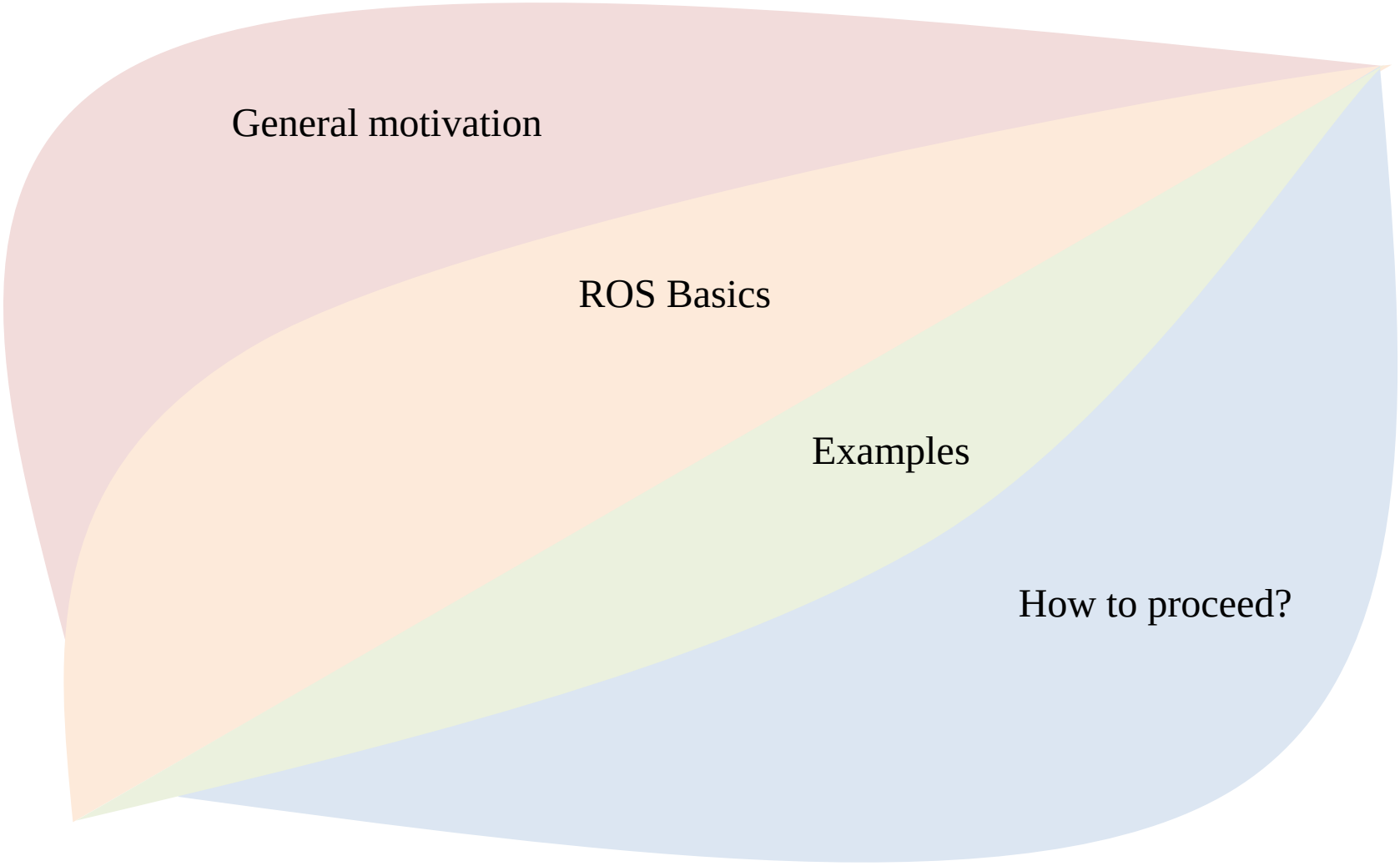
Maastricht University

Introduction to Robot Operating System

Kirill Tumanov
k.tumanov@maastrichtuniversity.nl

2017

Outline



What is it all about?

What is ROS?

- An open-source full stack OS for robotic development
- Extendable environment for robot control

Why ROS?

- It is free
- It is fast
- It is well-structured
- It is scalable
- It has a lot of things built-in
- A vast community

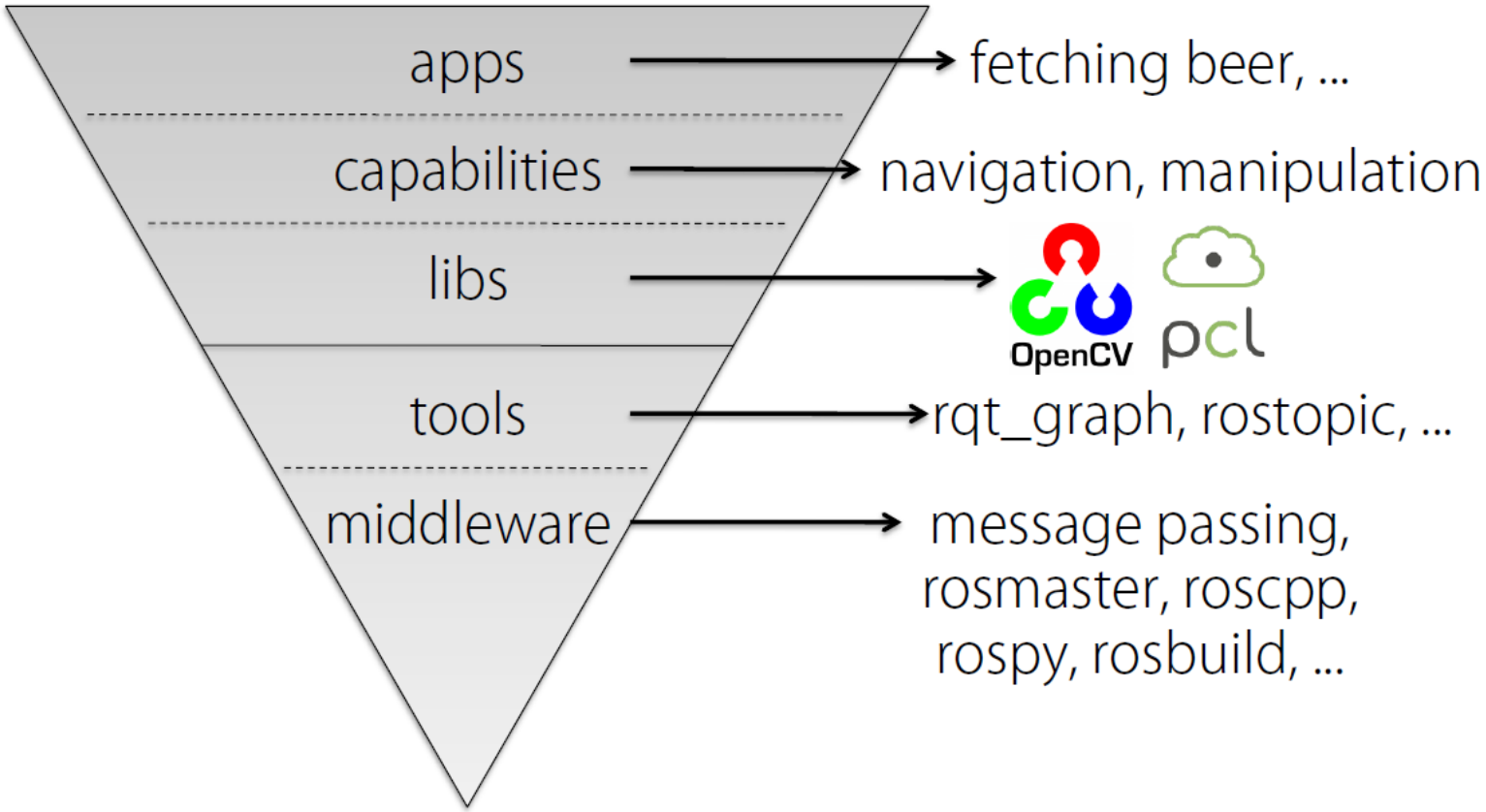
If not ROS, then what?

Open-source suites:
- URBI
- CARMEN Toolkit (Carnegie Mellon)
- Player Project
- Mobile Robot Programming Toolkit (MRPT)
- …

Suites with license restrictions:
- Webots (commercial)
- Microsoft Robotics Developer Studio (MRDS)
- …

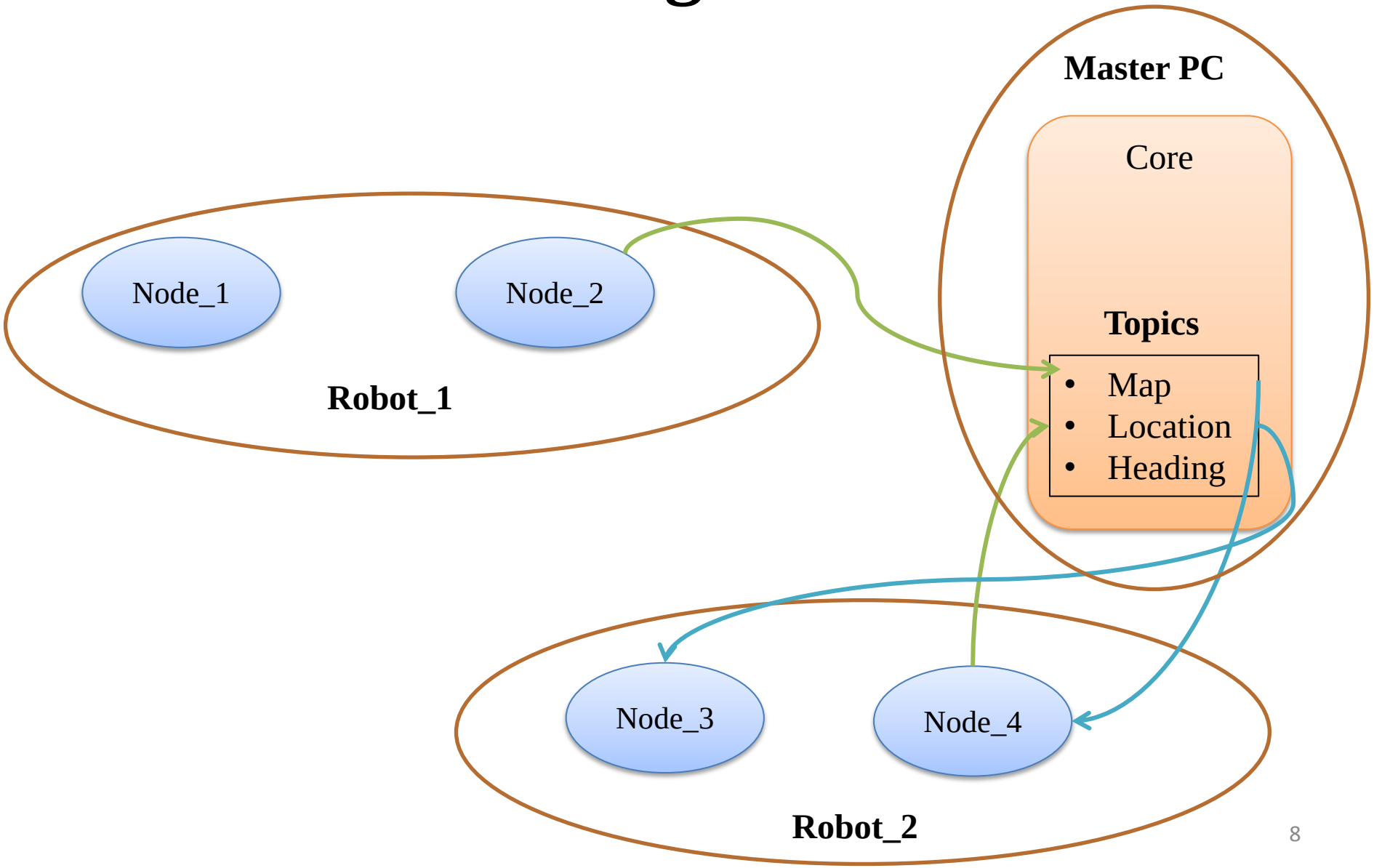
Software and hardware limitations:
- LEGO Mindstorms NXT
- NAO Choreographe
- …

ROS: Stack



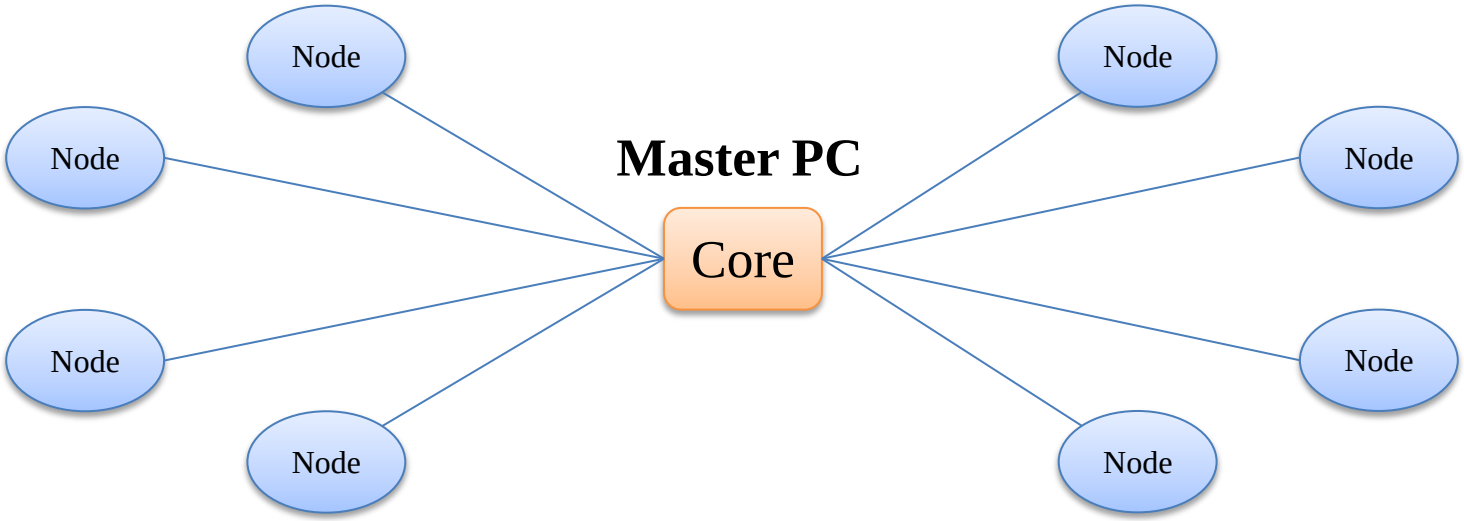
Courtesy: Daniel Claes, 2013

ROS: Organization



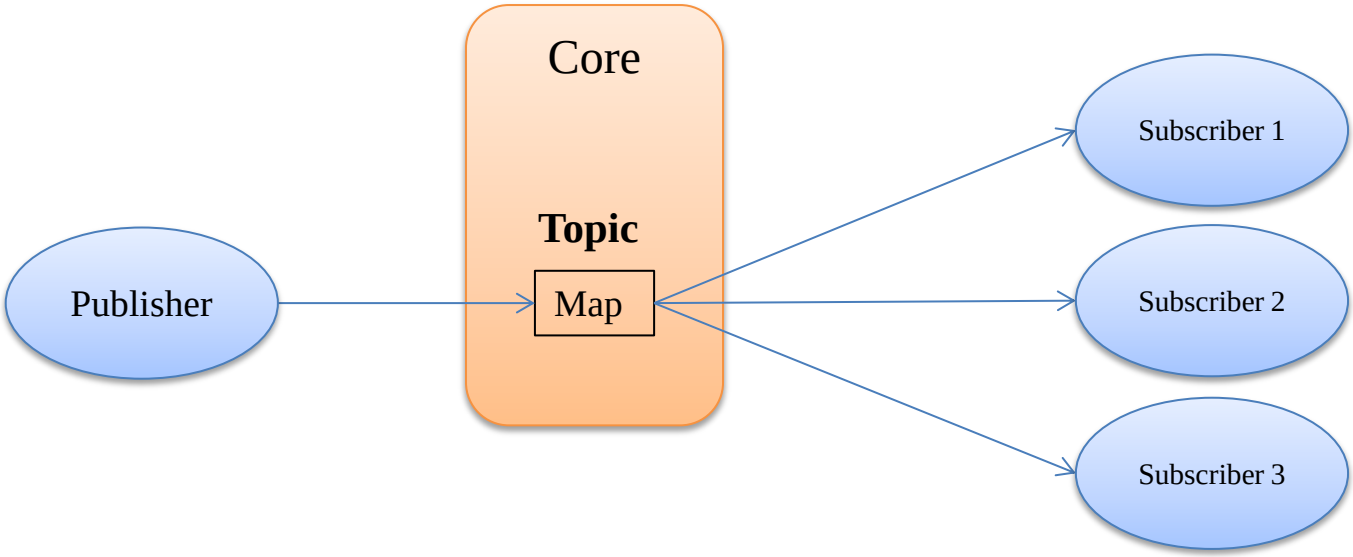
ROS: Nodes

- Node ⇔ Process
- Node is a module of a (complex) robot controller which eases maintenance
- Nodes run in parallel in a possibly distributed robotic environment
- Nodes know about each other and communicate through a *roscore* running on a master PC



ROS: Topics

- Topic ⇔ Named Message Log
- Topic allows sending information in a robotic environment
- Falls under the client-server or publisher-subscriber relationship
- Information dropped by publisher on a topic is picked-up by (multiple) subscribers



ROS: Messages

- Message ⇔ Filled data structure
- Sending messages allows to pass information in an organized fashion
- Message fields are well-documented so you know what you will receive

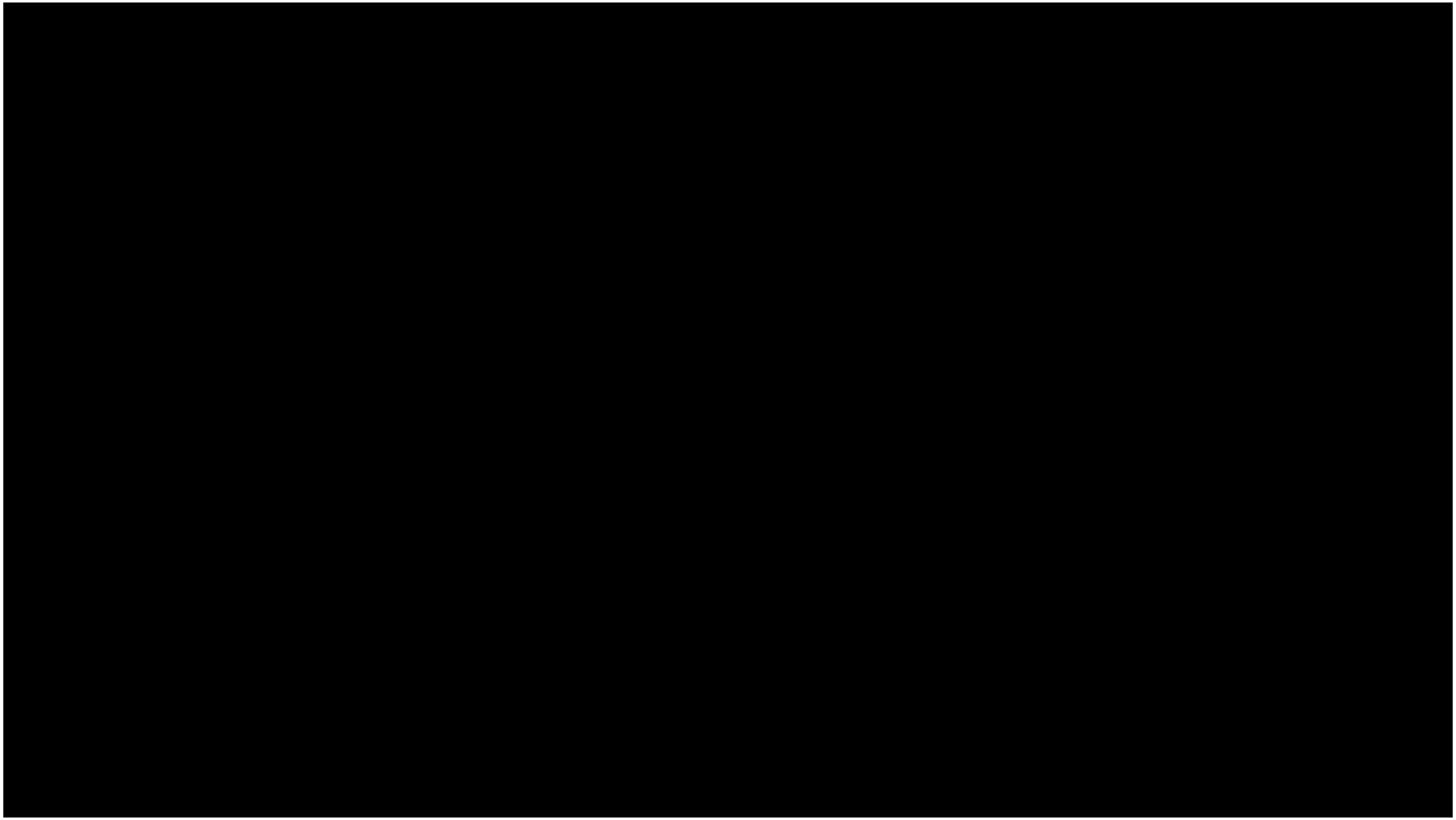
```
Map:
        header:
                seq:            123987
                stamp:          1441980732
                frame_id:       1
        info:
                time:           1441983633
                height:         120
                width:          230
                resolution:     0.05
                origin:
                        point:          (0,0,0)
                        orientation:    (0,0,0,0)

        data:   [0 1 0 0 100 0 0 100 …]
```

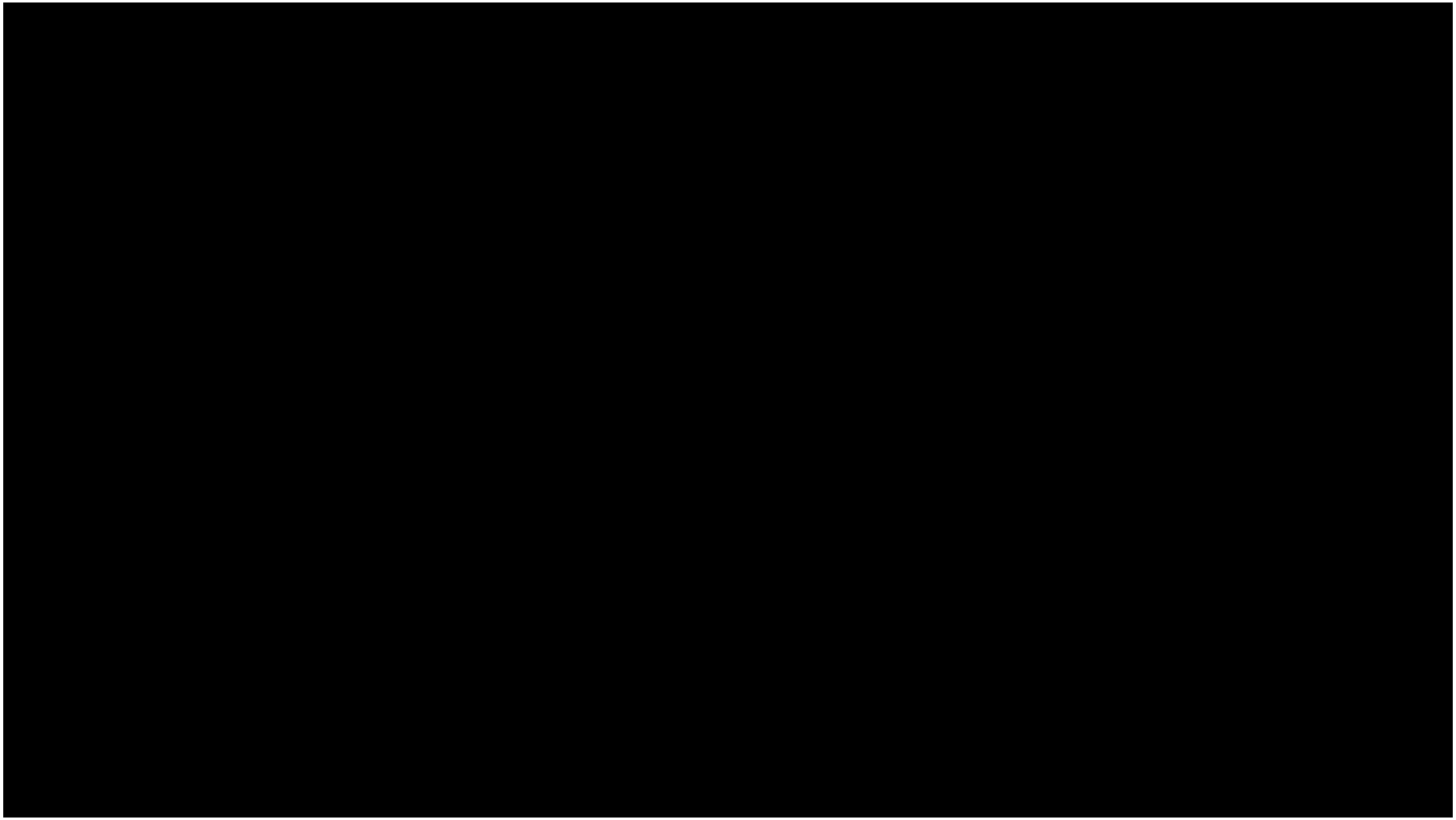
ROS: Overview

- Runs of Linux (Ubuntu), Windows, MacOS, Android
- Supports C++, Python, Lisp, Java, Lua
- Plug-and-Play ⇔ Install and Develop
- Everything can be made custom, but …
- … usually there is already more than one solution of what you want to do
- Compatible with leading toolkits: Gazebo, Player/Stage, Orocos, …
- Switching from simulation to real-life is made as easy as possible – really
- Scale your solutions without (much) pain

ROS use example



Another ROS use example …



More ROS examples

- See Swarmlab videos: https://www.youtube.com/user/SwarmLabMaastricht/videos
- Search others online

What to start with?

- Check out ROS website: ros.org
- Check Swarmlab page containing instructions on ROS installation: http://swarmlab.unimaas.nl/autonomous-systems/installation/

swarmlab
robots, agents & interaction
maastricht university

What will you do with ROS?

ROS-based assignments:

- Assignment #1 (Random walk) is not submitted, **but** you are encouraged to do it
- Assignment #2 (Bayes Filter) is submitted in **2** weeks time
- Assignment #3 (Particle Filter) is submitted in 1 week
- Assignment #4 (Navigation) is not submitted, **but** you are encouraged to do it

Follow the guidelines in the assignment,
and upload working code on Student Portal.

If in doubt of have any questions see FAQ:
http://swarmlab.unimaas.nl/autonomous-systems/faq/

Questions?

Kirill Tumanov | k.tumanov@maastrichtuniversity.nl