10:30 **Industrial Robot Operating System (ROS)**
Geir Hovland, Professor og Senterleder
SFI Offshore Mechatronics
Universitetet i Agder

Knut Berg Kaldestad, UiA

SFI OFFSHORE MECHATRONICS

Industrial Robot Operating System

2016-03-17

E-mail: sfi@mechatronics.no
https://sfi.mechatronics.no

Agenda

1. Short Introduction to SFI Offshore Mechatronics
2. Mechatronics Innovation Lab (MIL)
3. Introduction and Short History of ROS
4. What is ROS (ros core, node, topics, service, tools)
5. Industrial ROS and Initial Setup at UiA
6. Summary and Conclusions

SFI OFFSHORE MECHATRONICS

Partners

Centre Coordinator:

University of Agder

Academic Partners:

Aalborg University

Aalesund University College

NTNU

RWTH Aachen

Research Institutes:

International Research Institute of Stavanger (IRIS)

Teknova

Clusters:

GCE | NODE GCE NODE

Companies:

ABB

Applica

Cameron

Devoteam

Klüber Lubrication

Lundin Norway

MacGregor

MHWirth

National Oilwell Varco

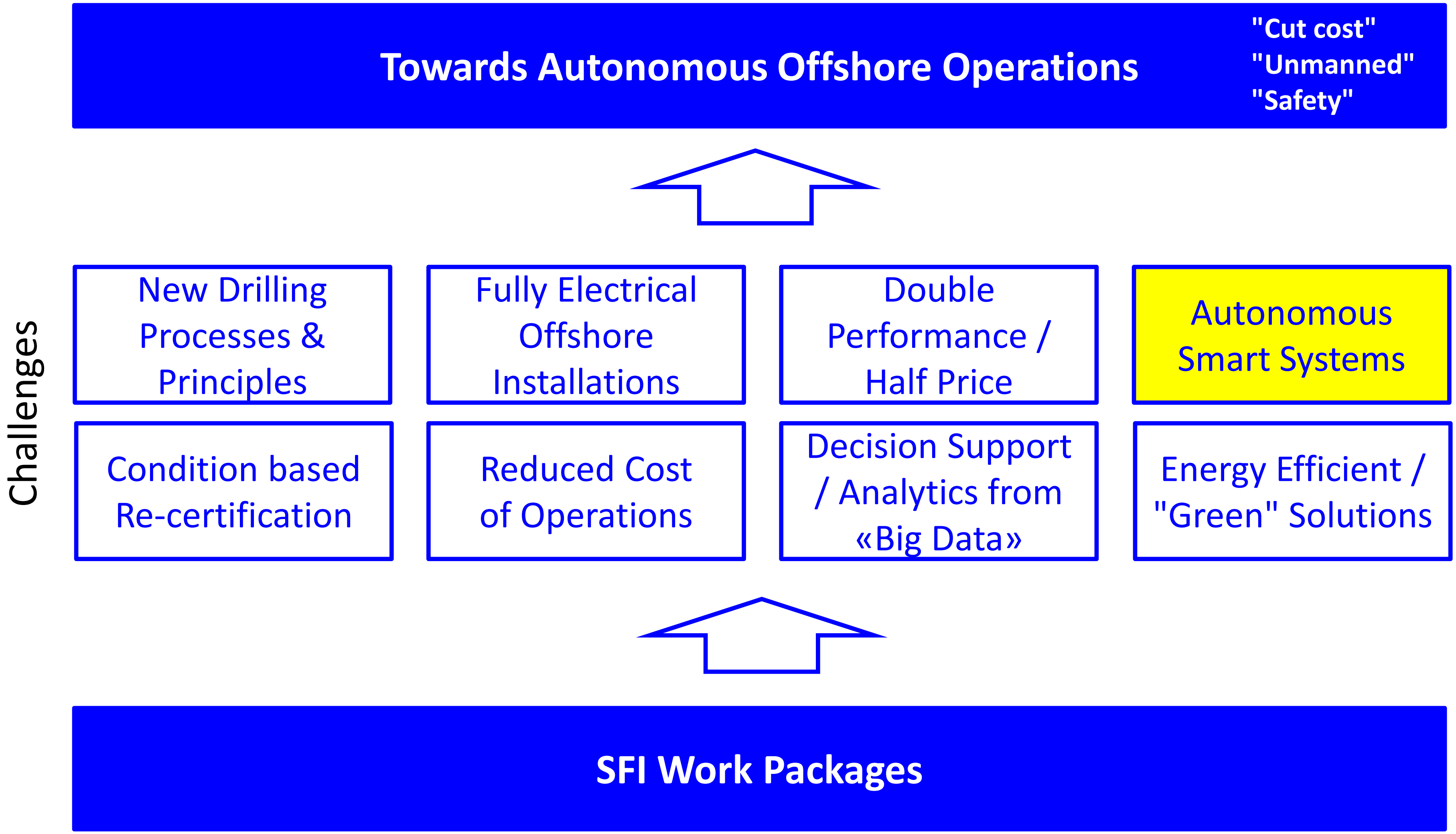
Skeie Technology

Stepchange

New Partner in 2016

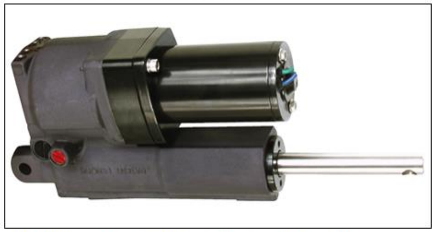
Rexroth Bosch Group

SFI "Grand Challenges"



SFI
OFFSHORE
MECHATRONICS

Parker, Compact EHA



http://www.parker.com/Literature/Oildyne/Oildyne%20-%20PDF%20Files/Compact-EHA-Catalog-HY22-3101E-7-13.pdf

Proposed Activities, 2016

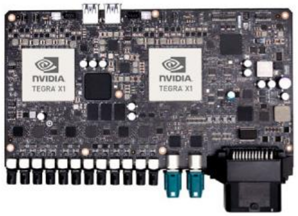
Color codes:
(PhD, UiA), (Post.Doc, UiA), (PhD, AAU), (PhD, NTNU), (Post.Doc, NTNU)
(PhD, RWTH Aachen), (Post.Doc, HiAls), (Researcher, Teknova), (GCENode)

WP1 Drives: Michael Rygaard Hansen

WP1.1 Secondary control in hydraulic systems
WP1.2 Using digital hydraulic in secondary control of motor drive
WP1.3 Using digital hydraulic in secondary control of cylinder drive
WP1.4 Electrical and electrohydraulic linear actuators (Candidate Cameron)

WP2 Motion Compensation: Olav Egeland

WP2.1 Computer vision and 3D sensors for topside automation of offshore drilling
WP2.2 High-performance control for motion compensation (Candidate NOV)
WP2.3 Nonlinear friction compensation in motion compensation systems with significant elasticity
WP2.4 Vision systems for offshore crane control in ship-to-ship operations
WP2.5 Real-time multiple DOF motion compensation using an industrial robot, sensor fusion and conformal geometric algebra
WP2.6 Optimal control for an offshore drilling rig (reformulation)



THE PLATFORM

Powered by NVIDIA's fastest SOCs and leveraging the same architecture as the world's most powerful supercomputers, DRIVE PX enables self-driving applications to be developed faster and more accurately. Key features of the platform include:

- Dual NVIDIA Tegra® X1 processors delivering a combined 2.3 Teraflops
- Interfaces for up to 12 cameras, radar, lidar, and ultrasonic sensors
- Rich middleware for graphics, computer vision, and deep learning
- Periodic software/OS updates

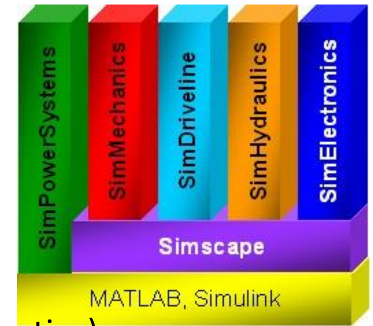
WP3 Robotics and Autonomy: Geir Hovland

WP3.1 Development of offshore 3D sensor package (Candidate NOV)
WP3.2 Autonomy systems foundation development (Candidate NOV)
WP3.3 Handling of sensor fusion, point-clouds and 3D maps
WP3.4 Implementation of situational awareness/human factors concepts for operators using virtual arena
WP3.5 Design and Verification Methods for Hybrid Control Systems

SFI OFFSHORE MECHATRONICS

SimulationX Powered by ITI

MODELICA



WP4 Modeling and Simulation: Olav Egeland

WP4.1 Integrated simulation of multi-physical systems in offshore operations (re-advertise)

WP4.2 Component-based simulation systems for drilling automation and crane systems (re-advertise)

WP4.3 Protocols and standard for integration of simulation models and co-simulation

WP5 Monitoring Techniques: Thomas Meyer

WP5.1 Tapered big bearings in top drive applications (Candidate MHWirth)

WP5.2 Large diameter steel ropes for subsea lifting applications

WP5.3 Fiber ropes for heave compensated subsea crane (re-advertise)

WP5.4 Condition-based lifetime prediction as result of calculated component loads

WP6 Data Anlytics, IT Integration and Big Data: Baltasar Beferull Lozano

WP6.1 Distributed in-network intelligence across multiple components

WP6.2 Design of architecture and self-organized cross-layer protocols for a heterogeneous wireless network platform

WP7 Technology Vision: Anne Grete Ellingsen

WP7.1 Technology vision

The Edvard Grieg platform has been successfully installed and commissioning is ongoing.

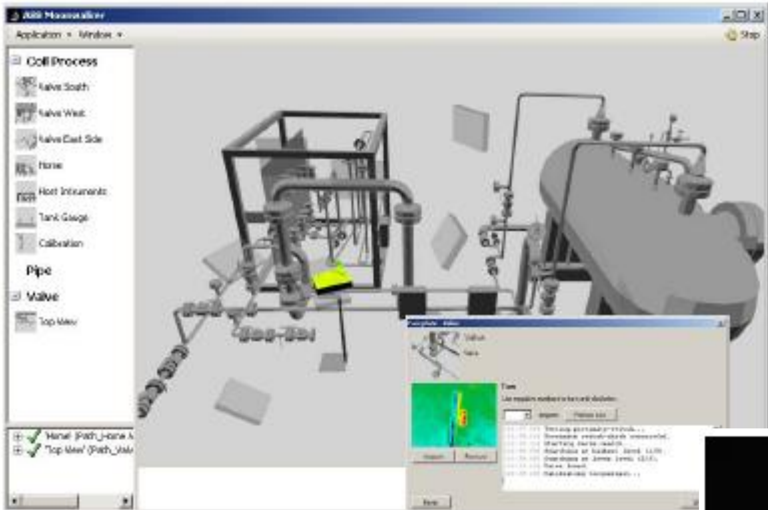
SFI OFFSHORE MECHATRONICS

Mechatronic Innovation Lab (MIL), to open Q3-2017. About 1500 sq.m.

- Outdoor robotic lab (from ABB. Previously used at Kårstø)
- Indoor robotic lab , 300 sq.m. (lab moved from ABB Oslo)

SFI OFFSHORE MECHATRONICS

Proceedings of the 2012 IFAC Workshop on Automatic Control in Offshore Oil and Gas Production, Norwegian University of Science and Technology, Trondheim, Norway, May 31 - June 1, 2012



A step-wise approach to oil and gas robotics ⋆

David A. Anisi * Charlotte Skourup *

* *Department of Technology & Innovation, Div. of Process Automation - Oil, Gas & Petrochemicals, ABB, Oslo, Norway*
{*david.anisi, charlotte.skourup*}*@no.abb.com*

(a) Inspection robot

(b) Valve manipulation tool

Fig. 2. The ABB indoor test facility in Oslo, Norway.

Why ROS ?

In a large consortium like SFI Offshore Mechatronics, there is a need for a common development framework for Autonomous Systems.

Company specific tools are good when there is one supplier, for example:

- RobotStudio from ABB
- 800xA from ABB
- TIA Portal from Siemens

In the SFI, ROS is a good candidate as the development platform:

- Lots of open source tools
- Allows for integration of company specific tools
- Does not get too close to company specific product development

SFI OFFSHORE MECHATRONICS

Scott Hassan: Software architect and developer of predeccesor to Google.

Willow Garage shut down in early 2014

Willow Garage History

Scott Hassan founded Willow Garage in late 2006 to accelerate the development of non-military robotics and advance open source robotics software. Named as an homage to the garages that spawned the computer industry, he envisioned an innovative research lab infused with the engineering expertise of a product development company. The goal was simple: to push the frontiers of robotics, both scientifically and commercially. Scott put together a funding package with the unique goals of impact first, return on capital second, with the strong belief that success in the first goal (impact) would provide plenty of opportunities to excel at the second (return on capital).

Highlights

Here are some highlights from our blog:

- August 21, 2013 - Willow Garage Employees Join Suitable Technologies
- April 18, 2011 - Introducing TurtleBot
- March 28, 2011 - Announcing PointClouds.org for the Point Cloud Library
- March 2, 2011 - ROS Distributions: Diamondback Release
- December 8, 2010 - Willow Garage joins PrimeSense to launch OpenNI
- November 8, 2010 - ROS Third Anniversary
- August 3, 2010 - ROS Distributions: C Turtle Release
- June 29, 2010 - Milestone 4 Complete, 11 PR2s Shipped
- May 4, 2010 - 11 Sites Named as PR2 Beta Program Recipients
- April 2, 2010 - Towel Folding PR2 (UC Berkeley)
- March 1, 2010 - ROS Distributions: Box Turtle Release
- March 1, 2010 - PR2 Beta Program Video
- February 1, 2010 - Lots of Texas Robots
- January 22, 2010 - Milestone 3 Complete
- January 22, 2010 - ROS 1.0 Released
- January 15, 2010 - PR2 Beta Unveiled, Call for Proposals
- November 09, 2009 - Texas Robot on ABC 7 News
- October 26,2009 - Texas Robot First Prototype
- September, 2009 - code.ros.org launched
- August 20, 2009 - ROS.org launched
- August 13, 2009 - Towards Milestone 3
- July 2,2009 - Milestone 2 Explained
- June 19,2009 - PR2 Sensor Head
- June 03,2009 - Milestone 2 Reached
- April 22,2009 - PR2's fine motor skills
- February 10, 2009 - Lots of ROS Repositories
- February 10,2009 - ROS 0.4 released (first stable release)
- December 12,2008 - Milestone 1 Reached
- November 11,2008 - PR2 Alphas Have Arms
- October 17, 2008 - First PR2 Alpha Pictures

SFI OFFSHORE MECHATRONICS

History of ROS

How we got here

ROS began life as the development environment for the Willow Garage PR2 robot. Our primary goal was to provide the software tools that users would need to undertake novel research and development projects with the PR2. At the same time, we knew that the PR2 would not be the only, or even the most important robot in the world, and we wanted ROS to be useful on other robots. So we put a lot of effort into defining levels of abstraction (usually through message interfaces) that would allow much of the software to be reused elsewhere.

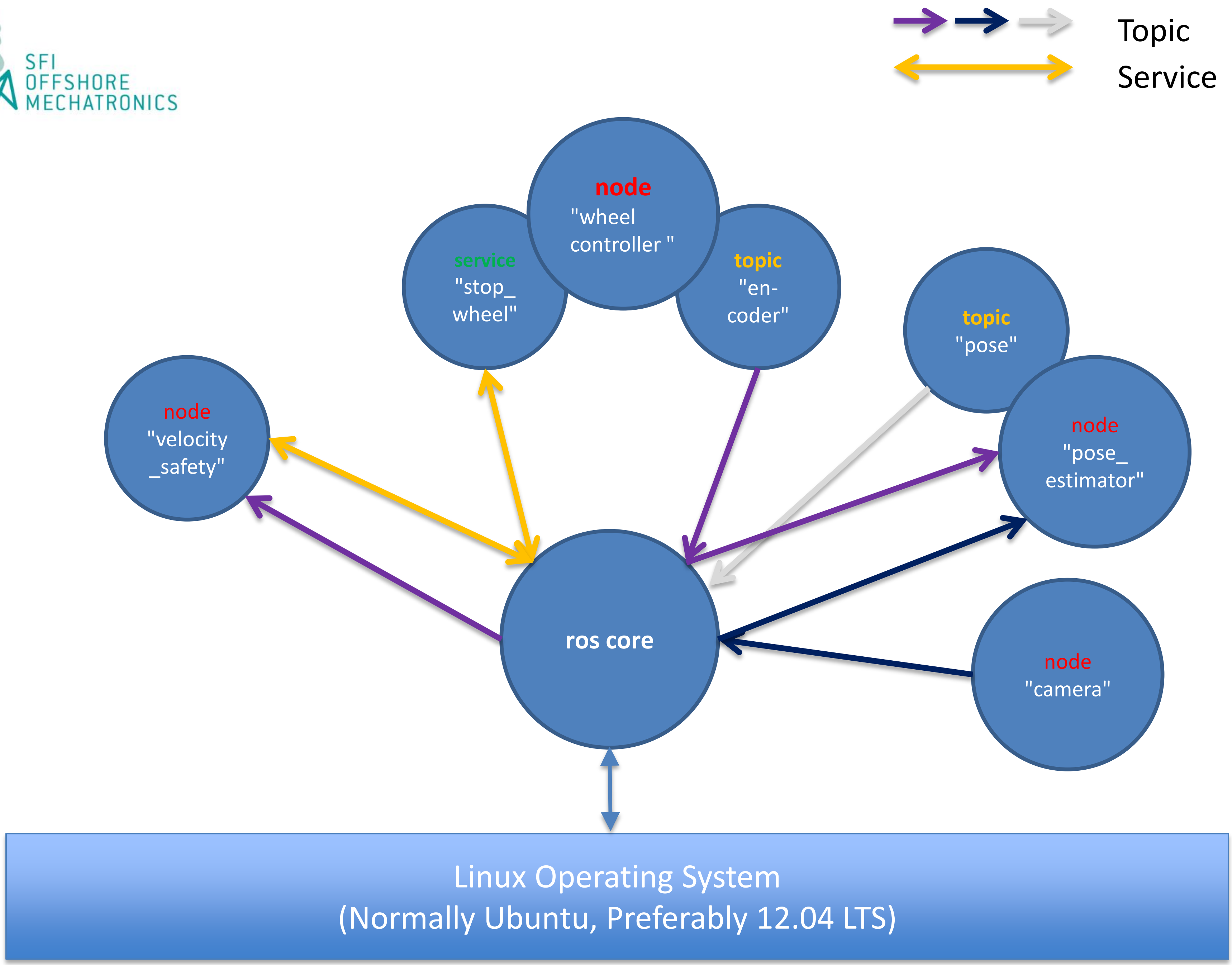
Still, we were guided by the PR2 use case, the salient characteristics of which included:

- a single robot;
- workstation-class computational resources on board;
- no real-time requirements (or, any real-time requirements would be met in a special-purpose manner);
- excellent network connectivity (either wired or close-proximity high-bandwidth wireless);
- applications in research, mostly academia; and
- maximum flexibility, with nothing prescribed or proscribed (e.g., "we don't wrap your main()").

It is fair to say that ROS satisfied the PR2 use case, but also overshot by becoming useful on a surprisingly wide variety of robots. Today we see ROS used not only on the PR2 and robots that are similar the PR2, but also on wheeled robots of all sizes, legged humanoids, industrial arms, outdoor ground vehicles (including self-driving cars), aerial vehicles, surface vehicles, and more.

In addition, we are seeing ROS adoption in domains beyond the mostly academic research community that was our initial focus. ROS-based products are coming to market, including manufacturing robots, agricultural robots, commercial cleaning robots, and others. Government agencies are also looking more closely at ROS for use in their fielded systems; e.g., NASA is expected to be running ROS on the Robonaut 2 that is deployed to the International Space Station.

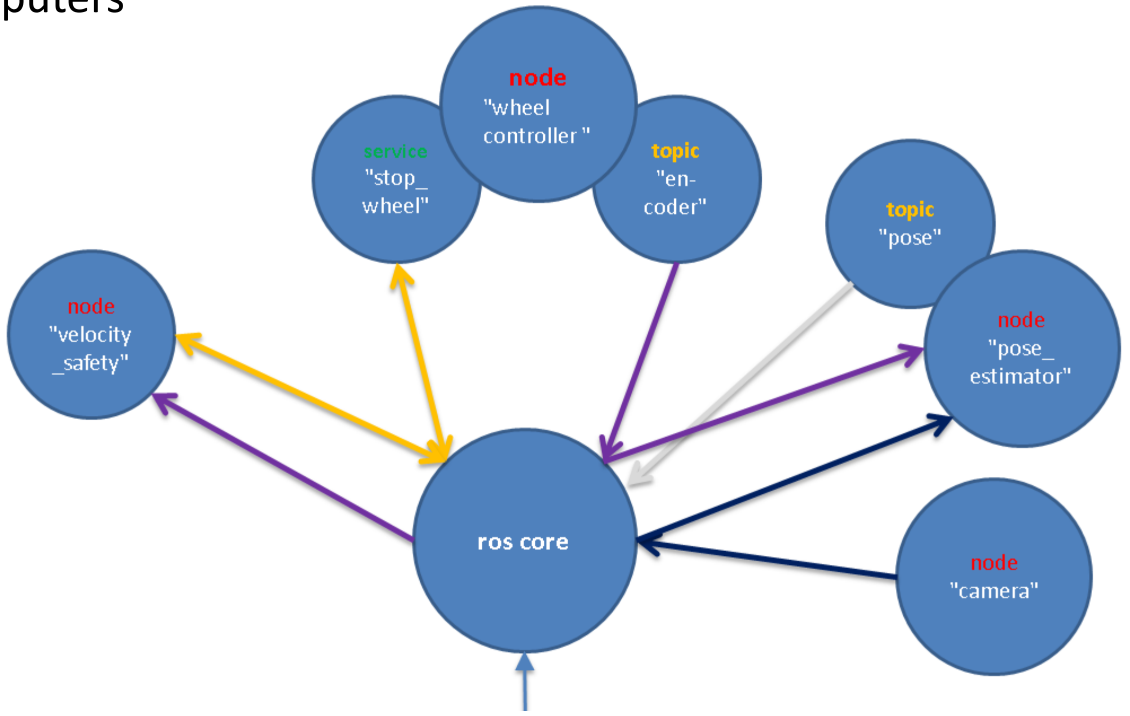
SFI OFFSHORE MECHATRONICS



SFI OFFSHORE MECHATRONICS

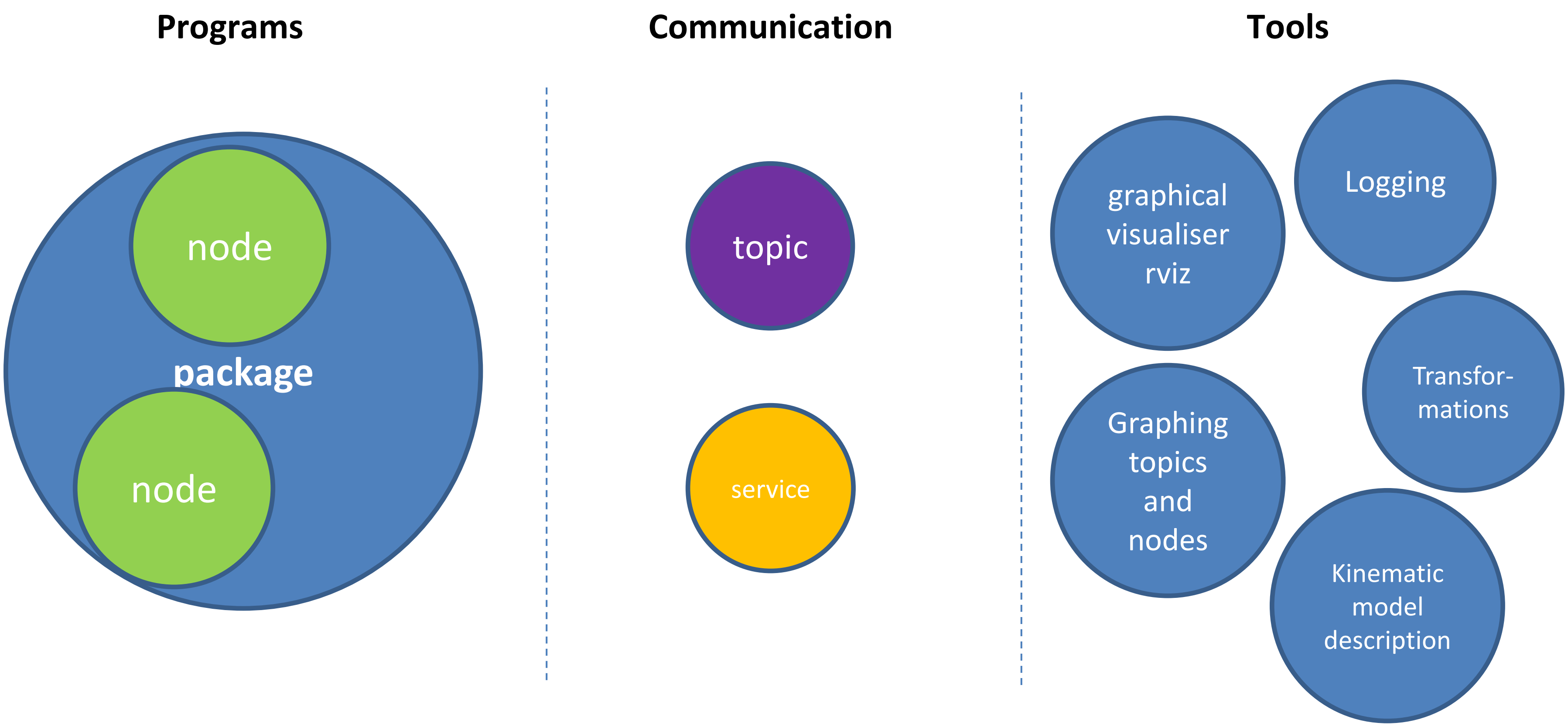
Benefits of ROS:

- ROS core architecture enables collaborative environment
- ROS is driven by an open source community
- A feature (node/executable program) developed by someone can easily be used by others
- The system can still operate if a node is shut down
- On the fly shutdown and startup of nodes
- For example, a safety node can be updated / recompiled without affecting other nodes
- Nodes can be on separate computers



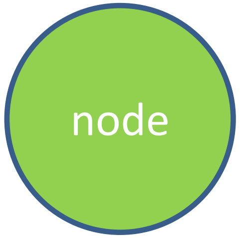
Programs, Communication and Tools

- «topic» is for example a stream of sensor data
- «service» is a callable function, for example «stop wheels»
- «tools» provide basic or advanced functionality which do not have to be nodes



Robot Operating System (ROS) – Node

Node



- A node is an executable program
- It interracts with other nodes through ros core
- Typically a node is only responsible for one part of the whole system

Package

- A unit containing relevant project data
  - Source code, models, compilation files
- A collection of zero or more nodes

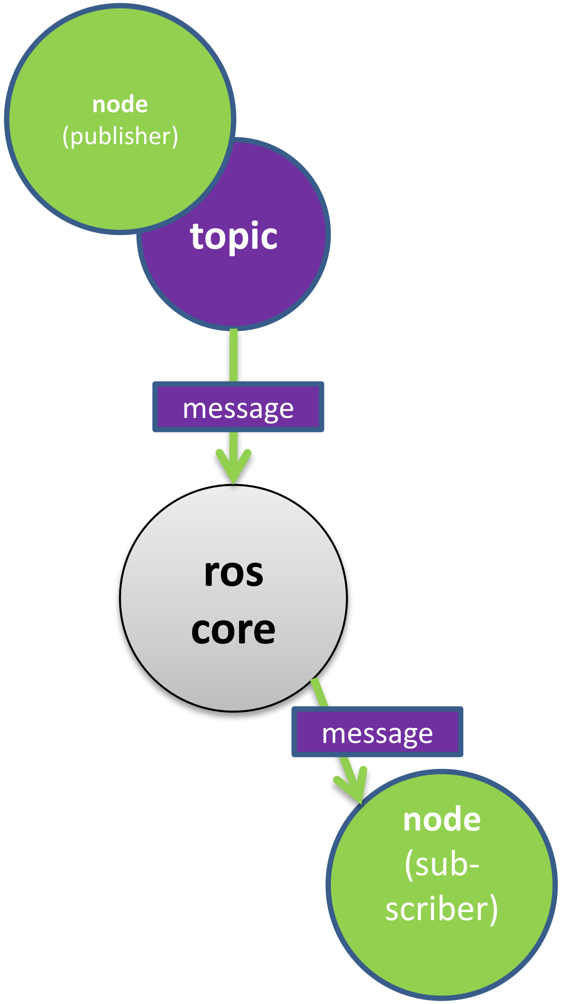
Robot Operating System (ROS) – Topic

Topic

- In ros, a node has two ways to communicate,

one is by using a topic

- Publish or subscribe to messages
- It is a «channel» a message is passed through
- Used for streaming data: camera, encoder, position



Message

- A message passed on the topic
- It defines the structure of the data
  - An example:

string description

float32 encoder1

float32 encoder2

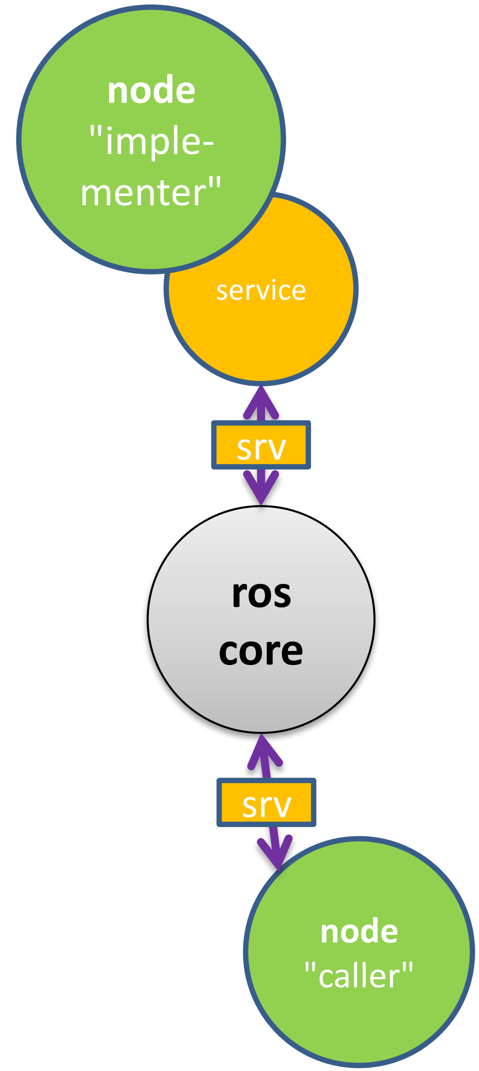
Robot Operating System (ROS) – Service

Service

- The second way for node communication

is implementing/using a service

- Used for calling a method on a node, request/response
- Can return void or data



srv

- Similar to the *message* type, but can also return data
- Example:

    bool stopRobot

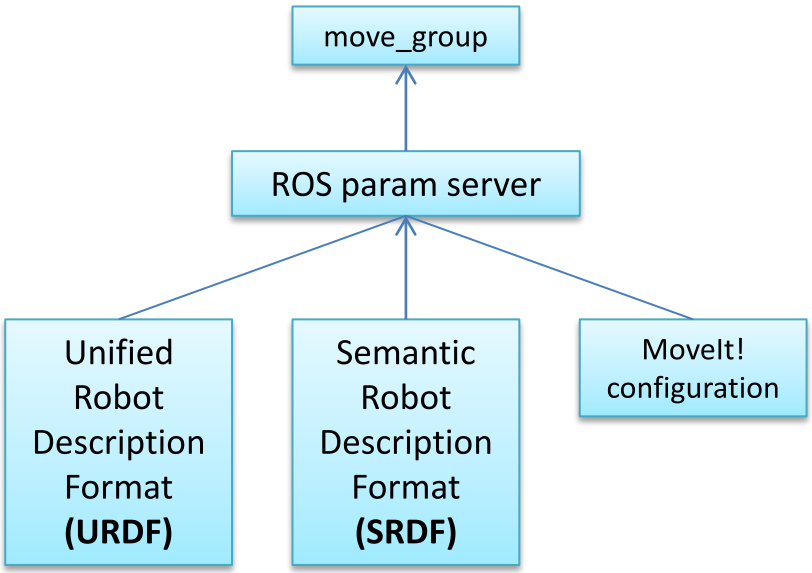
    ---

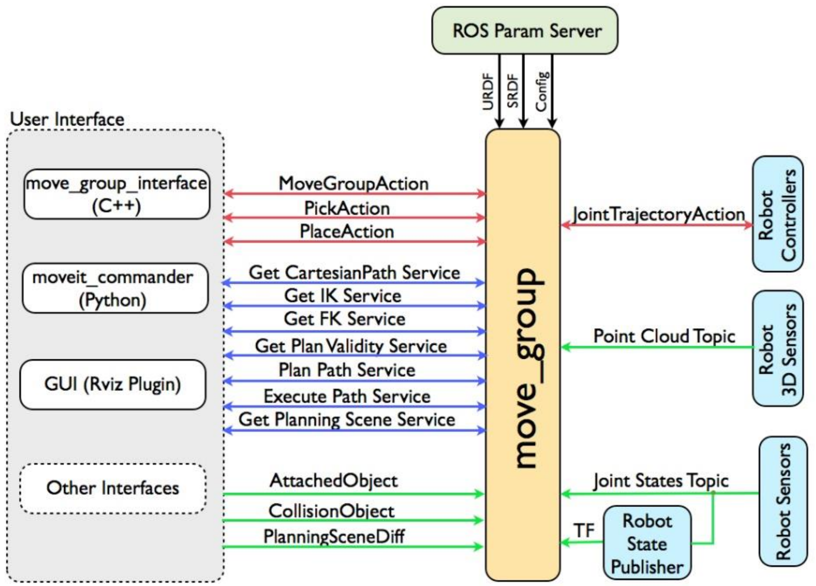
    bool robotStopped

Robot Operating System (ROS) – MoveIt!

Provides high-level system architecture

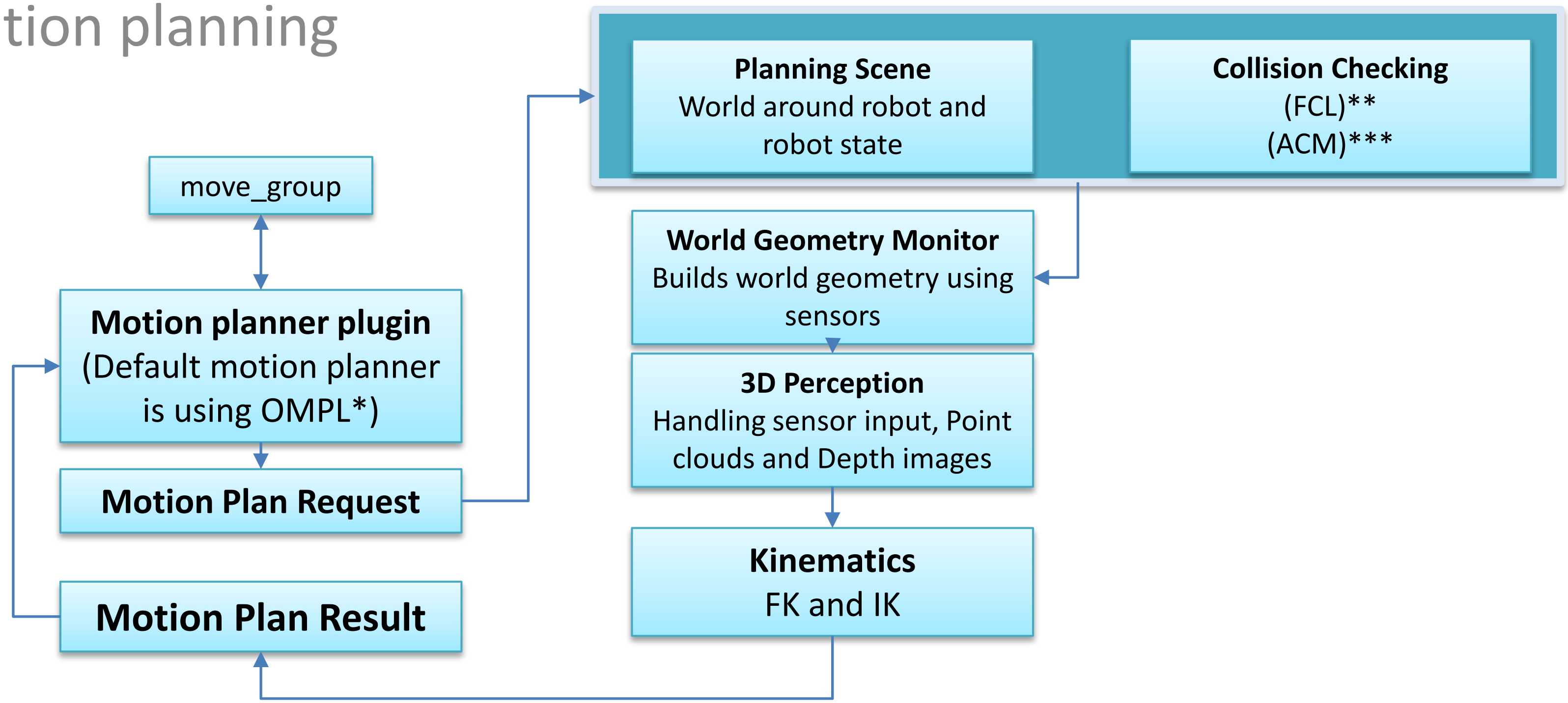
ROS Industrial depends on it.





Robot Operating System (ROS) – MoveIt!

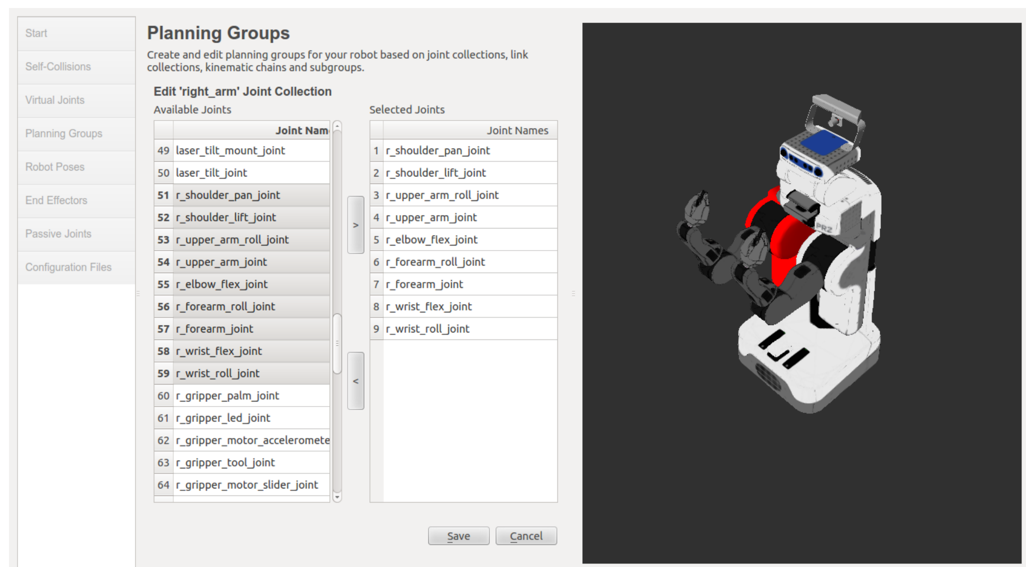
Motion planning



* OMPL - Open Motion Planning Library
** FCL – Flexible Collision Library, primary collision checking library for MoveIt!
*** ACM - Allowed Collision Matrix. A way to avoid the expensive collision checking for objects which are e.g. out of range.

Robot Operating System (ROS) – MoveIt!

- Configuring the robot using MoveIt! Setup Assistant
- Generate Self-Collision Matrix (disables links that can not collide)
- Defining virtual joints for primarily attaching robot to world
- Defining planning groups
  - Arm, leg, gripper etc.
  - Set kinematic solver
- Add poses, e.g. home position
- Add End Effectors
- Add Passive Joints
  - Example caster wheel
- Generate Configuration Files



SFI OFFSHORE MECHATRONICS

Robot Operating System (ROS) – ROS Industrial (ROS-I)

The two package types

1. **Vendor specific**

Packages targeting a particular vendor

Currently supports*: ABB, Fanuc, Motoman, Universal Robot

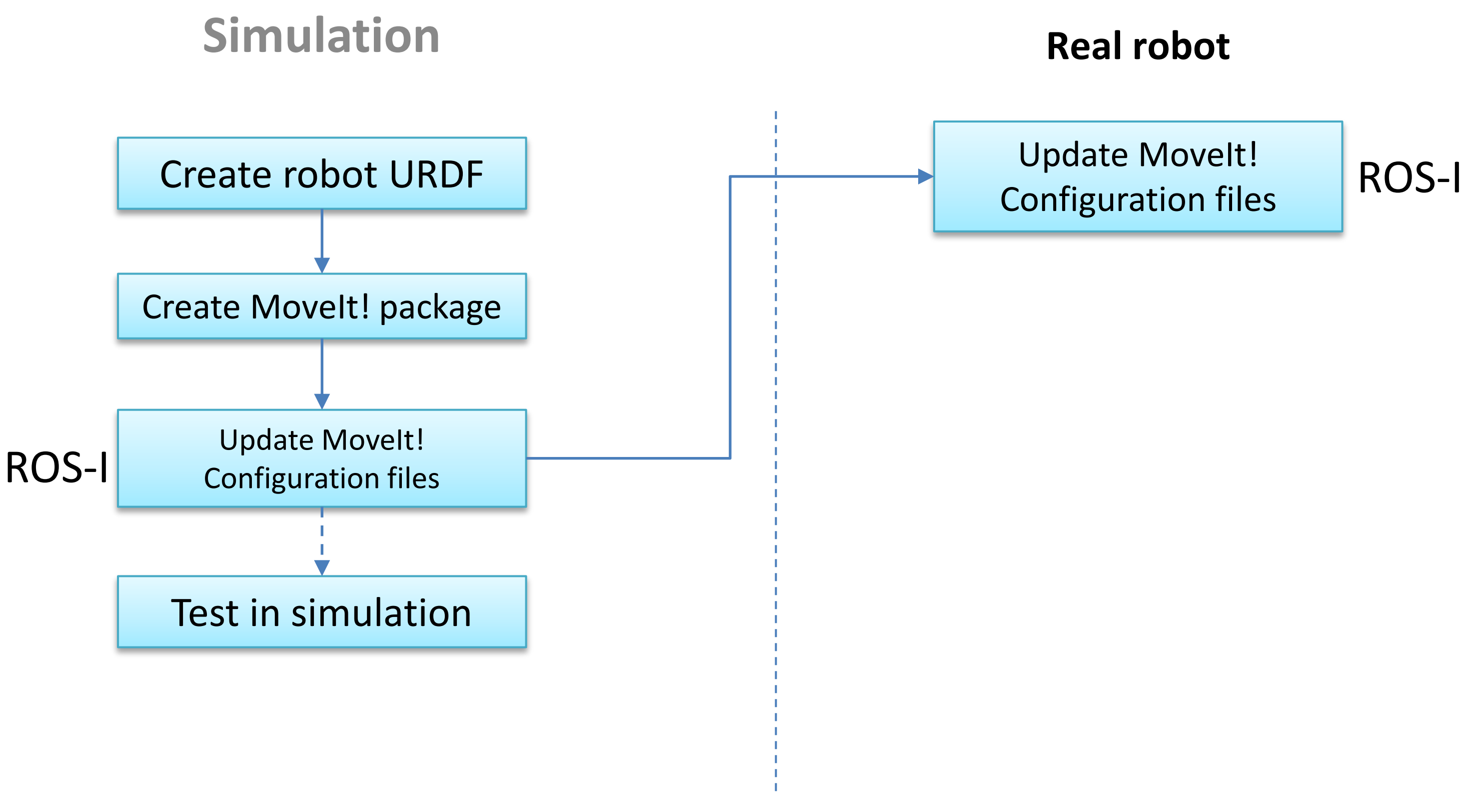
Can be used with no or minor modifications

2. **General**

Creation (or partial creation) of a ROS-I package from scratch

* non-exhaustive list

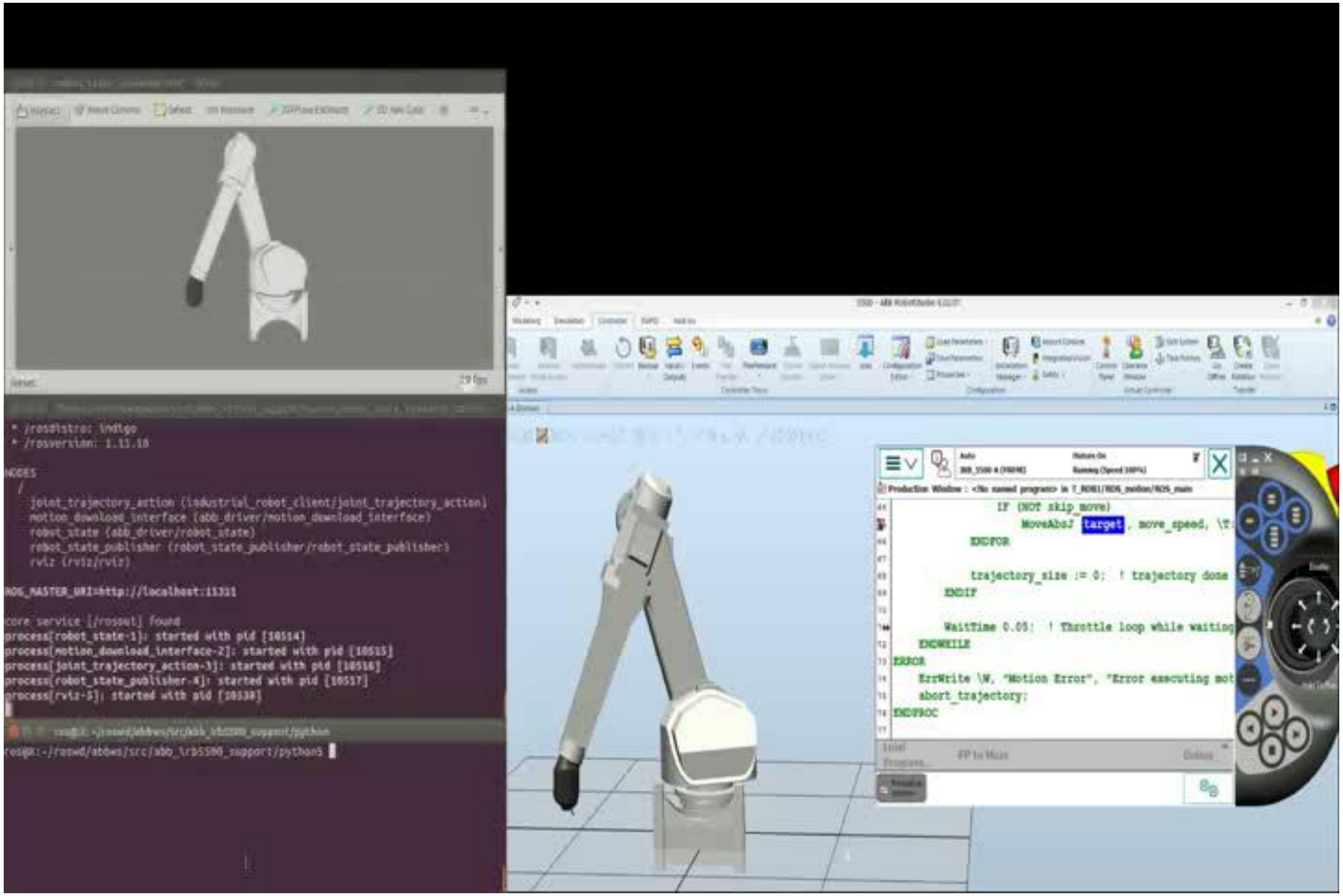
Robot Operating System (ROS) – ROS-I, General setup



* non-exhaustive list

SFI OFFSHORE MECHATRONICS

Video: Initial Results (Interface to RobotStudio)



Video: Initial Results (Interface to Real Robot)

Summary and Conclusions

- Industrial ROS seems like a good integration platform for the SFI
- Allows for connections to:
  - Company specific simulation software (eg. RobotStudio)
  - Company specific hardware (eg. ABB robots, Siemens PLCs, …)
- Open source software (eg. Point cloud library for 3D sensors)
- Allows for development between distributed research groups
  - UiA (Grimstad)
  - NTNU Aalesund
  - NTNU Trondheim