

# Cognitive Systems

## Practical Exercises – Session 3

**Prof. Dr.-Ing. habil. Alois Knoll**

Alexander Perzylo, Nikhil Somani, Ahmed Ghazi, Hendrik Walzel

Robotics and Embedded Systems (Informatik VI)

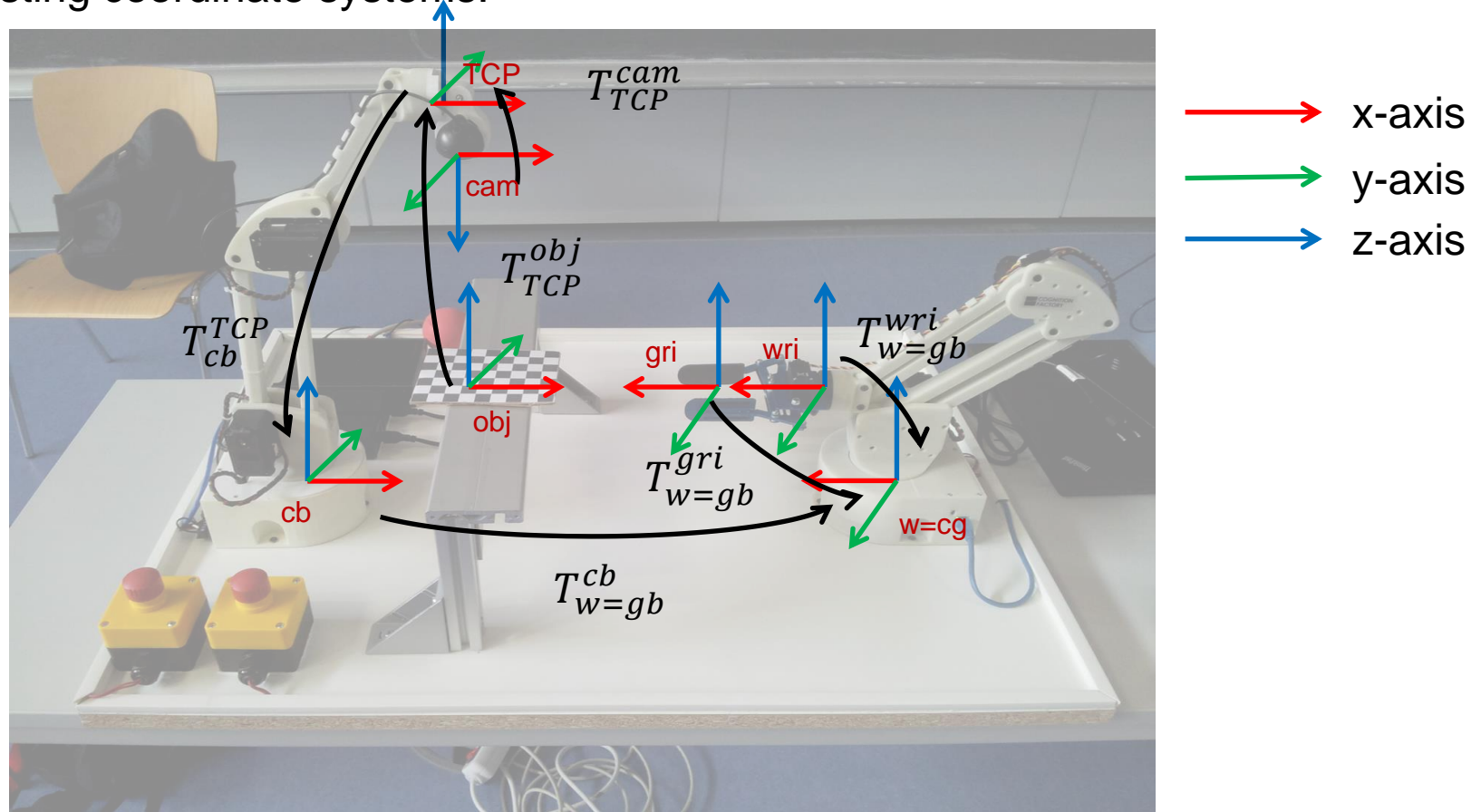
TUM

## Outline

- Overview of transformation tree for our robotic testbed
- Description of today's exercise
- Brief ROS summary

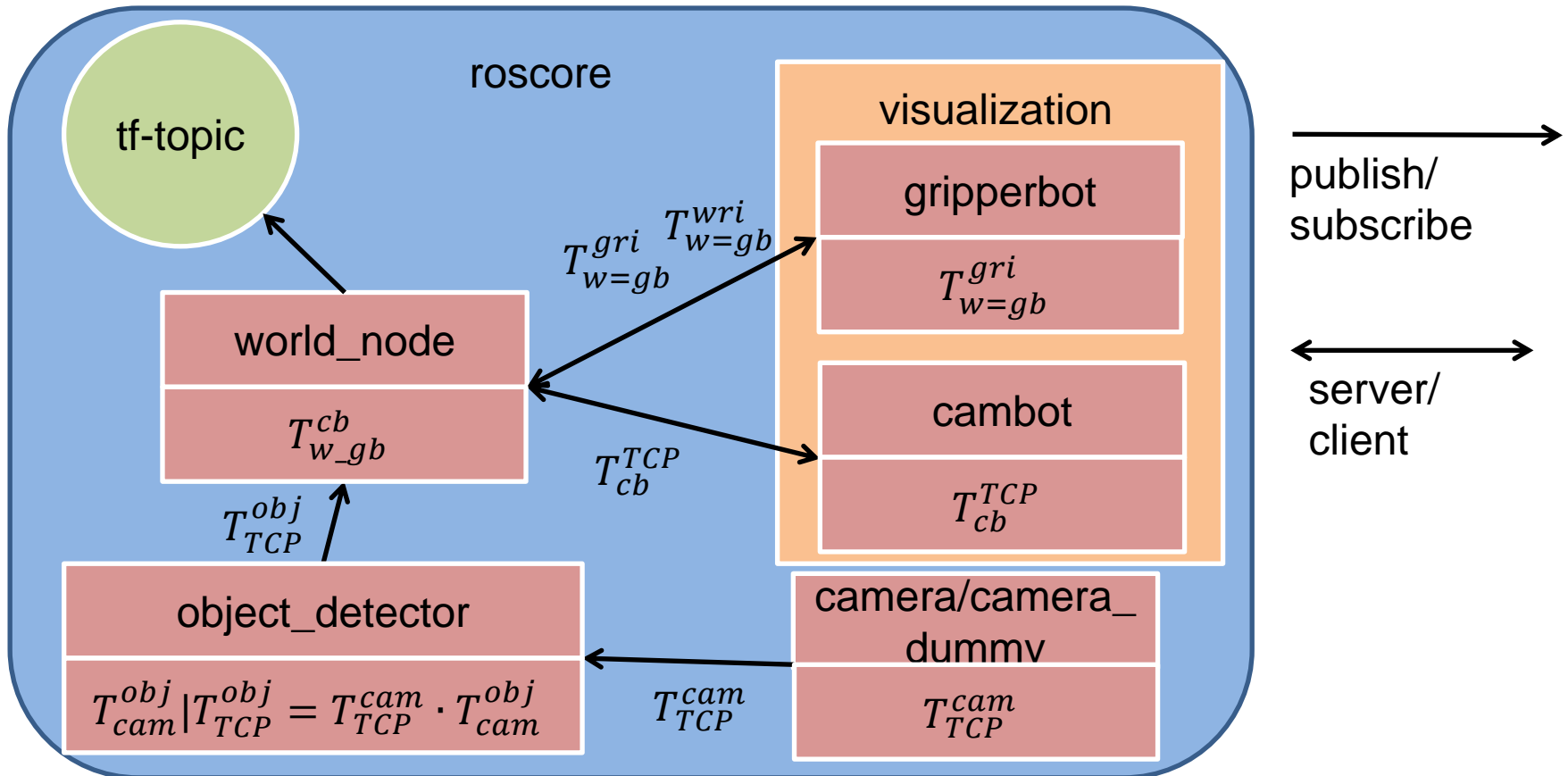
## Overview of transformation tree

- existing coordinate systems:



## Overview of transformation tree

- cogsys package dependencies:



## Today's exercise

- Finish the implementation of ROS package *world\_model* in src file *main.cpp*:
  - Add additional required TF publishers in function *main* using the given utility function *publish\_robot\_frame\_tf* . Some publishers for the gripperbot are already given. Add more for the cambot and one for the transformation between *gripperbot\_base* and *cambot\_base*. (Hints on how to do it are given in the source code)
- Finish the implementation of ROS package *demo\_cogsys\_cpp\_viz* in src file *main.cpp*:
  - Implement service call to */gripperbot\_control/move\_to\_cs* inside function *movegripperbot*
  - Implement function *grasp* (Hints on how to do it are given in the source code)
  - Be creative and add more functionality! ;-)

## Brief ROS summary

- packages: software organizing units, consist of executables, libraries, ...
- node: executable which uses ROS to communicate
- message: ROS data type used for publish/subscribe communication  
(<http://wiki.ros.org/ROS/Tutorials/CreatingMsgAndSrv>)  
(<http://wiki.ros.org/ROS/Tutorials/WritingPublisherSubscriber%28c%2B%2B%29>)
- topics: nodes can publish/subscribe to a topic in order to send/receive messages  
(<http://wiki.ros.org/ROS/Tutorials/UnderstandingTopics>)
- service: synchronized way of communication (server/client)  
(<http://wiki.ros.org/ROS/Tutorials/CreatingMsgAndSrv>)  
(<http://wiki.ros.org/ROS/Tutorials/WritingServiceClient%28c%2B%2B%29>)
- tf: feature that can build and maintain transformation trees  
(<http://wiki.ros.org/tf/Tutorials>)