API (Application Programming Interface)

- Software that allows two applications to communicate with each other
- Used to access, extract, and share data

NAOqi

- NAOqi is the main software application that runs on and controls the robot

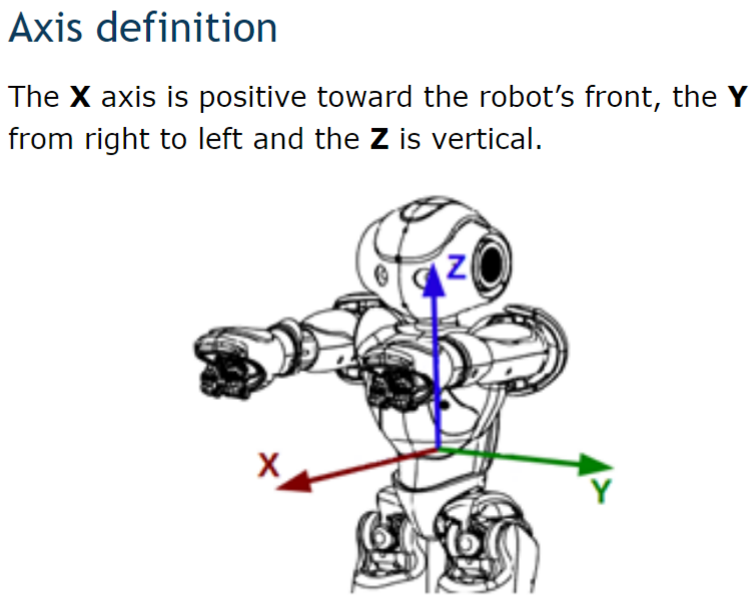
NAOqi Framework

- Programming framework used to program NAO
- Cross-platform supporting Windows, Mac or Linux
- Cross-language in Python and C++

Locomotion Control AL Motion

- The API used to control the movement of the robot
    - Controls joint stiffness
    - Controls joint position
    - Controls walk
    - Controls robot effector in the Cartesian space
- ALMotions runs at 50Hz or a cycle of 20ms. Every time a public method is called to request a motion, a “motion task” is created to handle the job.
- ALMotion is a core module that manages and updates the model during every cycle, thus consuming a constant CPU

ALMotion Axis Definition



Axis definition

The **X** axis is positive toward the robot’s front, the **Y** from right to left and the **Z** is vertical.

AL Motion Unit System

- SI International System of Units

    - Meters
    - Seconds
    - Radians

High-Level Controls

Operation	Function
ALMotionProxy::moveTo	a target pose on the ground plane, that the robot will walk to.
ALMotionProxy::move	the robot's instantaneous velocity (direction and intensity) in SI units (typically used to control the walk from a loop, with external input such as a visual tracker).
ALMotionProxy::moveToward	the robot's instantaneous normalized velocity (direction and intensity) interactively (typically used to control the robot from a joystick, when the input gets normalized anyway).

Robotics API

- Programming Language: Java
- Open Source under the MPL license

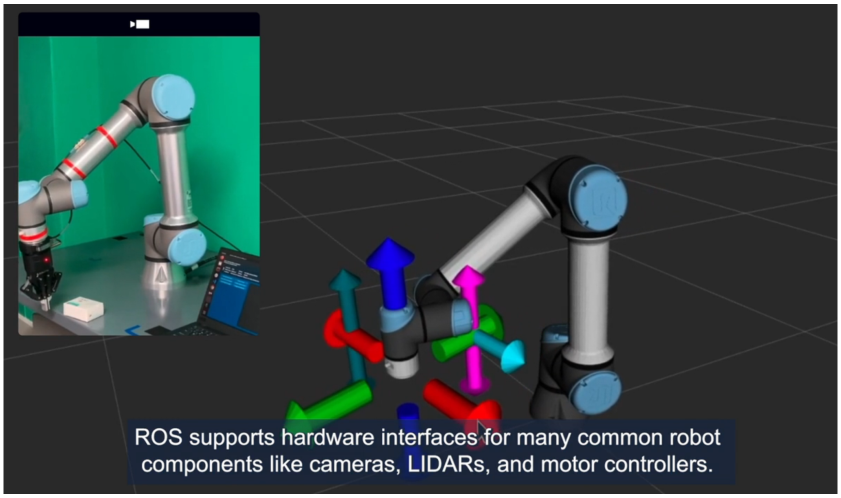
Note: Basic information can be found on their website though detailed software documentation sites are all unreachable.

Robot Framework Automation Framework

- Test automation
- RPA (Robotic Process Automation)
- Acceptance test-driven development (ATDD)
- Behavior-driven development (BDD)
- Python oriented
- User Guide

ROS (Robot Operating System) for simulations

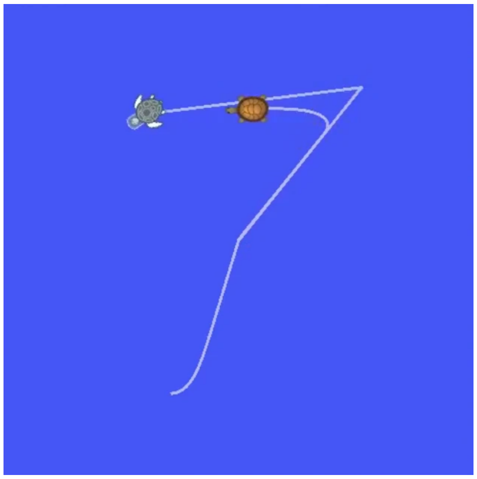
- Warning: There's a lot that goes into ROS Iron's installation
- Software libraries and tools that help build robot applications
- Defines all the tools, components, and interfaces to build a robot

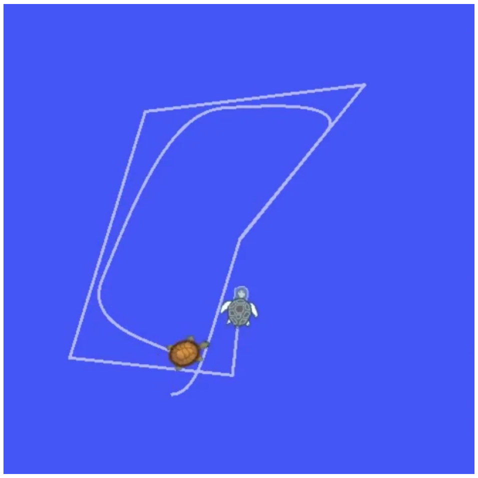
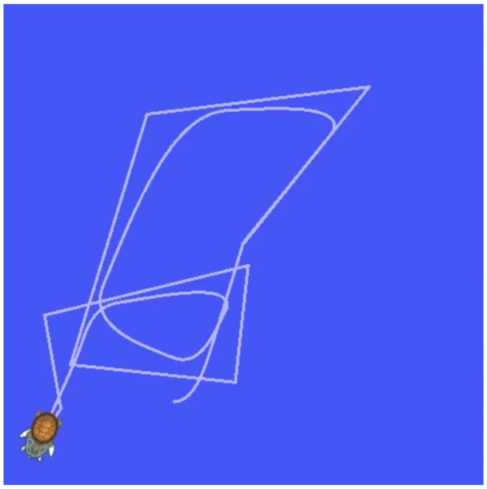
- 
ROS supports hardware interfaces for many common robot components like cameras, LIDARs, and motor controllers.

ROS2 tf2 transform library

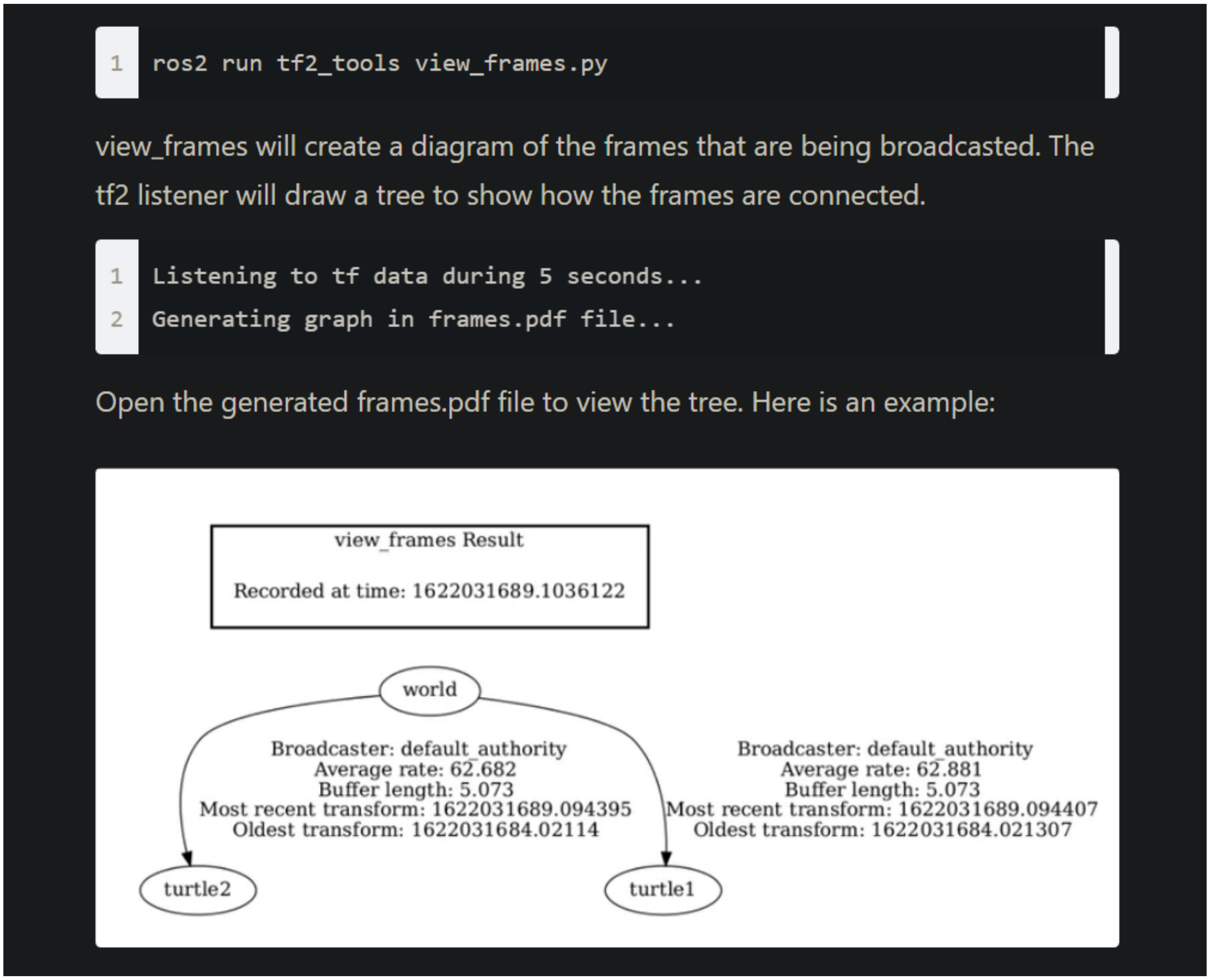
- tf2 is a transform library that allows the user to keep track of multiple coordinate frames over time. Operating in a distributed system, all the information about the coordinate frames and their relationships is available to the ROS 2 components
- tf2 maintains the relationship between coordinate frames in a tree structure buffered in time
- Allows the user to transform points and vectors between any coordinate frame whenever needed
- Any component of the distributed system can build a transformed information database
- Gather and store all transform information with a central node
- Listening and broadcasting transforms
  - To transform between coordinate frames the node would have to listen for the transforms
  - The transform is then buffered and broadcasted to the system
  - Inquire about the transformed information between frames

ROS2 Turtle Example

- 

- 
- 
- The player moves as the green turtle as the orange one calculates its location relative to the player thus continuously moving in its direction

tf2 tools



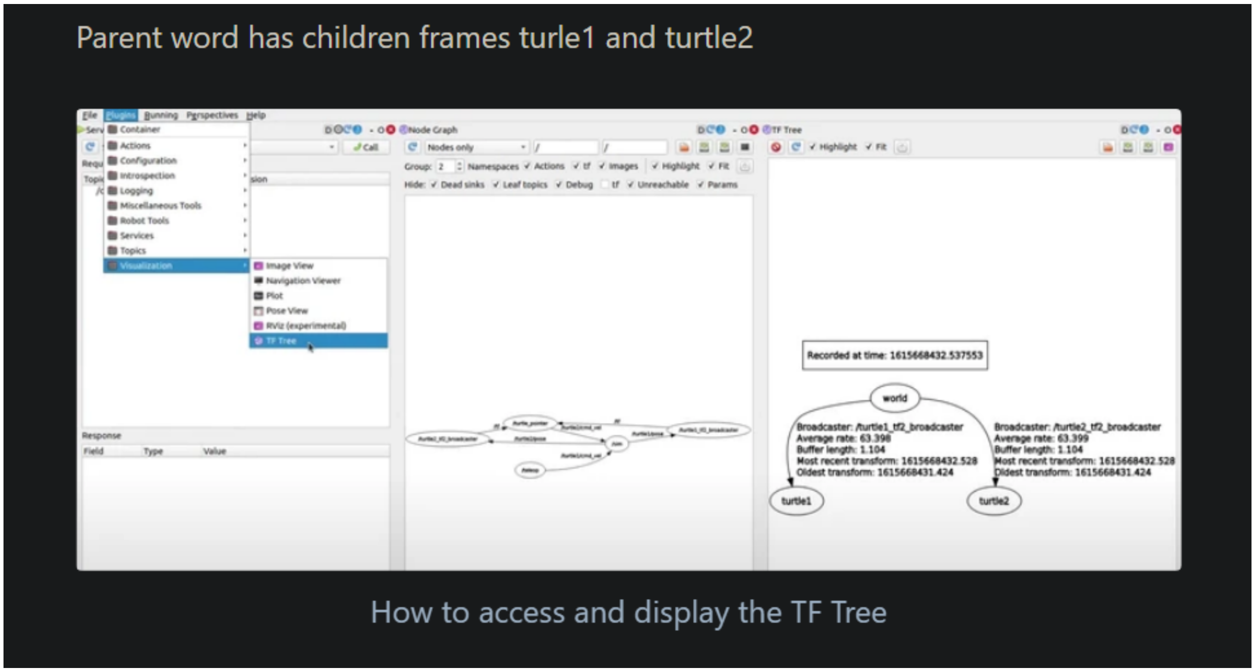
```
ros2 run tf2_tools view_frames.py
```

view_frames will create a diagram of the frames that are being broadcasted. The tf2 listener will draw a tree to show how the frames are connected.

```
Listening to tf data during 5 seconds...
Generating graph in frames.pdf file...
```

Open the generated frames.pdf file to view the tree. Here is an example:

view_frames Result
Recorded at time: 1622031689.1036122
world
Broadcaster: default_authority
Average rate: 62.682
Buffer length: 5.073
Most recent transform: 1622031689.094395
Oldest transform: 1622031684.02114
Broadcaster: default_authority
Average rate: 62.881
Buffer length: 5.073
Most recent transform: 1622031689.094407
Oldest transform: 1622031684.021307
turtle2
turtle1



Parent word has children frames turle1 and turtle2

```
Recorded at time: 1615668432.537553
```

```
Broadcaster: /turtle1_tf2_broadcaster
Average rate: 63.398
Buffer length: 1.104
Most recent transform: 1615668432.528
Oldest transform: 1615668431.424
```

```
Broadcaster: /turtle2_tf2_broadcaster
Average rate: 63.399
Buffer length: 1.104
Most recent transform: 1615668432.528
Oldest transform: 1615668431.424
```

How to access and display the TF Tree

Note:

Raspberry Pi libraries for physical robots