An introduction to ROS, The Robot Operating System

Seminar 150505, Department of Signals and Systems

Ashfaq Farooqui
ashfaqf@student.chalmers.se
Martin Viktorsson
viktorss@student.chalmers.se

Agenda

- Overview of ROS
- Basic ROS concepts and its capabilities
- How we are using ROS, and why?

Biography

- Ashfaq Farooqui
  - Systems Control & Mechatronics, Chalmers
  - Chalmers Formula Student 2014
  - B.Sc. Electronics and communication engineering, BMS Institute of Technology, India

- Martin Viktorsson
  - Systems Control & Mechatronics, Chalmers
  - Exchange studies at TU München, Germany
  - B.Sc Automation & Mechatronics, Chalmers

Motivation for ROS

- Distributed systems
  - Internet of Things (IoT)
- Modularity
  - Independent development of different applications
- ROS in research
  - No invention of the wheel again
  - Re-usability of modules and tools for debugging and visualization
  - Test on both hardware and simulator with minimal code changes.

ROS- Background

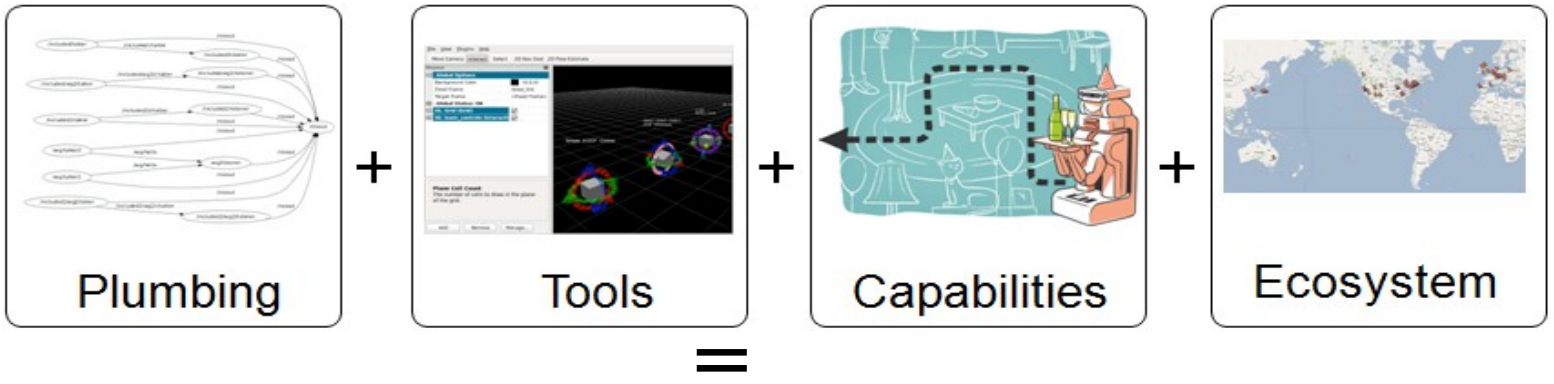
- STAIR – Stanford AI Robot
  - ~2005, In house prototypes of robots and systems
- Willow Garage – Founder of ROS
  - 2007, Personal Robotics project, iterations of framework design with Stanford
  - 2009, ROS 0.4 and PR2 Alpha
  - 2010, 11 PR2's delivered to industry and Universities.
  - 2013, ROS moves to Open Source Robotics Foundation

PR2

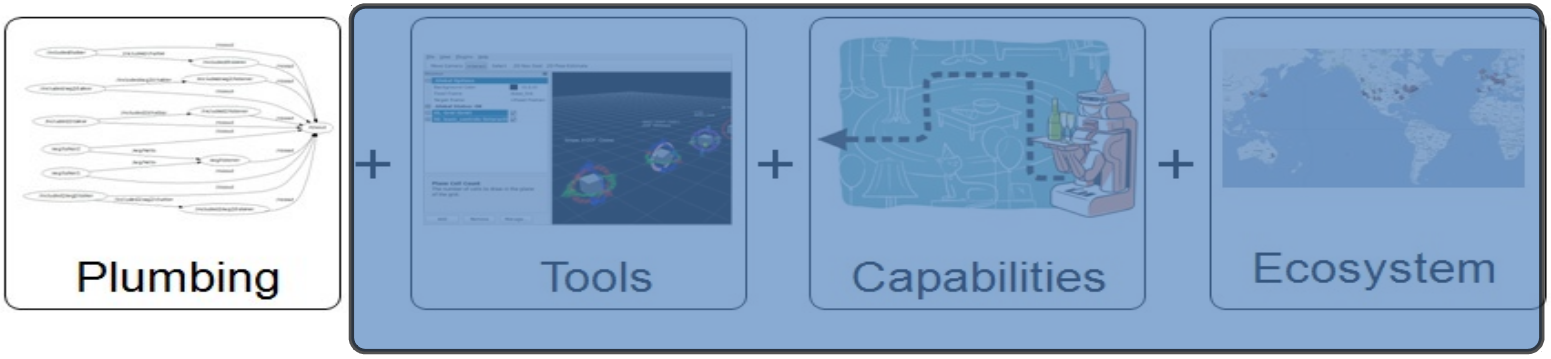
Source:http://www.flickr.com/photos/sixteenmilesofstring/5749414416/in/photostream/

- $280 000 (with 30% Open Source discount)

ROS: Robot Operating System



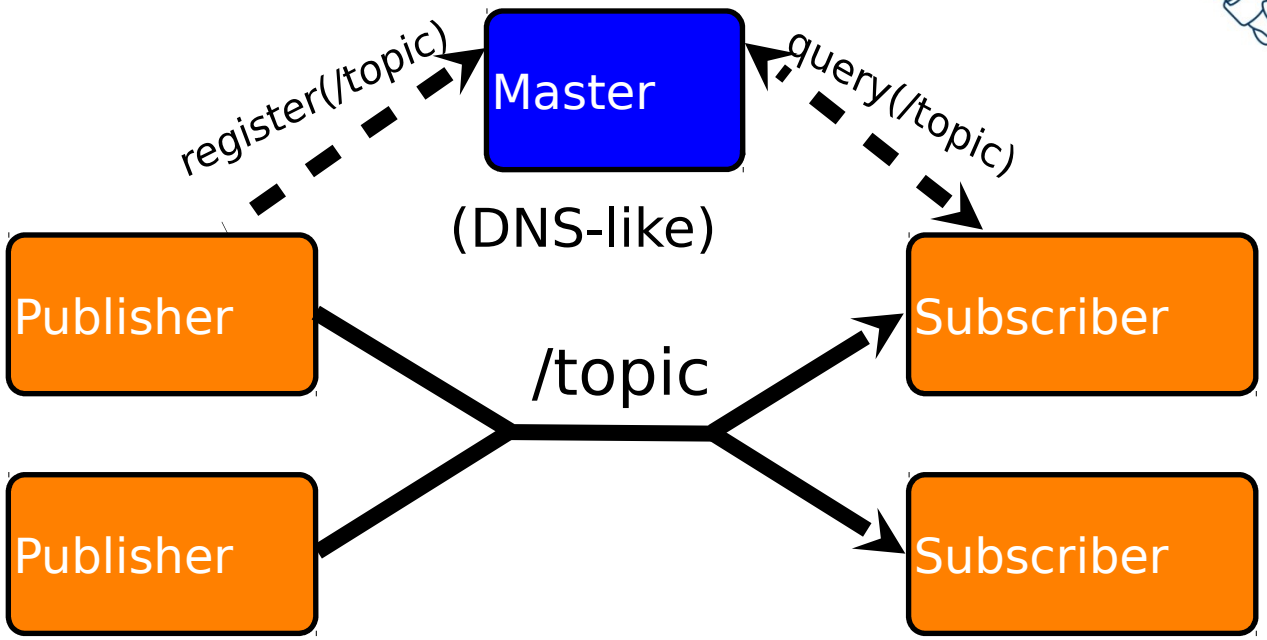
ROS: Plumbing



ROS provides publish-subscribe messaging infrastructure designed to support the quick and easy construction of distributed computing systems.

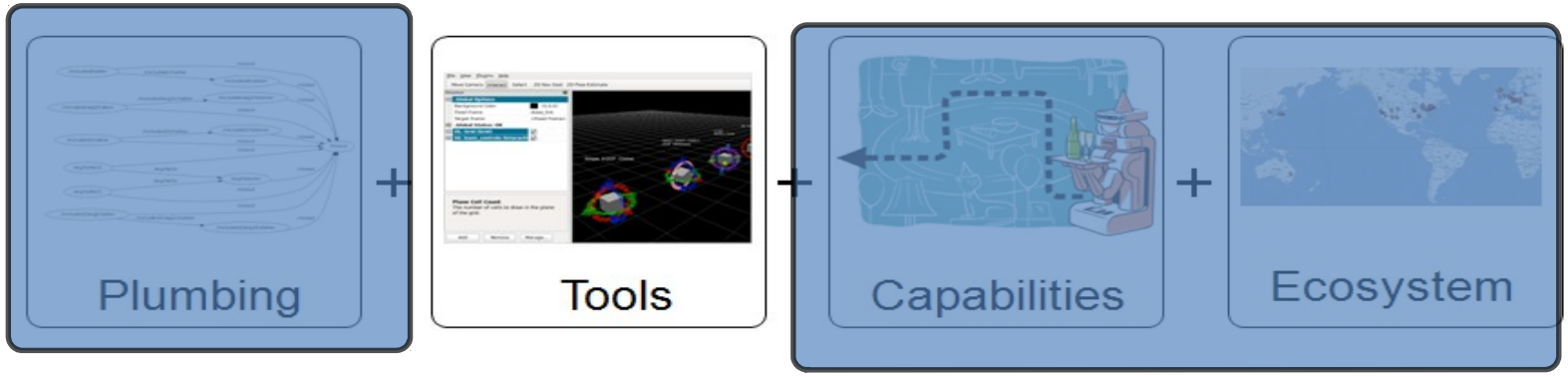
- Messages
- Topics
- Services
- Actions
- Parameters
- Nodes
- Roscore

Publish/Subscribe Messaging



- Simplifies complex programming
- Dynamic architectures allows reuse

ROS: Tools

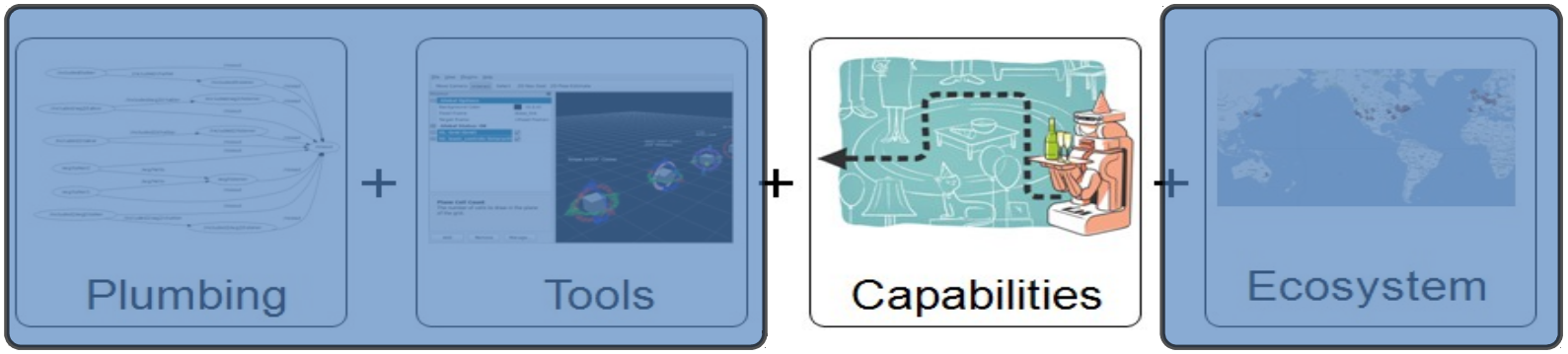


ROS provides an extensive set of tools for configuring, starting, introspecting, debugging, visualizing, logging, testing, and stopping distributed computing systems.

ROS: Tools

- Commandline ROS tools
  - Running the ROS systems
    - Rosrun
    - Roslaunch
    - Roscore
  - Interfacing and debugging running systems
    - Rosservice
    - Rostopic
    - Rosparam
    - Rosnode...
  - Install and build tools
    - catkin_make
    - Rosbuild
    - Roscd...
- System visualization and Simulation
  - Rviz
  - Gazebo
- Data Visualization tools
  - rqt_graph
  - rqt_plot
  - Rosbag: rqt_bag....
- Development tools
  - Standard linux tools
  - Boost,cpp,python,lisp, XMLRPC
  - GUI tools....

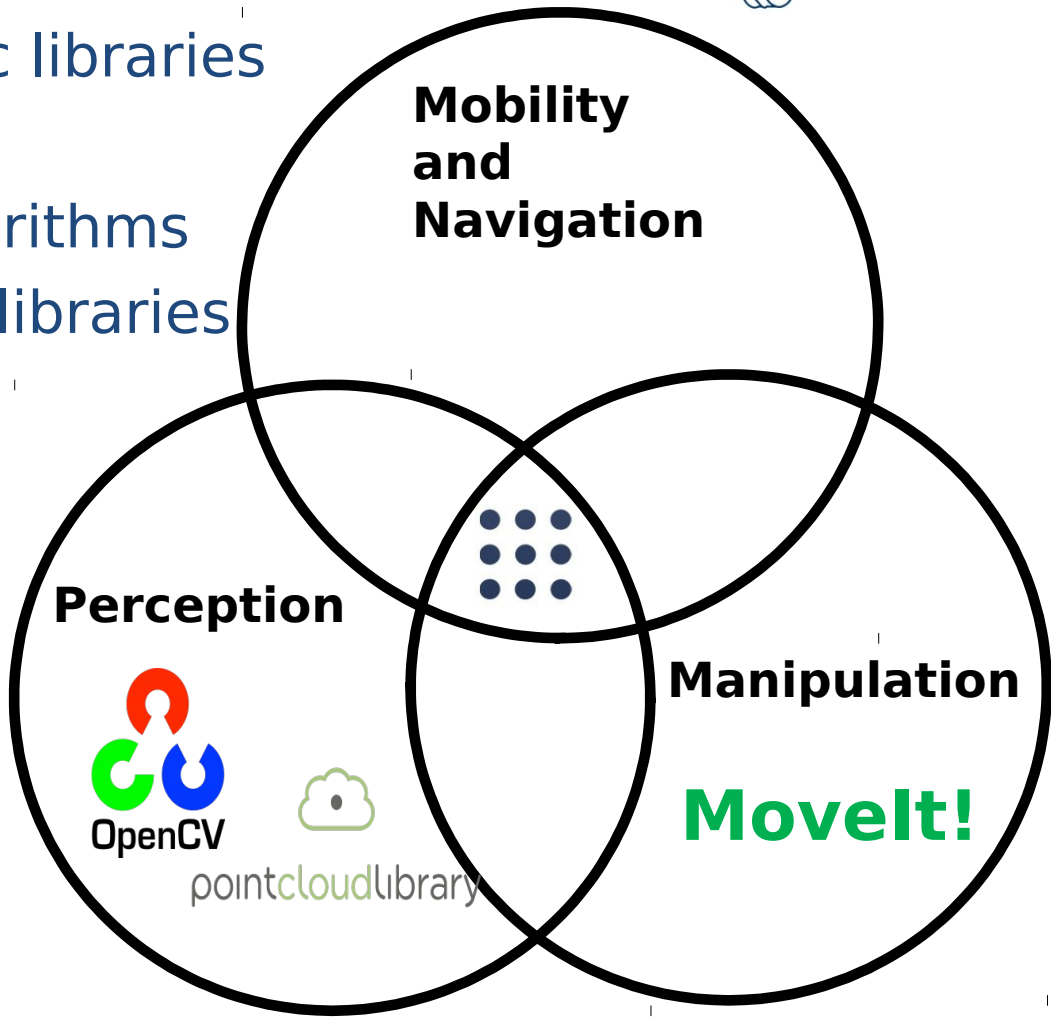
ROS: Capabilities



ROS provides a broad collection of libraries that implement useful robot functionality, with a focus on mobility, manipulation, and perception.

ROS: Capabilities

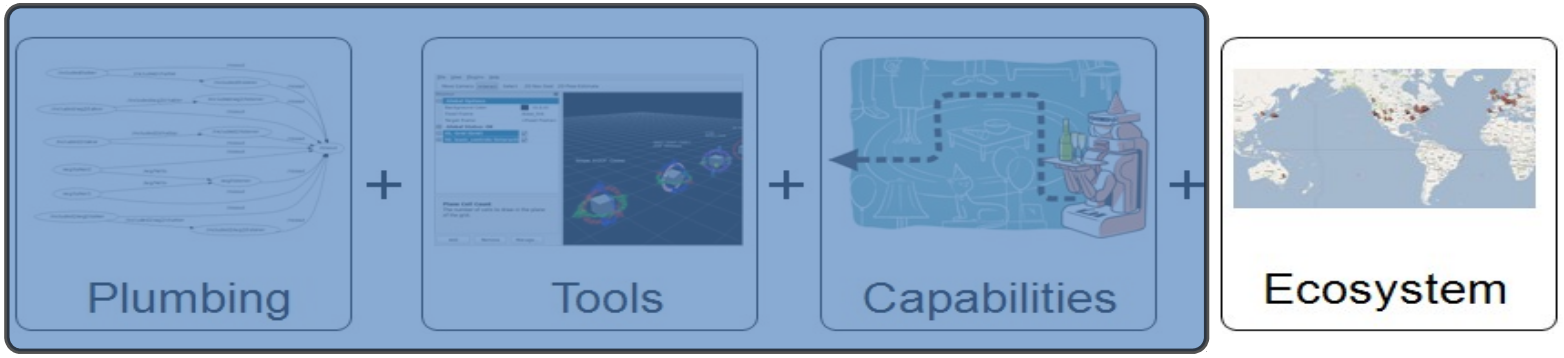
- Provide robot agnostic libraries
- Integration of various technologies and algorithms
- Possibility of reuse of libraries
- Rapid development from community



ROS: Capabilities

Video

ROS: Ecosystem



ROS is supported and improved by a large community, with a strong focus on integration and documentation. “ros.org“ is a one-stop-shop for finding and learning about the thousands of ROS packages that are available from developers around the world.

ROS-Industrial

ROS-I: Motivation

- Motivated by desire to solve hard problems encountered in industrial robotics and automation
- Driven by actual application needs (i.e. real life problems without commercial solutions)
  - Dynamic pick and place
  - Flexible automation (many small & diverse part runs)
  - Sensor driven automation
- Reduction in integration cost by standardizing interfaces and enabling reuse

ROS-I: Motivation

Industrial Robotics

- Limited by platform specific tools
  - Company provided hardware/software tools
  - Robot specific capabilities
  - Small user base
- Slow technology adoption
- Barriers to transition basic research
- Does not support independent developers

ROS-Industrial solves these

ROS-I: Supported Hardware

- Robots:
  - Universal Robot
  - Kuka
  - FANUC
  - ABB
  - Motoman
  - Adept
  - More to come....
- External Interfaces
  - SICK Sensors
  - Robotiq grippers
  - EtherCAT
  - USB
  - Ethernet

ROS: Limitations

- Guided by PR2 use-case
  - Single Robot design
  - Workstation class computation capabilities
  - Assumes Excellent network connectivity
- Not realtime
- Non-deterministic TCP protocol
- No Safety Standards compliance

ROS 2.0

Aims:

- Teams of multiple robots
- Be inclusive of small embedded platforms
- Inherently be a Real-time system
- Proscribed patterns for development
- Use of new technologies
  - DDS(Data distribution service)
  - Websockets
  - Protobuf
  - ZeroMQ-Serializer & Deserializer
- Provide middleware API interface for agnostic DDS
- UI based code generators

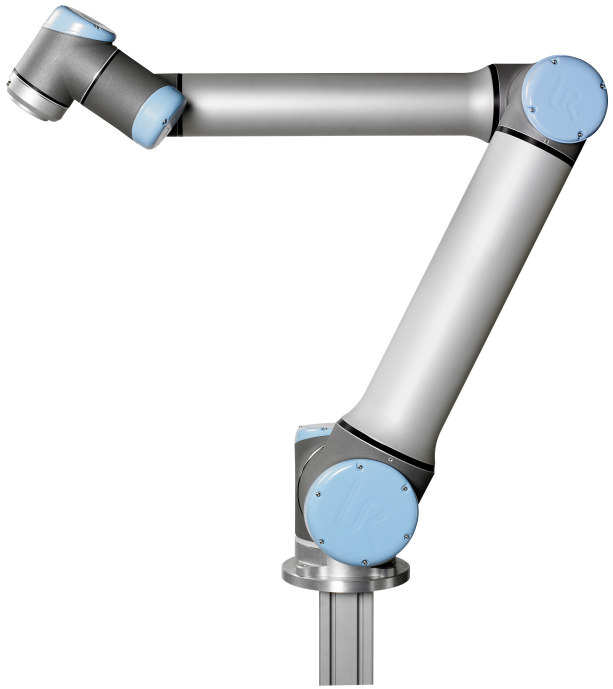
Our Implementation

Two theses, One Project!

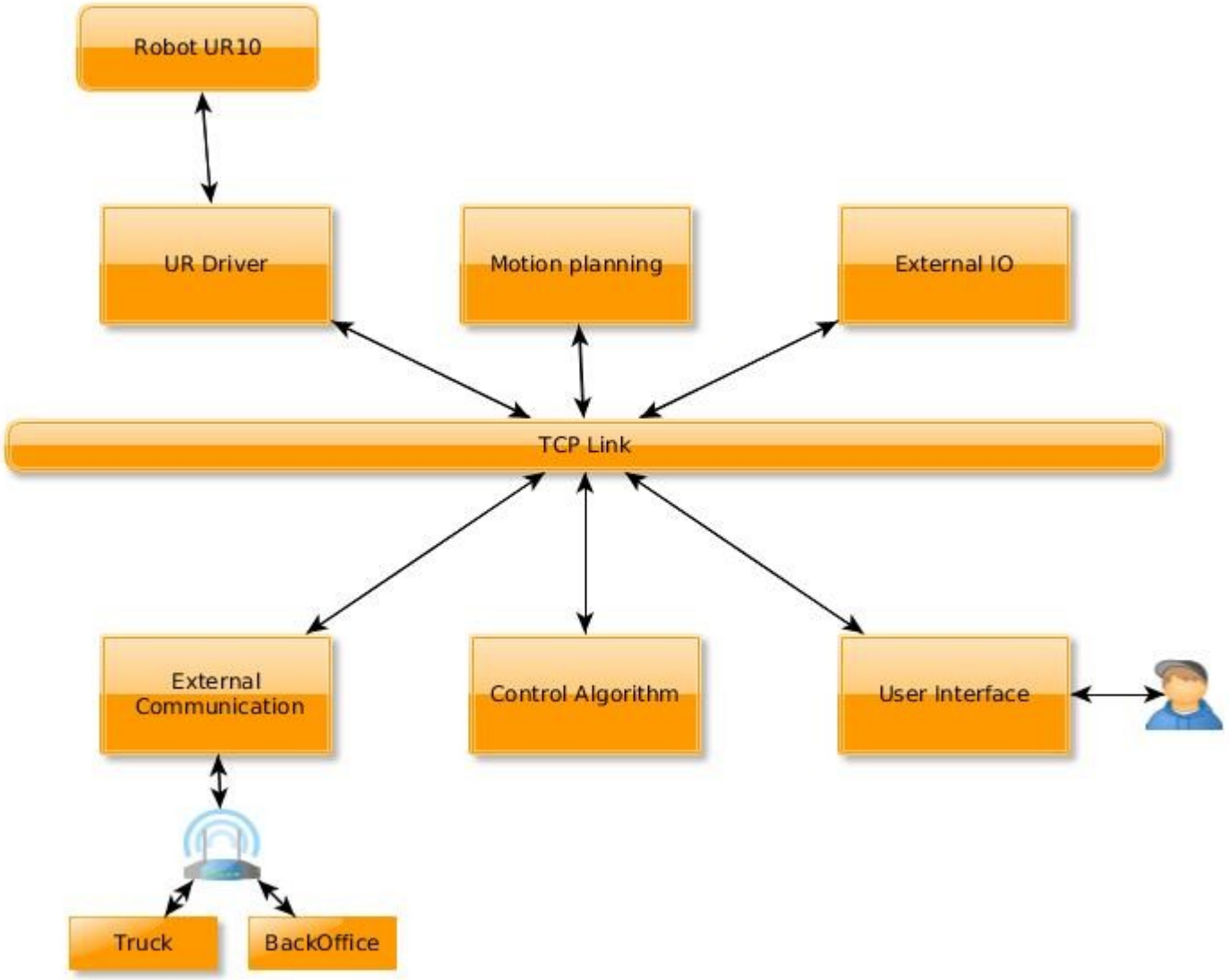
- Robot applications
  - Anton Olsson & Jonas Gustavsson
  - BSc Mechatronics, Chalmers (15ECTS)
- System configuration
  - Ashfaq Farooqui & Martin Viktorsson
  - MSc MPSYS, Chalmers (30ECTS)
- S2 Available theses, “Interfacing a Service Robot”

Universal Robot UR10

- Universal Robots UR10
  - Collaborative
  - Certified by TÜV
  - 10kg load
  - UR3, UR5



System Architecture



ROS (-I) Today

- ReApp
  - Financed by German authorities
  - Enabling technique in a cheap way for small and mid size companies
  - ROSi-libraries with standard operations
  - Plug & Produce
- FP7
  - Factory-in-a-day
  - SME-robotics
  - LIAA Lean Intelligent Assembly Automation

ROS (-I) Today

- BMW Autonomous driving

Development of ROS

- ROSi Consortium
  - South West Research institute (SwRI)
  - Fraunhofer IPA Stuttgart
- OEM
  - UR supports with wiki
  - ABB's ROS responsible is involved in Volvo projects and Robotdalen.
- Universities, Companies, Users etc.

ROS-I Montage

Video

Questions?

Content borrowed from:
Wiki.ros.org
Rosindustial.org