ROS

Anthony Kyung Guzman Leguel

ViaLab

ROS

Robot Operating System

ROS

Why?

- Lack of standards
- Not to much code reusability
- Modularity

What is it used for?

- Facilitate algorithm implementation
- Easy to create and deploy

ROS System

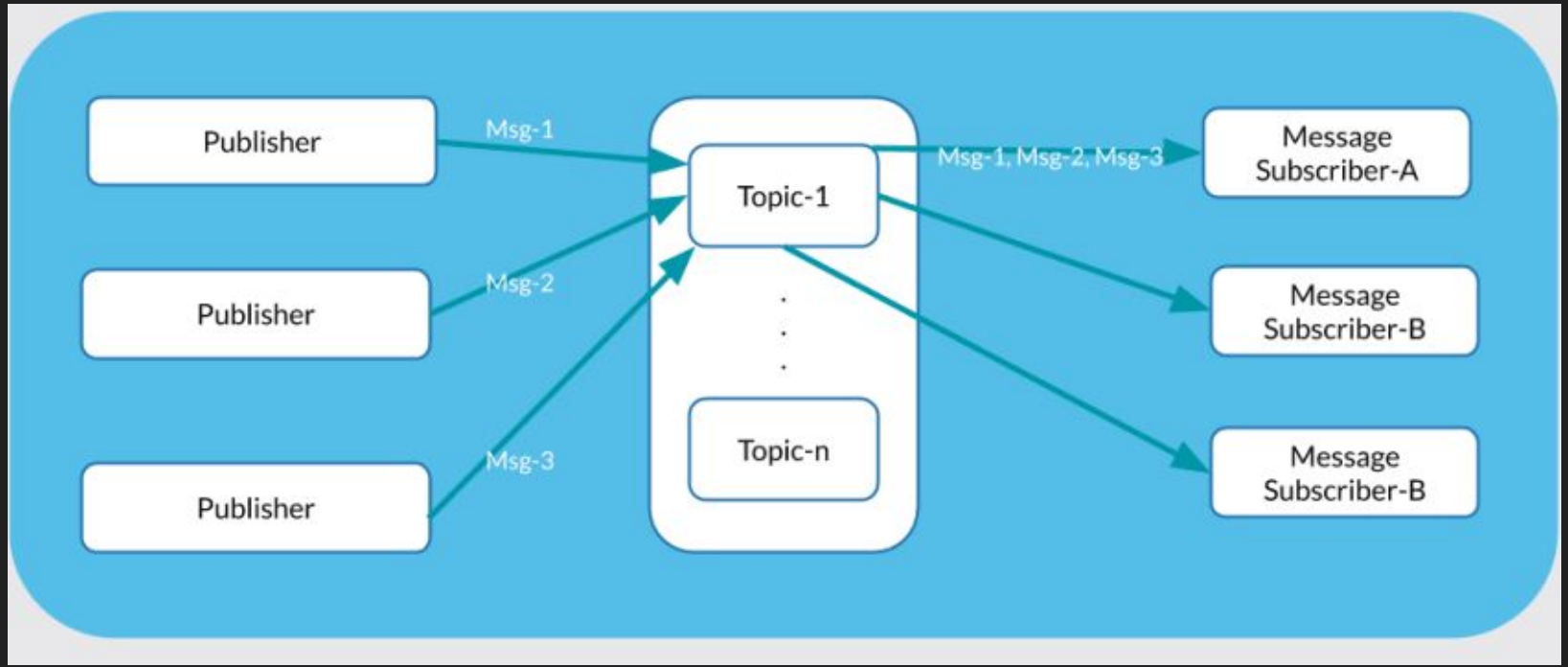

Image taken from: https://dev.to/ronnymedina/publish-subscribe-pattern-example-redis-kafka-1gd9

ROS System

Messages:

common_msgs: *actionlib_msgs* | *diagnostic_msgs* | *geometry_msgs* | *nav_msgs* | *sensor_msgs* | *shape_msgs* | *stereo_msgs* | *trajectory_msgs* | *visualization_msgs*

Topic:

/application_name/topic_name

E.g. /car/pose

Image taken from: http://wiki.ros.org/geometry_msgs?distro=noetic

E.g. geometry_msgs

ROS Message Types

- Accel
- AccelStamped
- AccelWithCovariance
- AccelWithCovarianceStamped
- Inertia
- InertiaStamped
- Point
- Point32
- PointStamped
- Polygon
- PolygonStamped
- Pose
- Pose2D
- PoseArray
- PoseStamped

geometry_msgs/PoseStamped Message

File: geometry_msgs/PoseStamped.msg

Raw Message Definition

```
# A Pose with reference coordinate frame and timestamp
Header header
Pose pose
```

Compact Message Definition

```
std_msgs/Header header
geometry_msgs/Pose pose
```

autogenerated on Wed, 02 Mar 2022 00:06:53

Images taken from: http://wiki.ros.org/geometry_msgs?distro=noetic

ROS System

Can also use services:

- Service request
- Receiving response

ROS File System

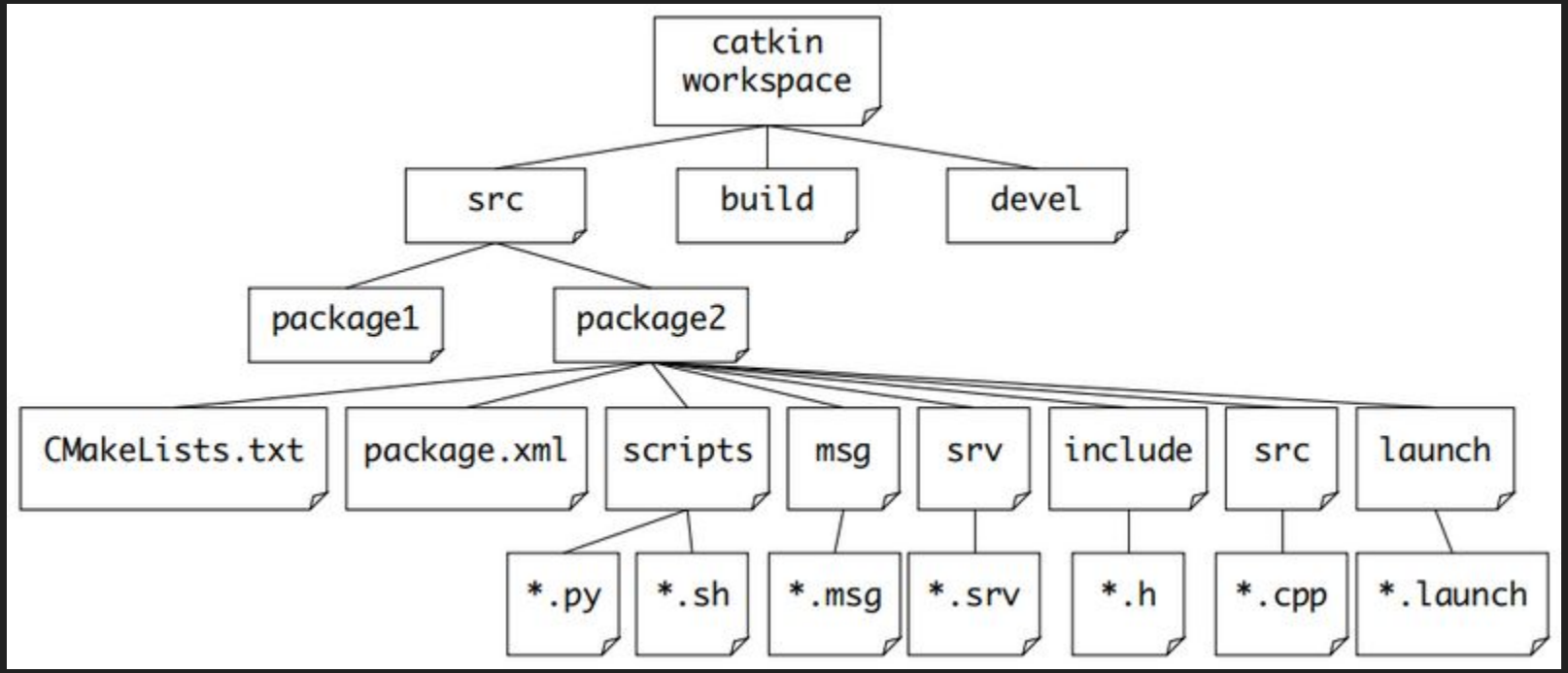

Image from: https://medium.com/swlh/7-simple-steps-to-create-and-build-our-first-ros-package-7e3080d36faa

ROS Packages

Create a package:

- catkin create pkg NAME [FLAGS]

Package minimum requirements:

- CMakelists.txt
- package.xml
- src

ROS Package CMakelists.txt

```
cmake_minimum_required(VERSION 3.0.2)
project(demo)

add_compile_options(-std=c++11)

find_package(catkin REQUIRED COMPONENTS
  roscpp
  rospy
  std_msgs
)


catkin_package(
#  INCLUDE_DIRS include
#  LIBRARIES demo
#  CATKIN_DEPENDS roscpp rospy st_msgs
#  DEPENDS system_lib
)

include_directories(
# include
  ${catkin_INCLUDE_DIRS}
)


add_executable(pub src/publisher.cpp)
target_link_libraries(pub ${catkin_LIBRARIES})
```

ROS Package package.xml

```xml
<?xml version="1.0"?>
<package format="2">
  <name>demo</name>
  <version>0.0.0</version>
  <description>The demo package</description>

  <maintainer email="anthony@todo.todo">anthony</maintainer>

  <license>TODO</license>

  <buildtool_depend>catkin</buildtool_depend>
  <build_depend>roscpp</build_depend>
  <build_depend>rospy</build_depend>
  <build_depend>stdd_msgs</build_depend>
  <build_export_depend>roscpp</build_export_depend>
  <build_export_depend>rospy</build_export_depend>
  <build_export_depend>std_msgs</build_export_depend>
  <exec_depend>roscpp</exec_depend>
  <exec_depend>rospy</exec_depend>
  <exec_depend>std_msgs</exec_depend>


  <!-- The export tag contains other, unspecified, tags -->
  <export>
    <!-- Other tools can request additional information be placed here -->

  </export>
</package>
```

ROS Package

Can have:

- include
- launch
- msg
- config

ROS Package launch .launch

```
<launch>
    <param name="use_sim_time" value="true"/>
    <node pkg="gmapping" type="slam_gmapping" name="slam_gmapping_2"
output="screen">
      <remap from="scan" to="/robot1/laser_1"/>
      <param name="sigma" value="0.05"/>
      <param name="base_frame" value="/robot1"/>
      <remap from="map" to="/gmapping/map2"/>
      <remap from="map_metadata" to="/gmapping/metadata2"/>
    </node>
</launch>
```

ROS Package config .yaml

```yaml
## this file correspond to aero parameters in real time experiments with D435 realsense camera

## Parameters for OOTP
# filter parameters
path_gains: [ -3.0104536742267722,-9.257906132819208,-10.574352237875011,-5.326766977660398, 1.5 ]
yaw_gains: [-3.0, -2.0]
path_error: 0.1
arc_velocity: 10.0
waypoint_lengh: 0.2
yaw_velocity: 2.0
yaw_error: 0.05
dynamics_iterations: 5
dynamics_step_size: 0.005 #0.1

# main params
agent_radius: 0.18
safe_radius_person: 0.1
max_lost_turns: 3
goal: [6.0, 0.5, 0.7]
```

ROS Build Node

At workspace:

- catkin build
- source devel/setup.bash

ROS Usage

1. roscore
2. rosrun package_name node_name

Alternative

1. roscore
2. roslaunch file_name.launch

ROS Commands

rosnode info /NAME_OF_NODE (shows info about node: publications (msg), subscriptions (msg), etc.)

rosnode list (shows the nodes that are running)

rqt_graph (shows the connections between nodes)

rostopic list (shows the lists of the topics being subscribed or published)

rostopic echo [topic] (shows the message being published)

rostopic hz (shows the node cycle)

rqt_console (pops a window that shows the ros debug messages)

rqt_logger_level (pops a window that can change the debug level)

rosbag record [topic] (records a bag of data of the topics of your choosing)

rqt_plot (plot data from messages)

DEMO

Homework

- Modify node control to get the control parameters from the .yaml file in config (adding control parameters) and publish the control signal in a geometry_msgs/Vector3Stamped
- Create actuator node and subscribe to the control signal
- Modify launch file to run the 4 nodes at the same time