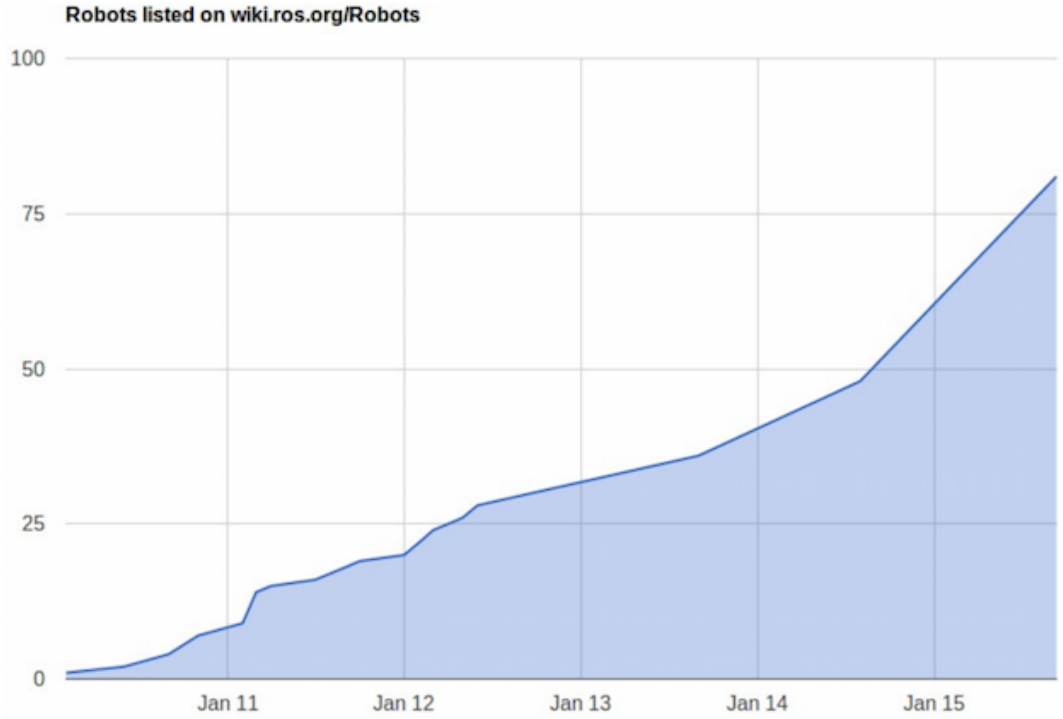
Robotic Manipulation

Introduction to ROS and git

Tran Daulet

Robotic Operating System

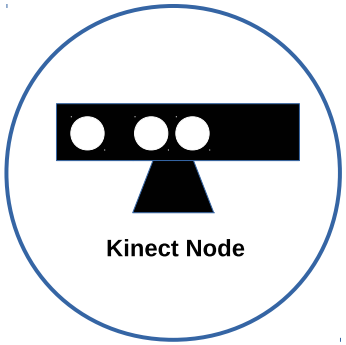
- ROS stand for Robotic Operating System and was released 2007 by a company known as Willow Garage.
- ROS is an open-source, meta-operating system for your robot.
- ROS is designed to be modular at a fine-grained scale.
- ROS is widely used in industry and academic research[1]

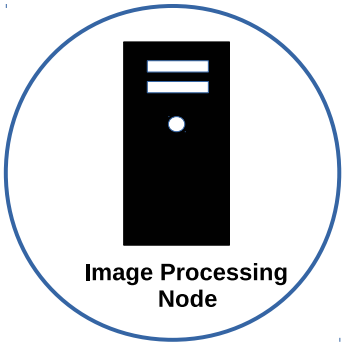


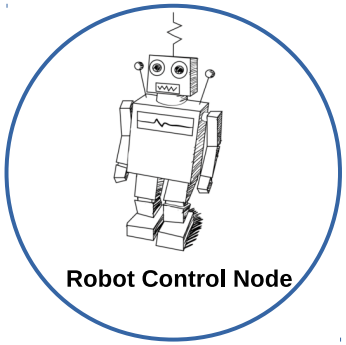
[1] https://spectrum.ieee.org/automaton/robotics/robotics-software/ros-robot-operating-system-celebrates-8-years

ROS concept

- ROS is build up of nodes

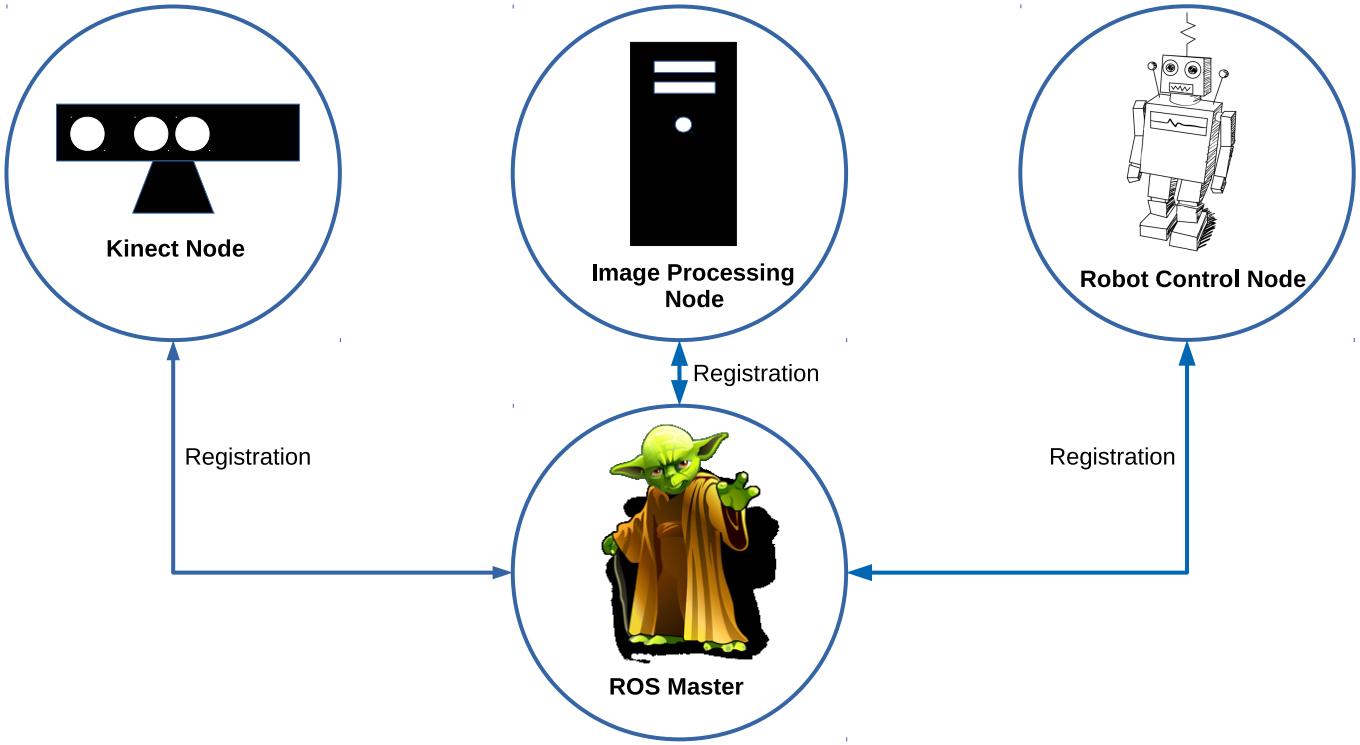






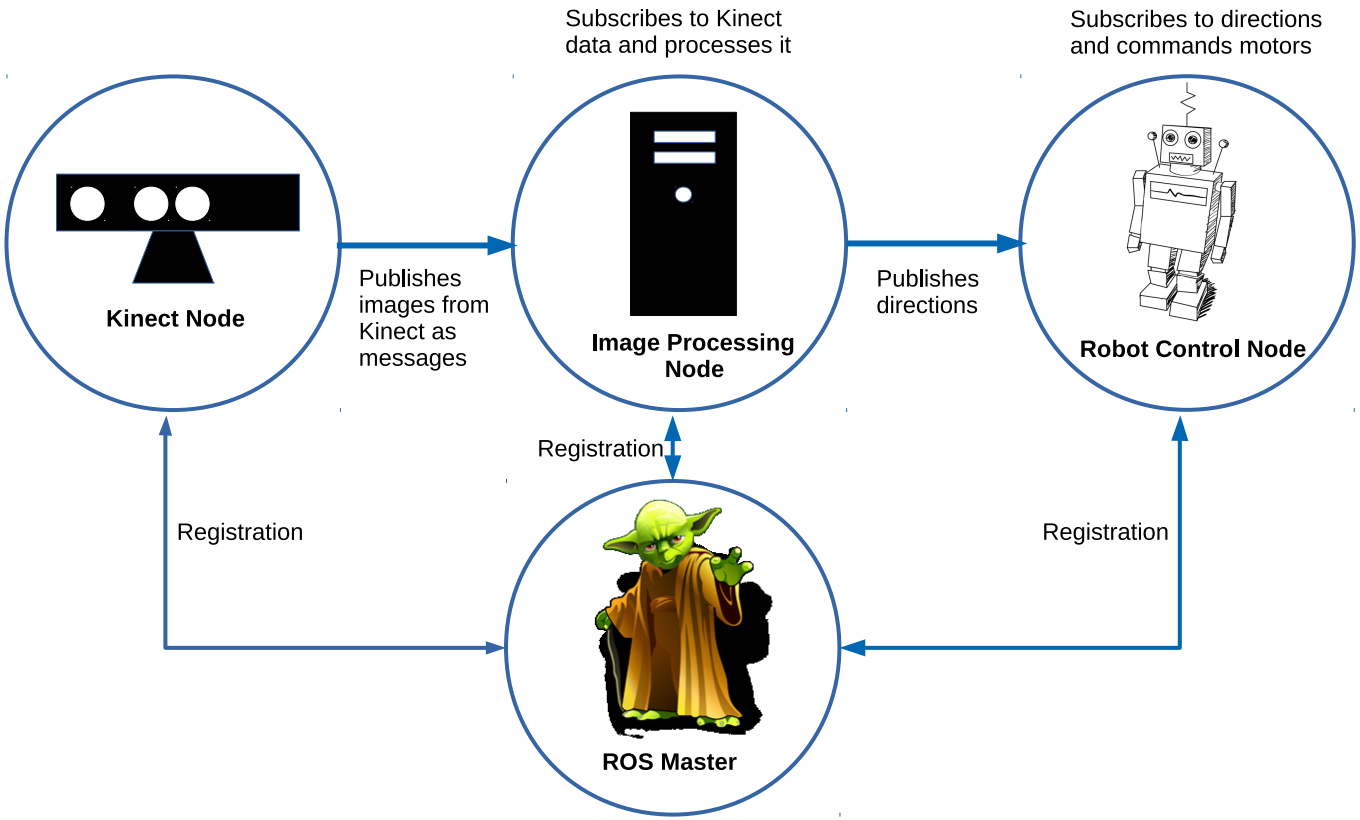
ROS concept

- ROS is build up of nodes
- ROS nodes are registered through a ROS Master



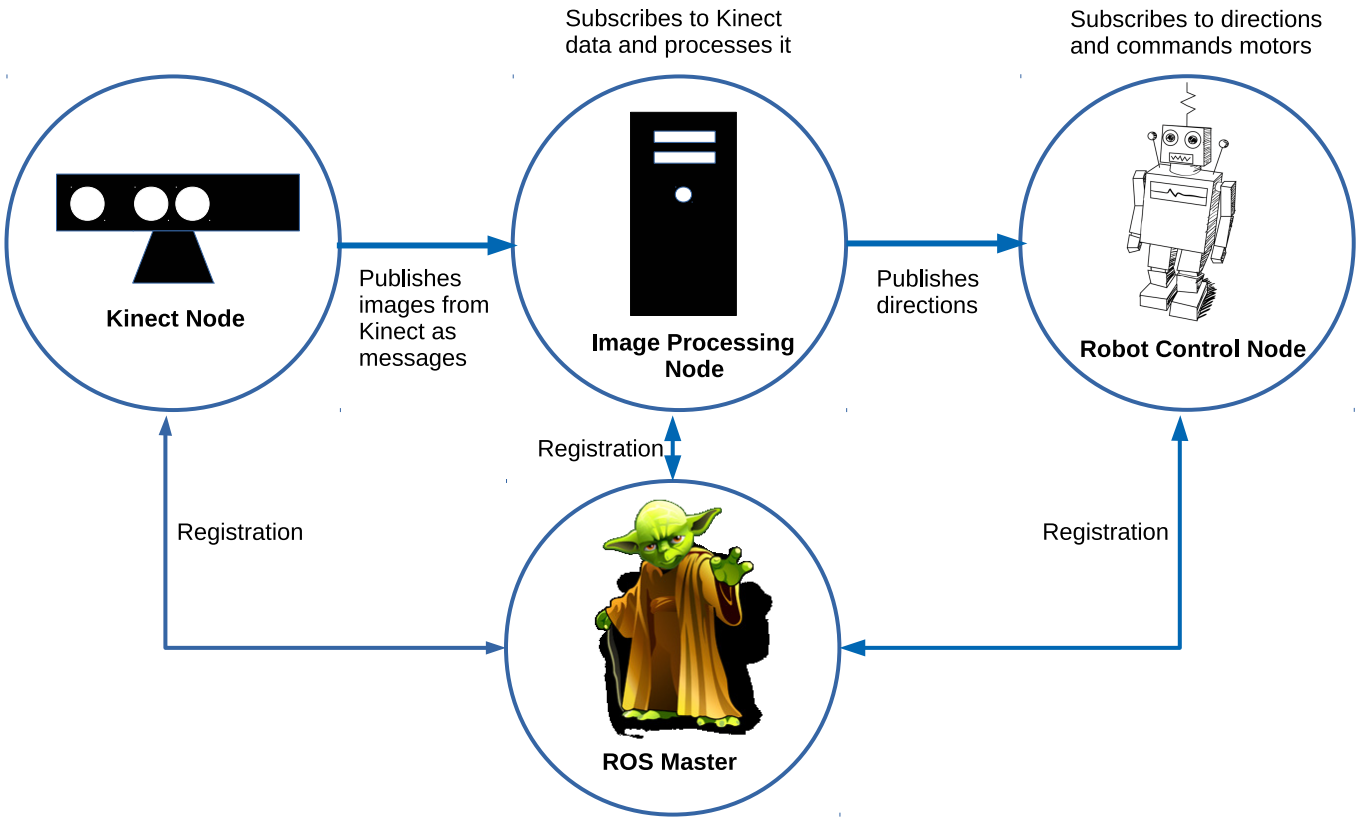
ROS concept

- ROS is build up of nodes
- ROS nodes are registered through a ROS Master
- Nodes can communicate with each other via topics



ROS concept

- ROS is build up of nodes
- ROS nodes are registered through a ROS Master
- Nodes can communicate with each other via topics
- For more in depth knowledge about ROS you can read, for example, `http://wiki.ros.org/ROS/Introduction`



Creating and using a ROS workspace

- Interactive session during the exercise session.
- You can also find information about creating a workspace and how to source the bash script from ROS wiki page `http://wiki.ros.org/catkin/Tutorials/create_a_workspace`
- Always remember to source devel/setup.bash in your workspace after you compiled the code in order to access the newly compiled ROS nodes.

git

- git is a version-control system.
- In this course, gitlab is used for storing all exercises. If you have no previous knowledge of git and/or gitlab then please read up about it online at, *e.g.* `https://docs.gitlab.com/ee/gitlab-basics/`
- To use Aalto gitlab you need to log in to `version.aalto.fi` and then set up your ssh key (`https://docs.gitlab.com/ee/gitlab-basics/create-your-ssh-keys.html`).
- or follow these two links here and here.

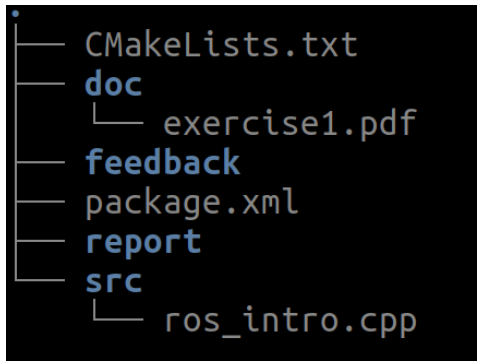
Creating a gitlab group, forking the course material, and pushing code

- Interactive session during the exercise session.
- For the gitlab repository, we created one subgroup for each one of you. You can use the following pattern to access that: `https://version.aalto.fi/gitlab/robotic_manipulation_students_2023/<youremailaddresswithout@aalto.fi>`
- On your computer, remember to always clone your newly forked exercise repository into the src directory of your ROS workspace

Exercise file system

The file system for each exercise is visualized in the figure to the right

- The src folder contains the template code you need to fix
- The feedback folder will contain the TA's feedback and points awarded
- In the report folder you will upload the exercise report as a pdf
- The docs folder will contain all necessary information for the current exercise.
- Other files are ROS specific which you do not need to touch.

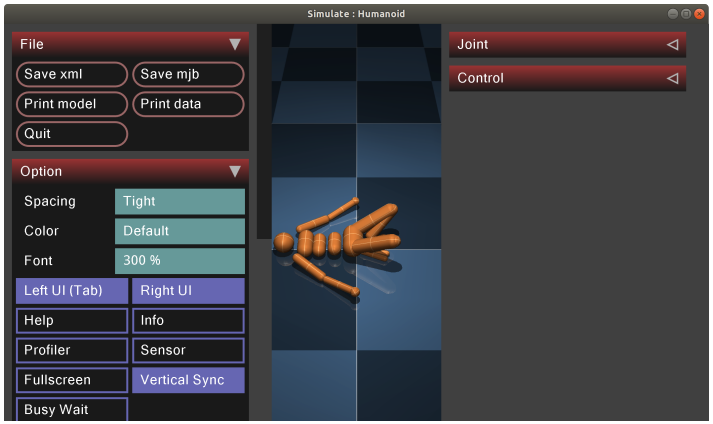


MuJoCo setup

- Download *mujoco200 linux* at `https://www.roboti.us/index.html`
- Download the MuJoCo license in MyCouses under the "For Aalto users" tab. **IMPORTANT**: The license is for **personal use only** and cannot be redistributed!
- Unzip the downloaded mjpro200 directory into /.mujoco/mjpro200, and place your license key (the mjkey.txt file) at /.mujoco/mjpro200/bin/mjkey.txt.
- Test if MuJoCo runs by opening a terminal and write from **.mujoco/** folder

```
~/.mujoco/mjpro200/bin/simulate ~/.mujoco/mjpro200/model/humanoid.xml
```

and check if the window that opens is similar to the one below



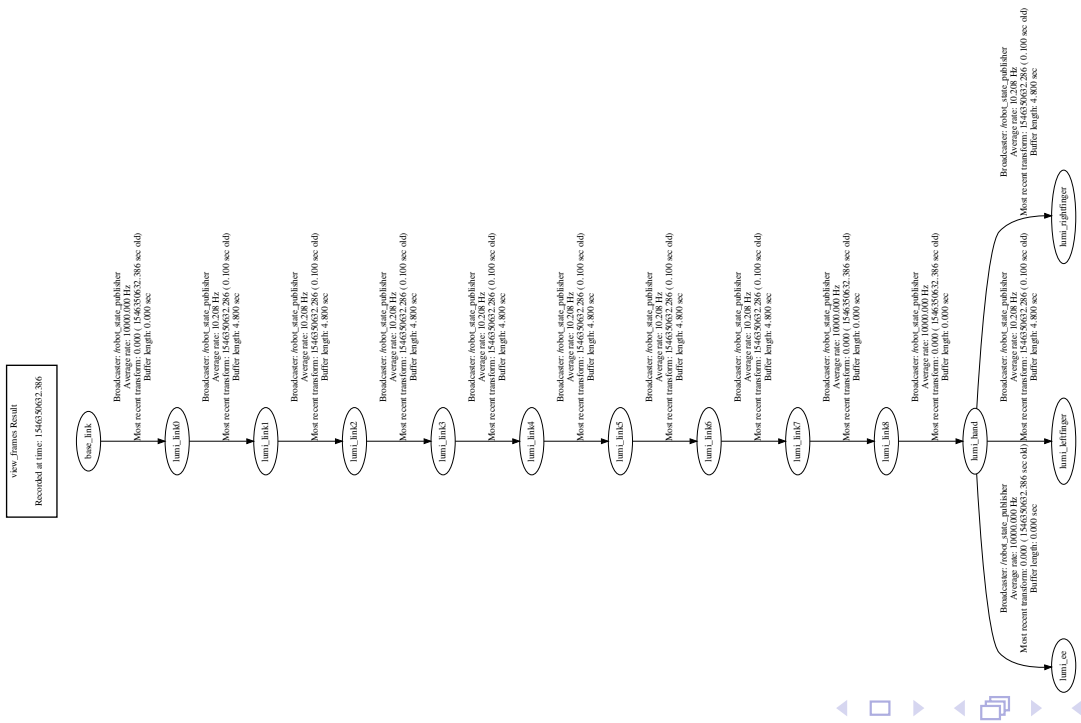
RViz and TF tree

- A robotic system typically has many 3D coordinate frames that change over time. These coordinate systems are naturally expressed in a transformation (TF) tree `http://wiki.ros.org/tf`.

RViz and TF tree

- A robotic system typically has many 3D coordinate frames that change over time. These coordinate systems are naturally expressed in a transformation (TF) tree `http://wiki.ros.org/tf`.
- You can visualize the current TF tree (`http://wiki.ros.org/tf/Debugging_tools`) by typing

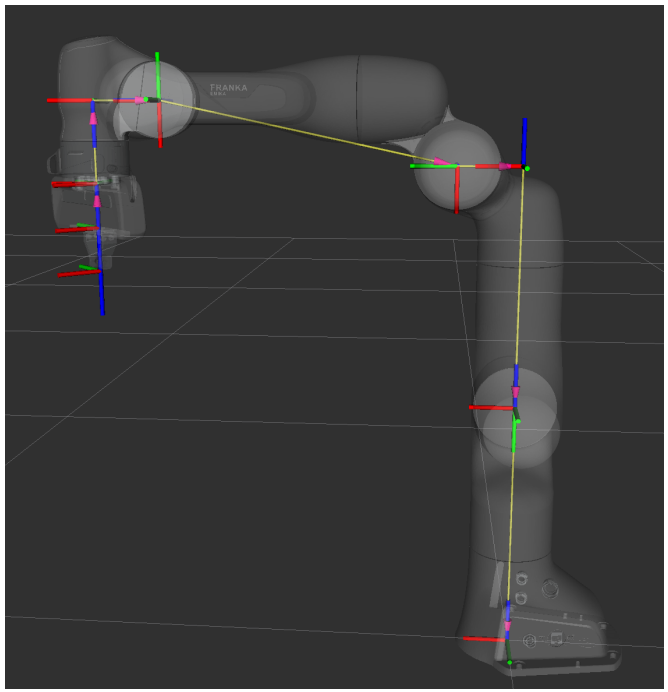
```
roslaunch lumi_description show.launch
rosrun tf view_frames && evince frames.pdf
```



RViz and TF tree

- A robotic system typically has many 3D coordinate frames that change over time. These coordinate systems are naturally expressed in a transformation (TF) tree `http://wiki.ros.org/tf`.
- You can visualize the current TF tree (`http://wiki.ros.org/tf/Debugging_tools`) by typing

```
roslaunch lumi_description show.launch
rosrun tf view_frames && evince frames.pdf
```



What did we not cover?

- Specifically to ROS, we did not cover concepts such as:
  - ROS Services `http://wiki.ros.org/Services`,
  - ROS Parameter Server `http://wiki.ros.org/Parameter`,
  - ROS Bags `http://wiki.ros.org/Bags`,
  - and much more `http://wiki.ros.org/ROS/Concepts`.
- With respect to Git we did not cover concepts such as
  - Git Branching and Merging `https://git-scm.com/book/en/v2/Git-Branching-Basic-Branching-and-Merging`
  - git-revert `https://git-scm.com/docs/git-revert.html`
  - git-diff `https://git-scm.com/docs/git-diff`
  - and much more `http://thepilcrow.net/explaining-basic-concepts-git-and-github/`
- You will probably not need to master nor need these concepts during the course, but it is good to know about them.