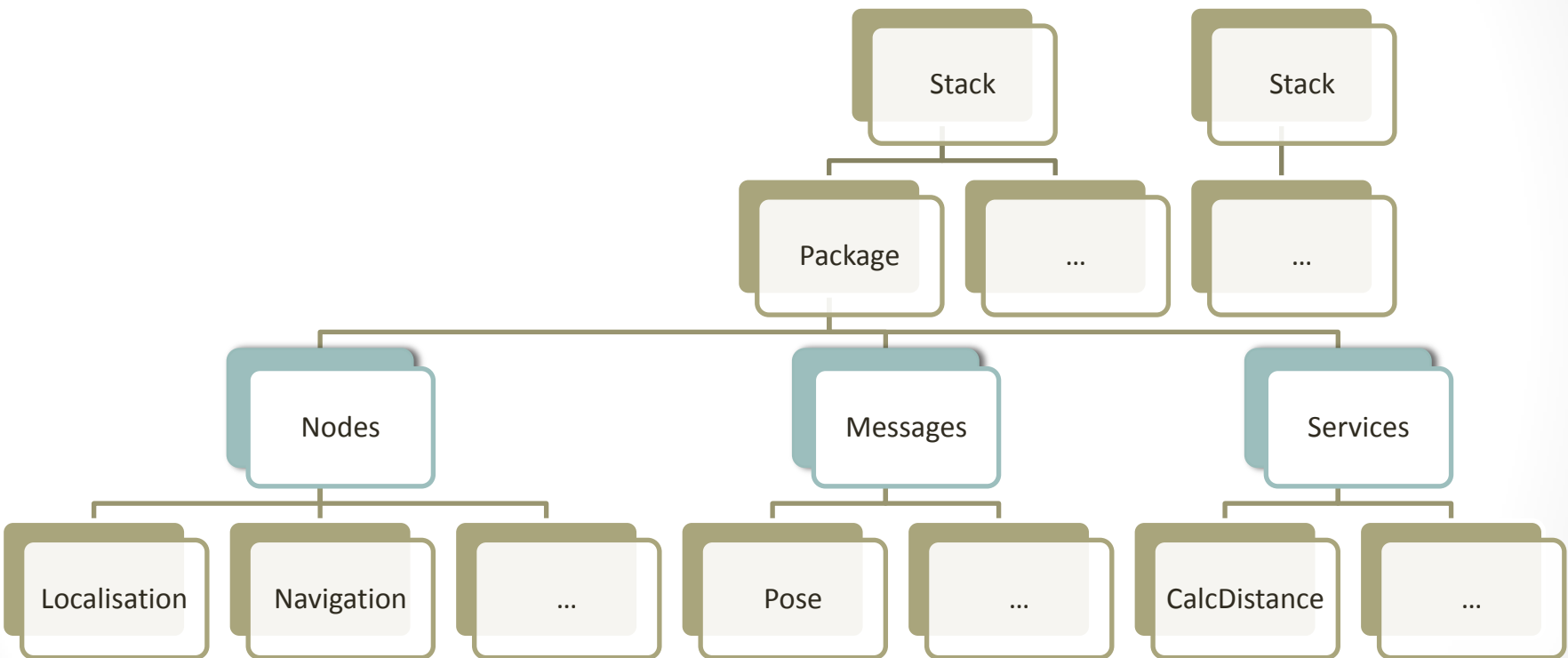


# ROS - Robotic Operating System

Luís Oliveira

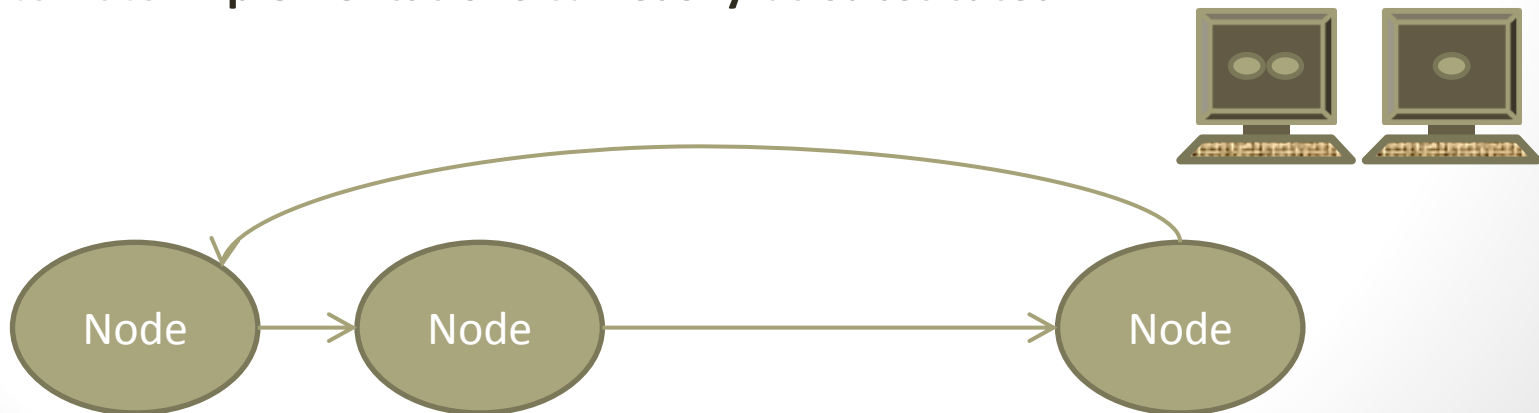
# INTRODUCTION

# System Organisation



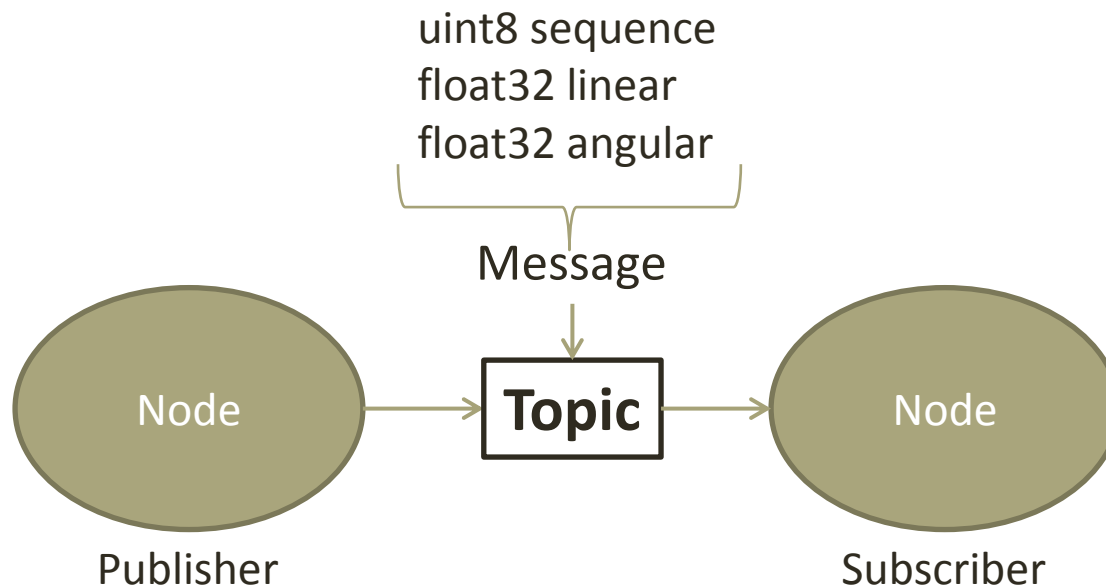
# Nodes

- A **node** is a **process** that performs computation.
  - Are **combined** together **into a graph** and **communicate** with one another using **topics, services**, and the **parameter server**.
  - **Need** a special service (**Master**) which **constructs** the **graph**
  - **A robot** control system will usually comprise **many nodes**. For example:
    - one node controls the robot's wheel motors
    - one node performs localization
- **Pros:**
  - There is an **additional fault tolerance** as crashes are **isolated** to individual **nodes**.
  - Code **complexity is reduced** in comparison to monolithic systems.
  - **Implementation** details are **hidden**
  - Alternate **implementations** can **easily** be **substituted**.

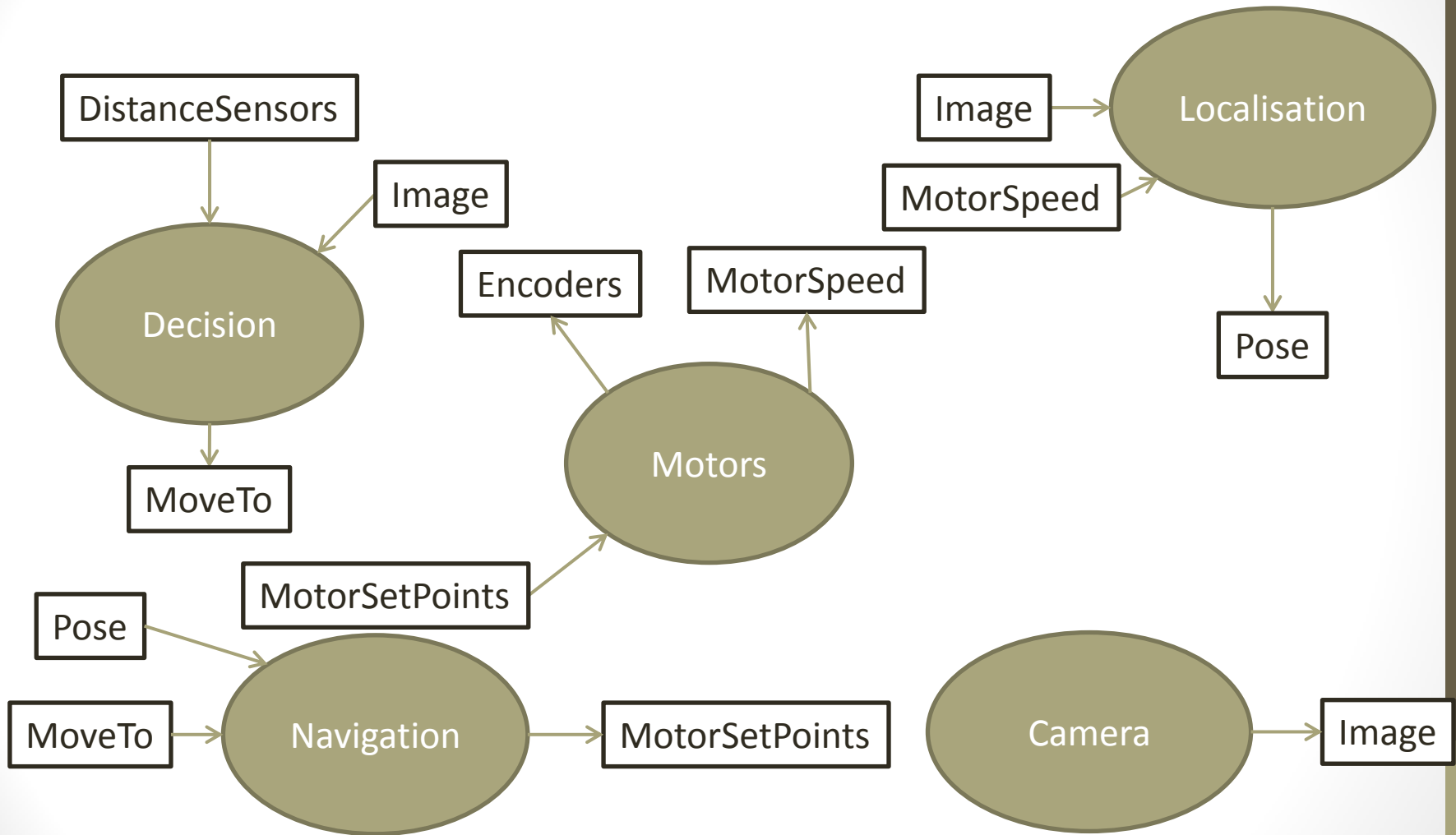


# Topics (Publish/Subscribe Model)

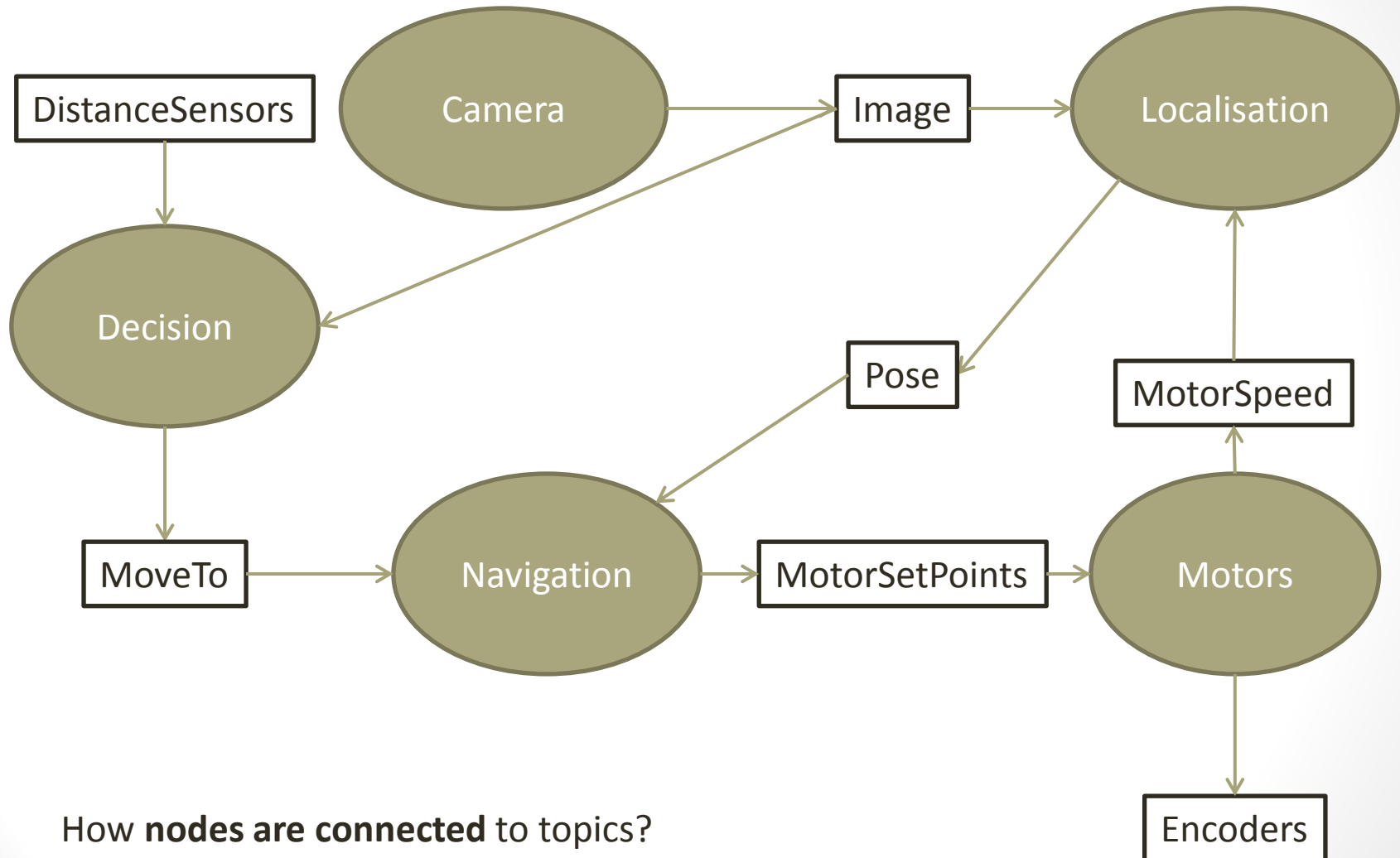
- **Topics** are named buses over which **nodes exchange messages**
  - Nodes are **not aware of who** they are communicating with.
  - Nodes that **generate** data **publish** to a topic.
  - Nodes that are **interested** in data **subscribe** the relevant topic.
  - There can be **multiple publishers** and **subscribers** to a topic.
  - A topic is **associated with a message**.



# Topics (Publish/Subscribe Model)



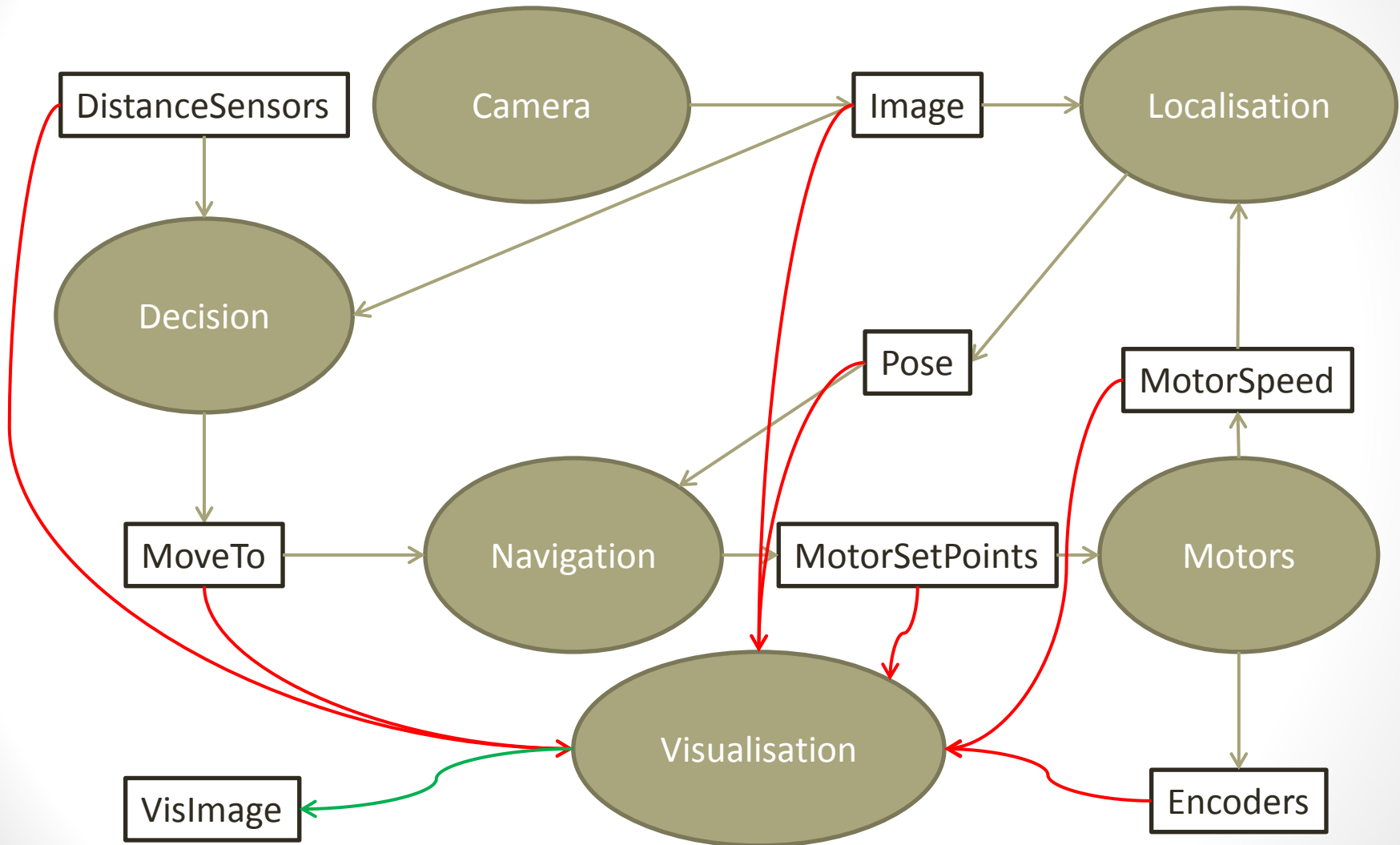
# Topics (Publish/Subscribe Model)



How **nodes** are connected to topics?

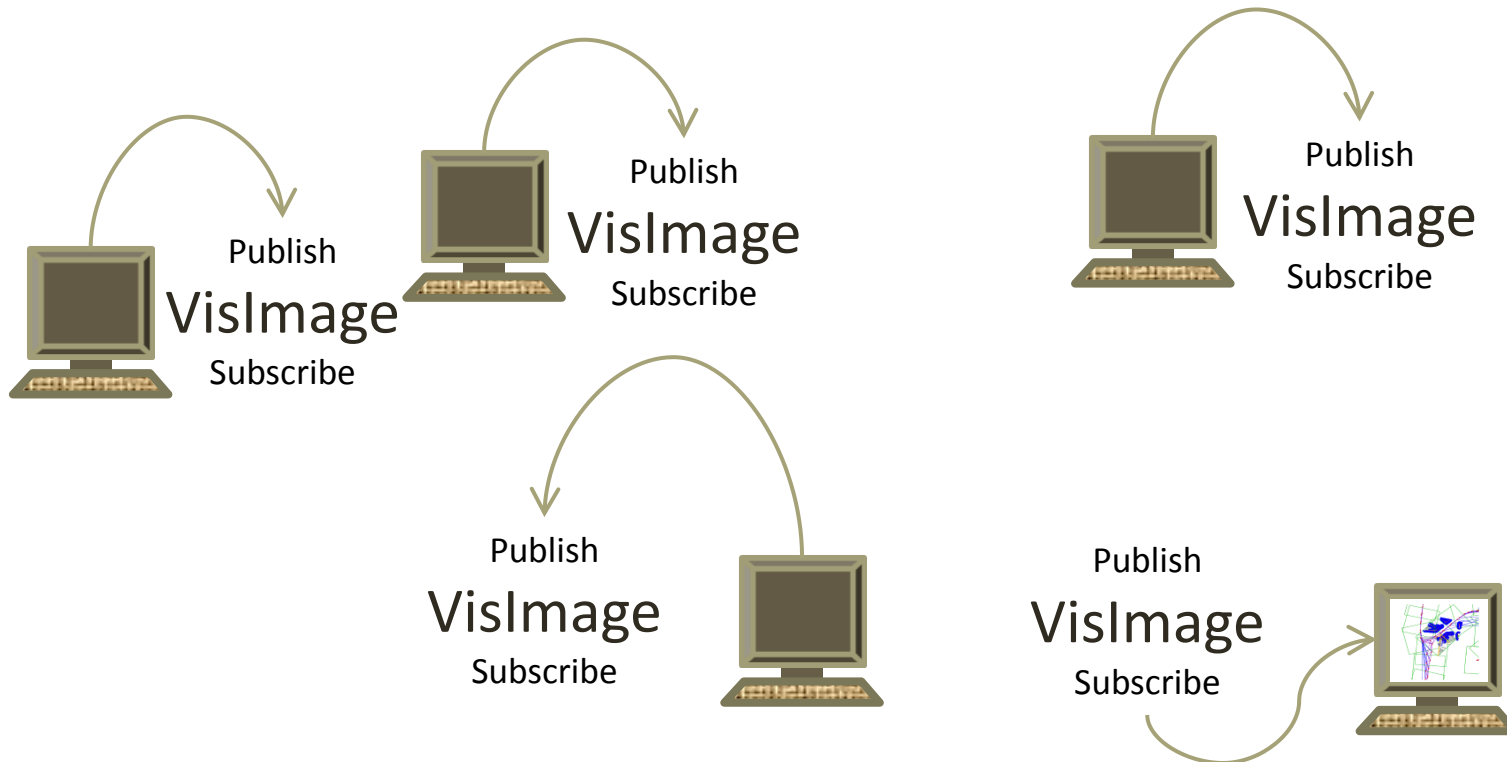
Through the master: <http://www.ros.org/wiki/Master>

# Topics (Publish/Subscribe Model)



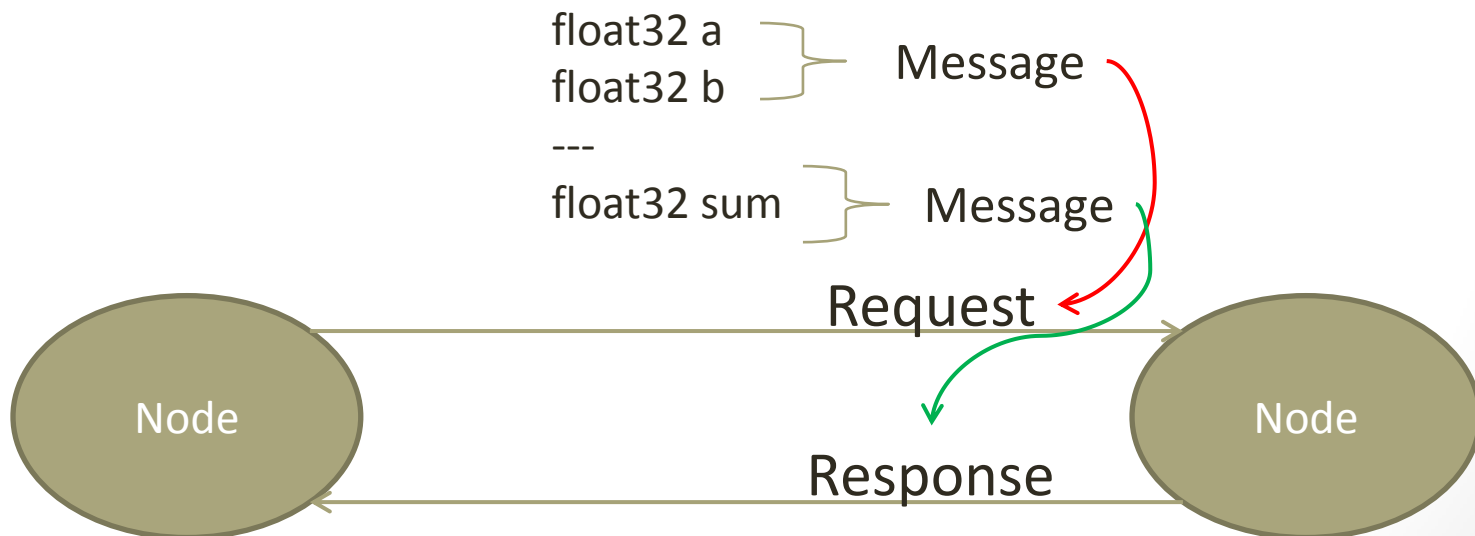


# Topics (Publish/Subscribe Model)

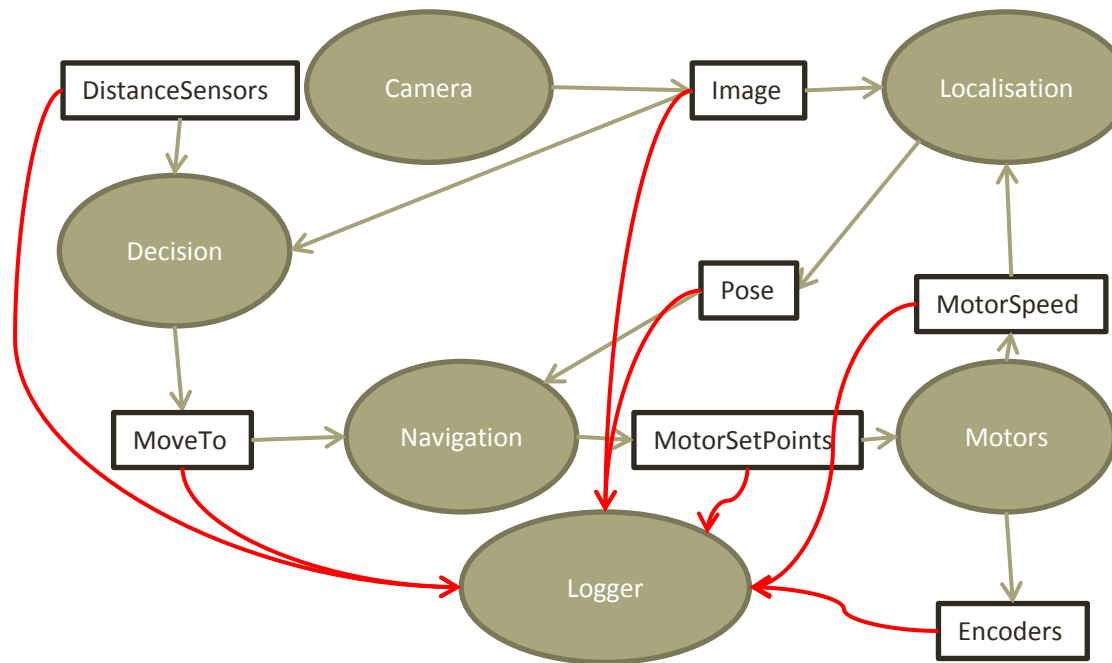


# Services

- Services are another way that nodes can communicate with each other.
  - More appropriate for **request / reply** interactions
    - Services allow nodes to send a **request** and receive a **response**.



# ROS BAGS



- A *bag* is a file format in ROS for storing ROS message data – named after their .bag extension
  - Tools provided:
    - **rosbag**: unified console tool for **recording, playback**, and other operations.
    - **rxbag**: **graphical** tool for **visualizing bag file data**.

[http://www.youtube.com/watch?v=pwlbArh\\_neU](http://www.youtube.com/watch?v=pwlbArh_neU)

APPLICATIONS DEVELOPED TO ROS

# The Office Marathon: Robust Navigation in an Indoor Office Environment

- Hardware Requirements
  - It is meant for both **differential** drive and **holonomic** wheeled robots only.
  - It assumes that the mobile base is controlled by sending desired velocity commands to achieve in the form of: **x velocity, y velocity, theta velocity**.
  - It requires a **planar laser mounted** somewhere on the mobile base. This laser is used for map building and localization.
- Navigation Stack was developed on a square robot
  - Performs best on **robots** that are **nearly square** or **circular**.
  - Robots of **arbitrary shapes** and sizes may have **difficulty** with large rectangular robots in **narrow spaces** like doorways.

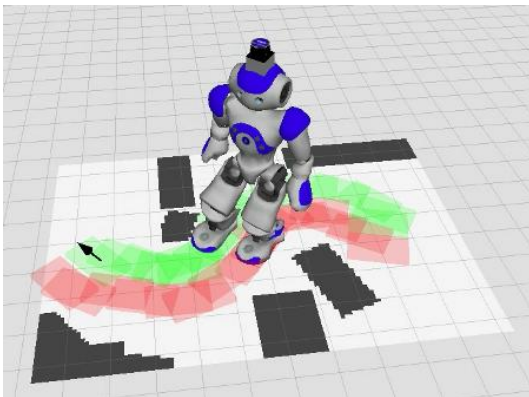
Paper: [http://www.ros.org/wiki/Papers/ICRA2010\\_Marder-Eppstein](http://www.ros.org/wiki/Papers/ICRA2010_Marder-Eppstein)

Code: <http://www.ros.org/wiki/navigation>

<http://www.youtube.com/watch?v=qziUJcUDfBc>

# Humanoid Navigation with Dynamic Footstep Plans

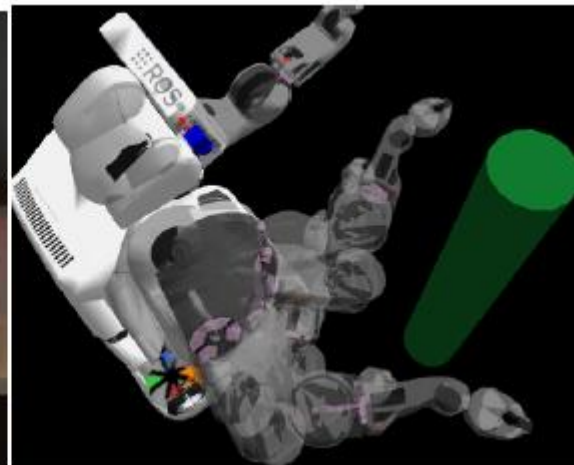
- **Footstep planning** to efficiently search for optimal paths in environments containing **planar obstacles using D\* Lite**
  - Reconfiguration of parameters to adjust to robot
    - Feet model
    - Movement heuristic
    - Etc.
  - Enables the robot to **reuse information** from previous searches when it has to revise its footstep plan according to changes in the environment.



<http://www.youtube.com/watch?v=o0rlrEHN1w4>

# STOMP: Stochastic Trajectory Optimization for Motion Planning

- Presents a new approach to motion planning using a stochastic trajectory optimization framework.
  - Relies on generating noisy trajectories to explore the space around an initial (possibly infeasible) trajectory to produce an updated trajectory with lower cost.
- **Available** as C++ source code (in ROS).



Paper: [http://www.ros.org/wiki/Papers/ICRA2011\\_Kalakrishnan](http://www.ros.org/wiki/Papers/ICRA2011_Kalakrishnan)  
Code: [http://www.ros.org/wiki/stomp\\_motion\\_planner](http://www.ros.org/wiki/stomp_motion_planner)  
[www.youtube.com/watch?v=KZqi8wAcC4k](http://www.youtube.com/watch?v=KZqi8wAcC4k)

ROS + Kinect

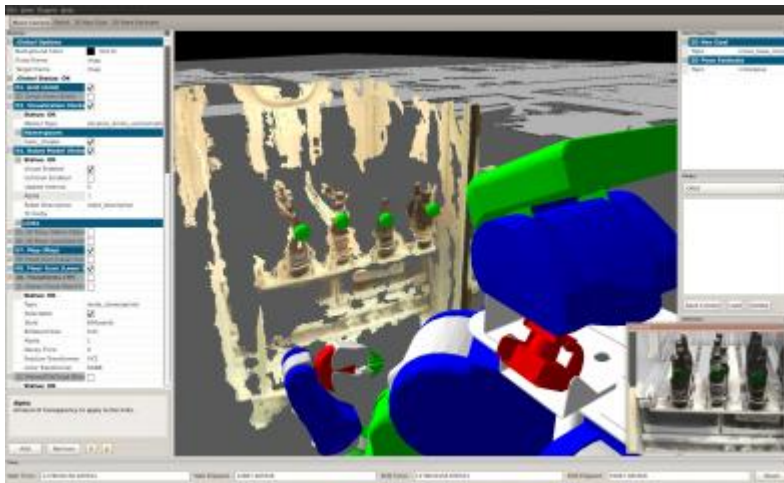
# WORK IN PROGRESS

<http://www.youtube.com/watch?v=rYUFu64VXkg>



# Beer Me, Robot

- <http://www.willowgarage.com/blog/2010/07/06/beer-me-robot>
- <http://www.youtube.com/watch?v=c3Cq0sy4TBs>



# Other Links

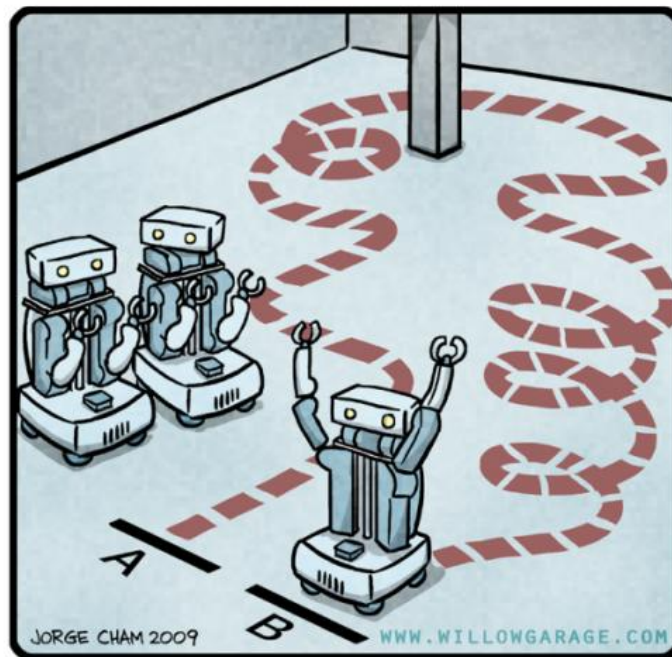
- <http://www.ros.org> – ROS webpage
  - Software
  - Tutorials
  - Publications, Courses, and Events
- <http://www.willowgarage.com/>  
<http://www.youtube.com/user/WillowGaragevideo>
  - Helped found and continue to contribute heavily to ROS
- <http://www.ros.org/wiki/Papers>
  - Papers that provide open source implementations through ROS repositories

# Conclusions

- The system
  - **Modular** (Nodes)
    - **Simpler** and more **reliable** development and deployment
    - Reusability of code
  - **Publish/Subscribe**
    - **Loosely-coupled** data sharing model
  - **Services**
    - **Complement** the data sharing model
- Availability
  - **Active project**
    - Several recent works
  - Open-source
    - **Free**

# Thank you

R.O.B.O.T. Comics



"HIS PATH-PLANNING MAY BE  
SUB-OPTIMAL, BUT IT'S GOT FLAIR."