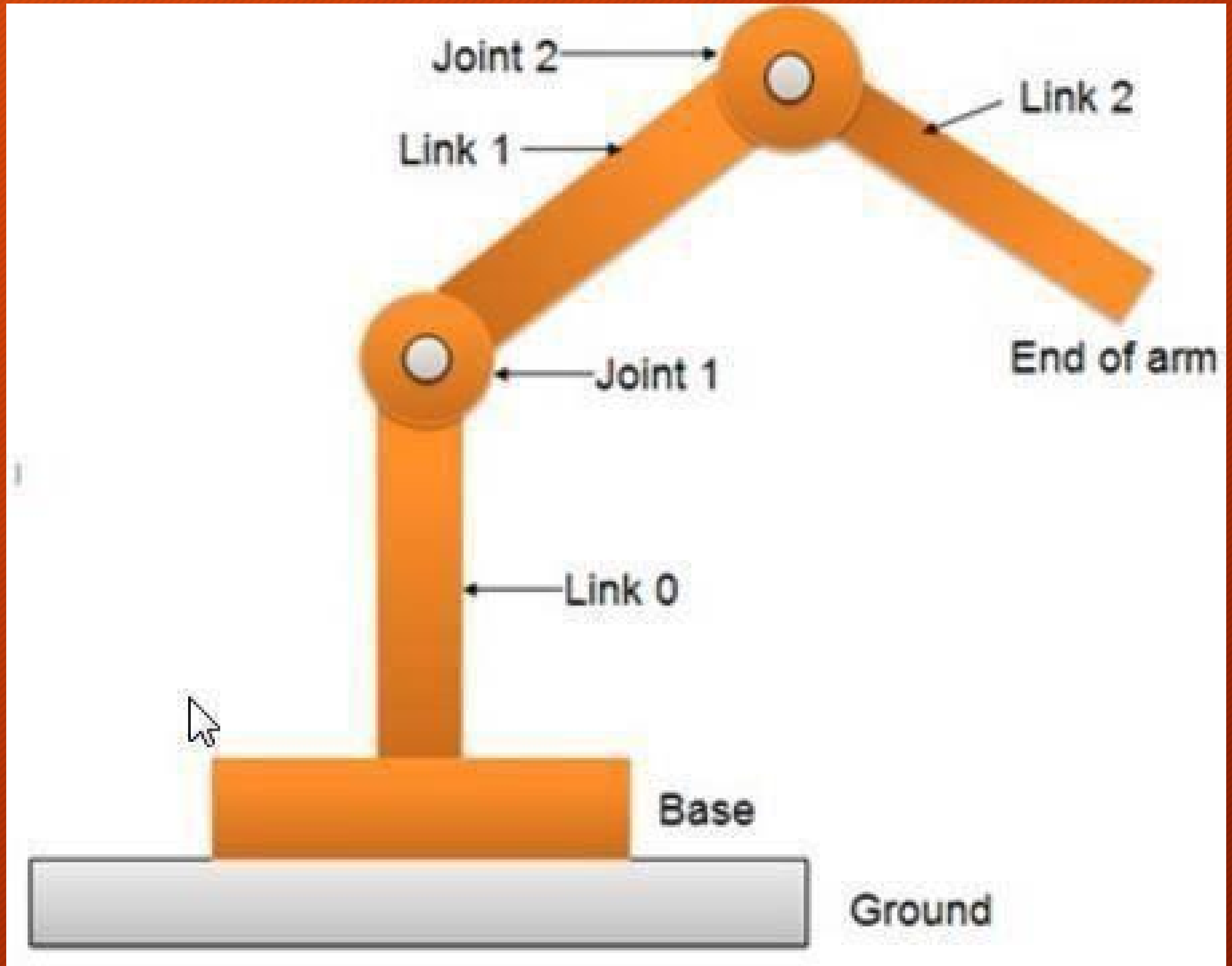
Introduction to URDF and RViz

ROS 2 Humble Edition
By: Susan Kimani

What is URDF?

- XML format for describing a robot
- Defines **links, joints, visuals, collisions**
- Used for **simulation, visualization & planning**



RViz

RViz (short for ROS Visualization) is a 3D visualization tool in ROS/ROS 2.

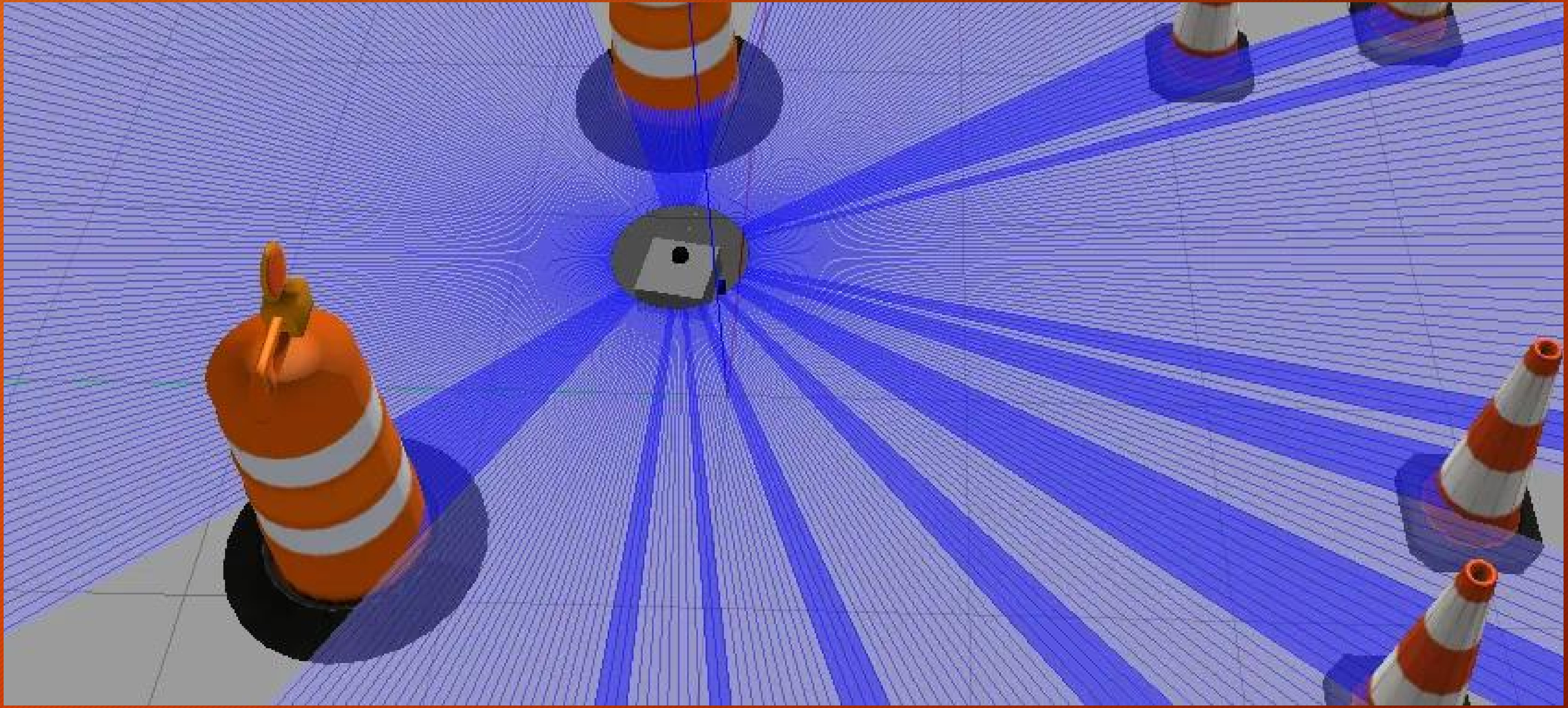
Gazebo

- Robotics simulation with SDF worlds and physics

- Open-source 3D robotics simulator.

- Provides physics, lighting, sensors, and environments.

Example Gazebo World



Why URDF?

- Lets ROS understand robot’s structure
- Enables RViz & Gazebo visualization
- Provides transforms (tf2) for navigation & planning

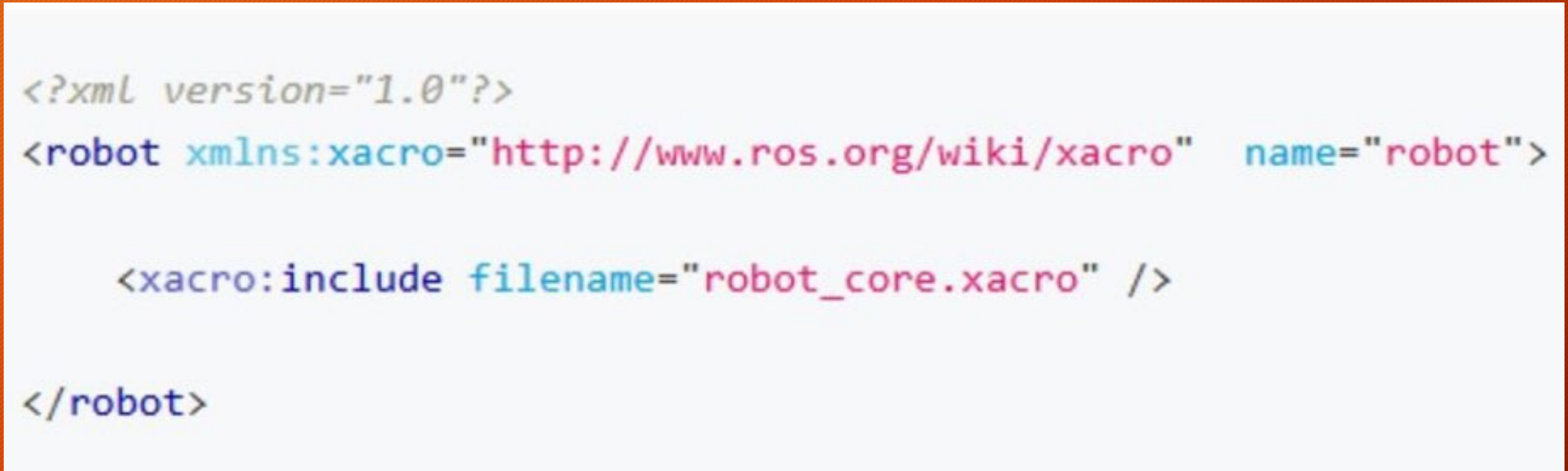
History & Context

- First in ROS1 → continued in ROS2
- Common XML
- Alternatives:
    - **Xacro** → URDF with macros (less repetition)
    - **SDF** → More detailed, mainly used in Gazebo

Xacro

- XML Macros for URDF
- Avoid repetition in robot description
- Useful for complex robots

Xacro



```
<?xml version="1.0"?>
<robot xmlns:xacro="http://www.ros.org/wiki/xacro"  name="robot">

    <xacro:include filename="robot_core.xacro" />

</robot>
```

SDF

- Simulation Description Format
- More detailed than URDF
- Preferred in Gazebo simulations

Meshes

- Used to represent detailed robot geometry
- File formats: .dae, .stl
- Makes visualization realistic

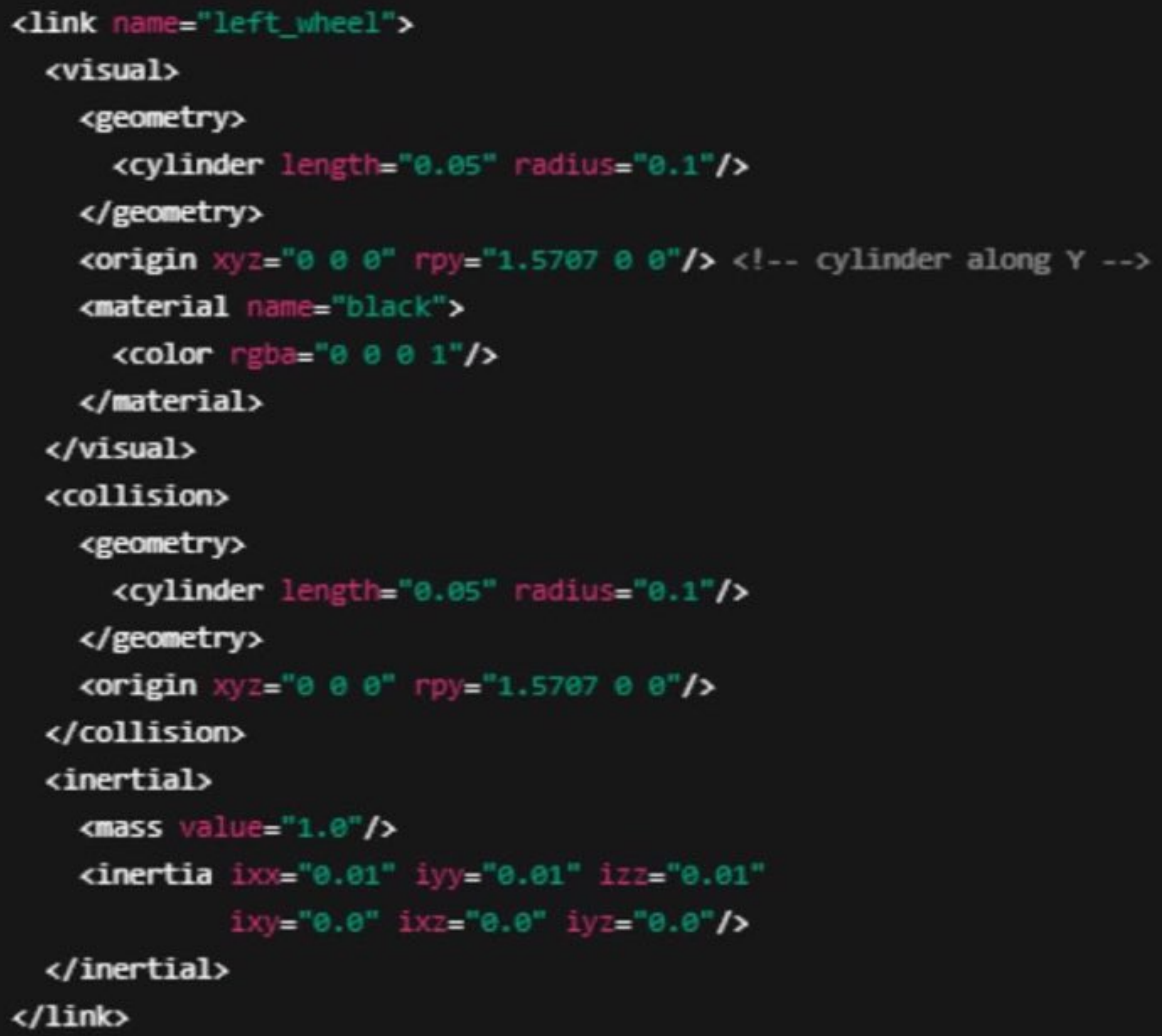
Key Elements of URDF

- **Root element** → Wraps everything <robot>
- **Links** → Robot parts (e.g., chassis, wheel)
- **Joints** → Connections (fixed, revolute, continuous)

Key Elements of URDF

- **Visual** → How it looks in Rviz
- **Collision** → Shape for physics simulation
- **Inertial** → Mass & center of gravity

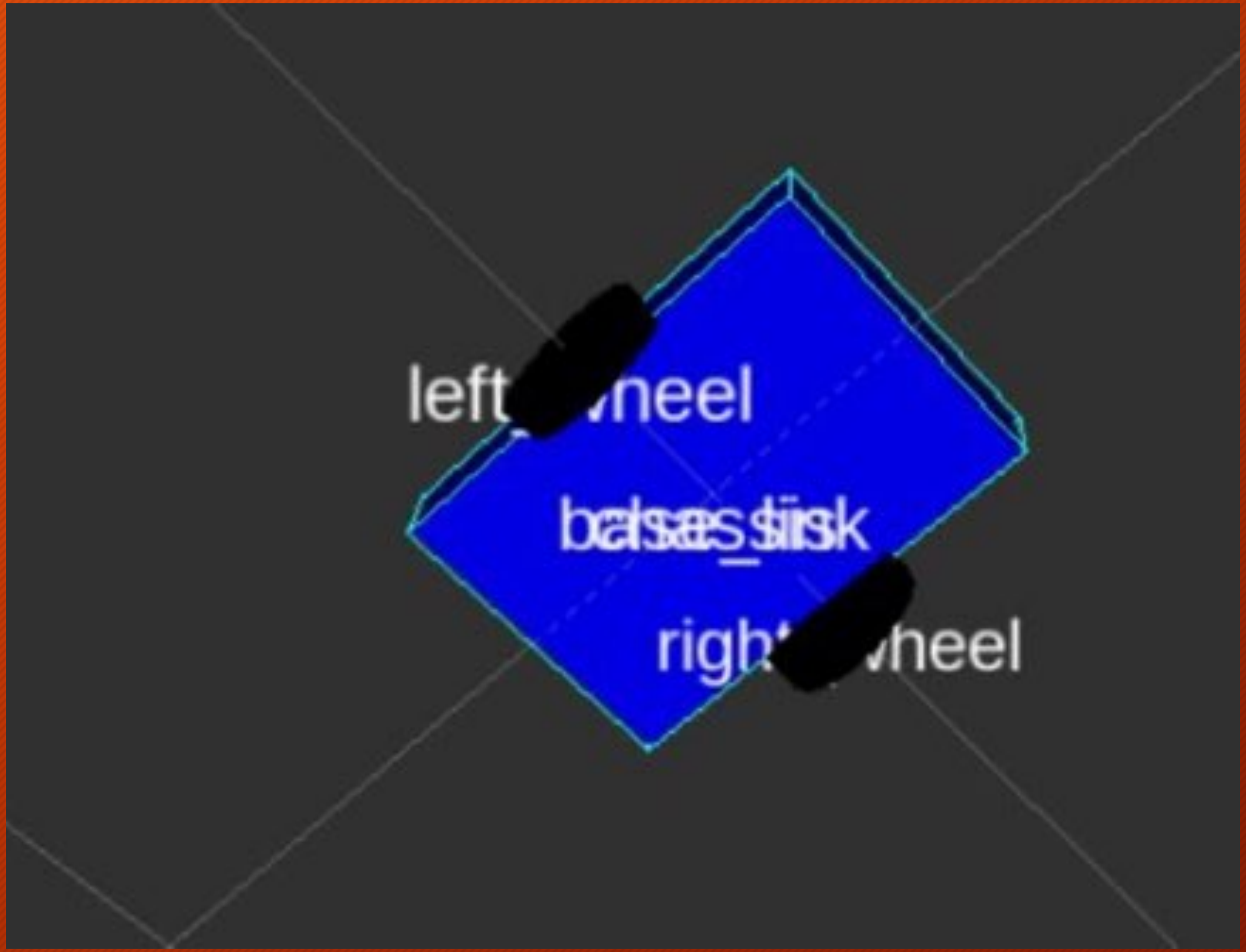
Here is an example of the key elements in an xml file



```
<link name="left_wheel">
  <visual>
    <geometry>
      <cylinder length="0.05" radius="0.1"/>
    </geometry>
    <origin xyz="0 0 0" rpy="1.5707 0 0"/> <!-- cylinder along Y -->
    <material name="black">
      <color rgba="0 0 0 1"/>
    </material>
  </visual>
  <collision>
    <geometry>
      <cylinder length="0.05" radius="0.1"/>
    </geometry>
    <origin xyz="0 0 0" rpy="1.5707 0 0"/>
  </collision>
  <inertial>
    <mass value="1.0"/>
    <inertia ixx="0.01" iyy="0.01" izz="0.01"
             ixy="0.0" ixz="0.0" iyz="0.0"/>
  </inertial>
</link>
```

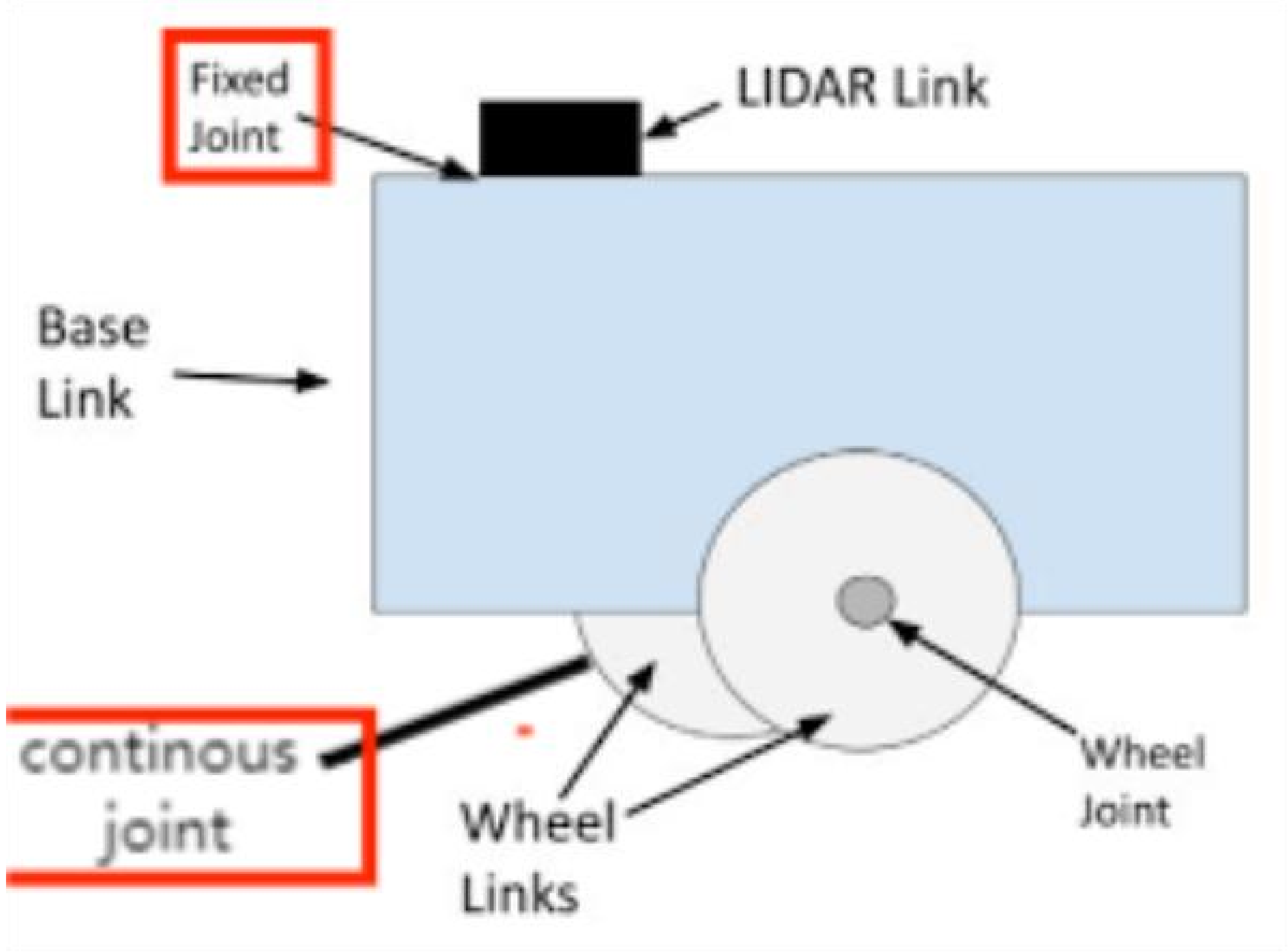
URDF: Links

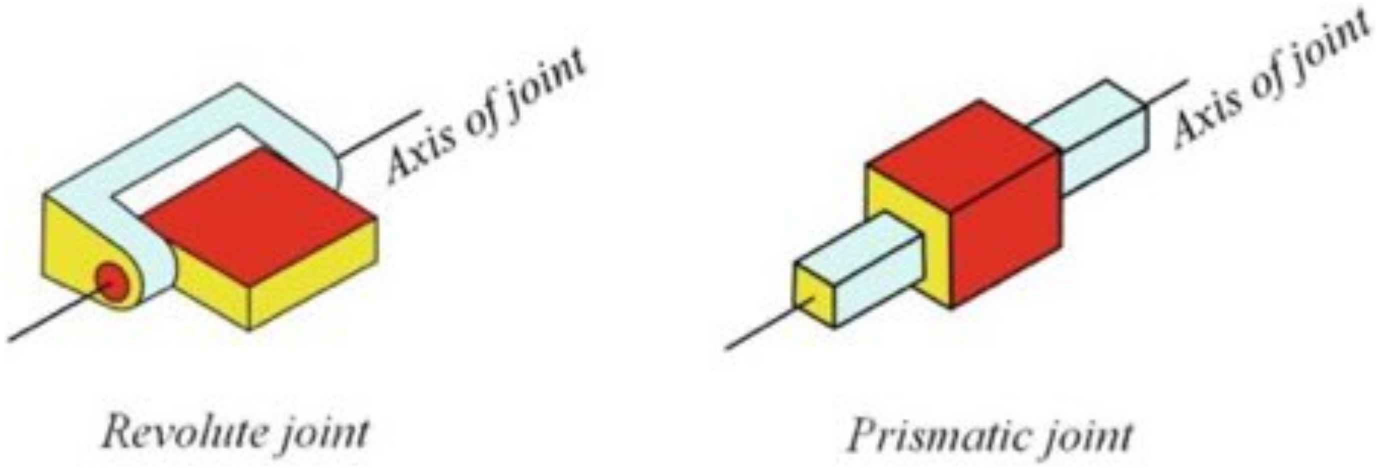
- Represents rigid body parts
- Example: chassis, wheel, arm segment



URDF: Joints

- Defines how links connect
- Types: fixed, revolute, continuous, prismatic
- Example: wheel joint with chassis





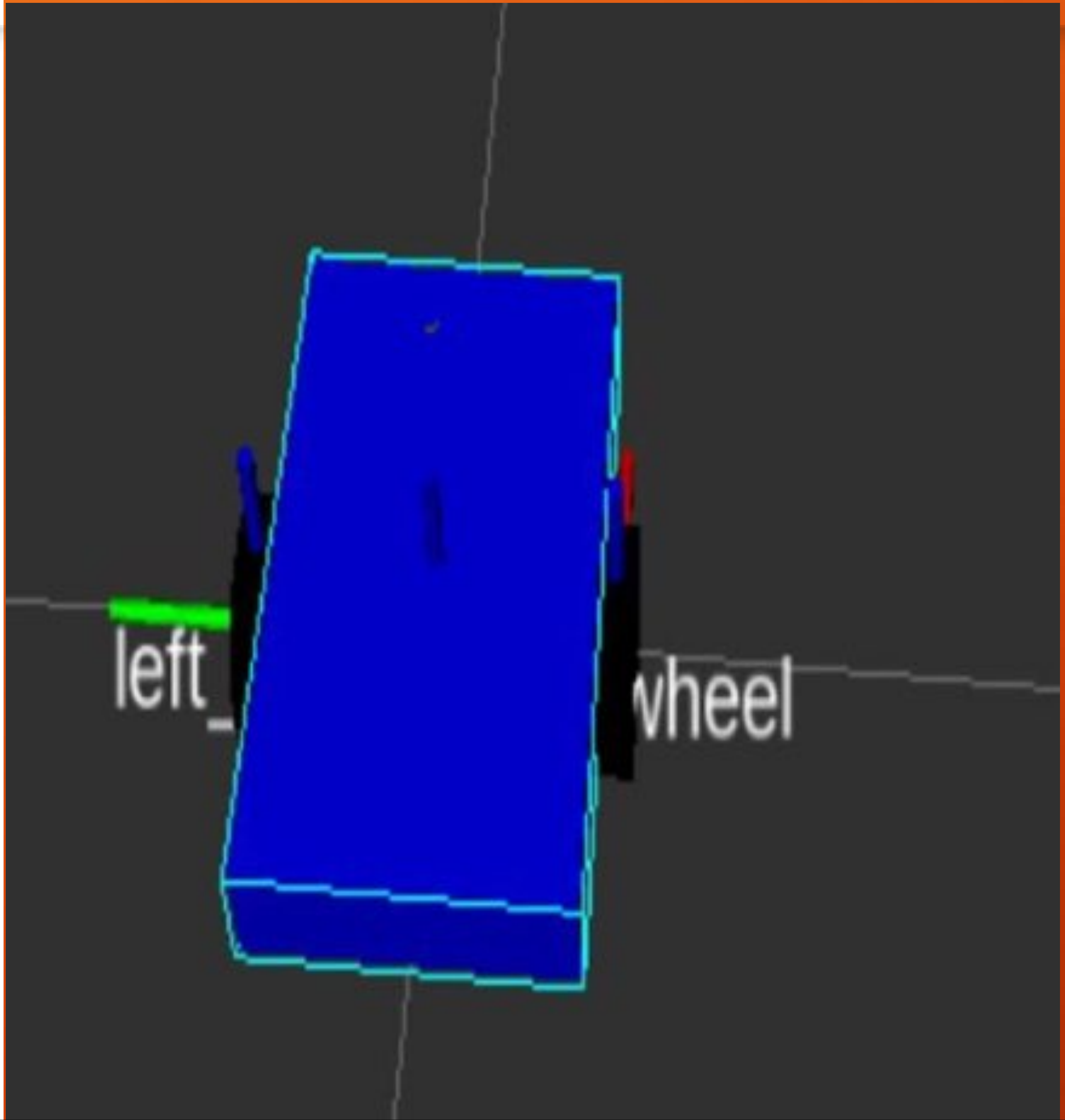
Revolute joint

Prismatic joint

URDF: Visual vs Collision

- Visual = appearance in Rviz
- Collision = simplified physics shape; what robot bumps into
- Often use box/cylinder for collision

Visuals



- **Visuals in robotics** are pictures, 3D models, or animations that help us **see what the robot looks like and how it moves**.

- They make it easier to **understand and debug robots** without needing the real hardware

- Here is an exam of a visual loaded from URDF file

Collision

- **Collision** means when two objects in a robot's world **touch or bump into each other**.

- In simulation, we give robots **collision shapes** (like boxes, cylinders, spheres) so the computer knows their boundaries.

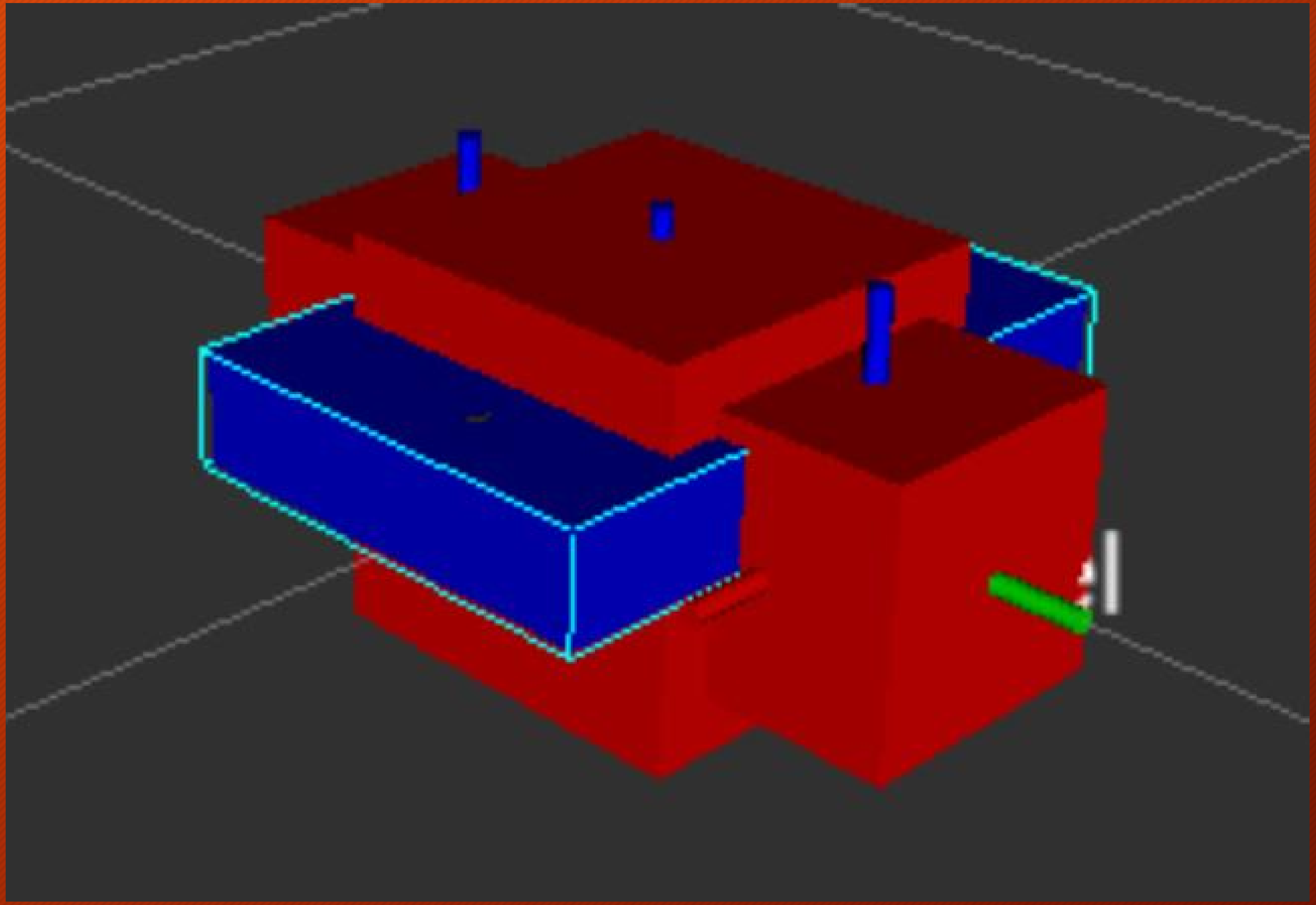
- This helps the robot **avoid accidents** and lets the simulator check if movements are safe.

URDF: Inertial

- Defines mass & center of gravity

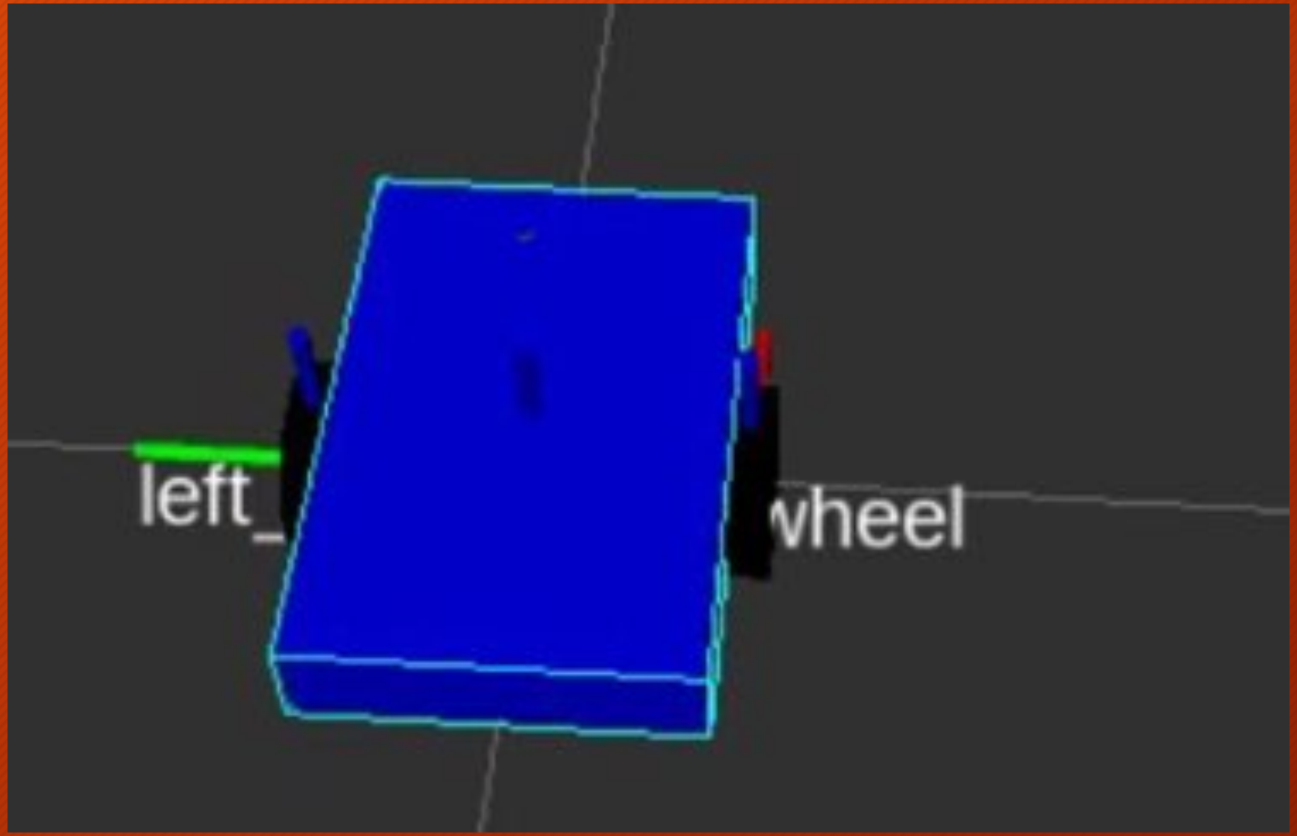
- Important for physics simulation

- Not always needed for basic visualization



Example URDF

- Simple mobile robot base
- 1 chassis + 2 wheels
- Fixed frame: chassis



Live Demo

Takeaway

- URDF is the **blueprint** of a robot in ROS2
- Needed for **visualization, planning & simulation**
- Start simple (chassis + wheels), build complexity later

References

1. https://roboticsdojo.substack.com/p/understanding-urdf-building-the-blueprint

2. https://wiki.ros.org/urdf

3. https://wiki.ros.org/urdf/Tutorials

Introduction to RViz in ROS 2 Humble

What is RViz?

- 3D visualization tool for robots
- Rviz is like the eye of your
- it lets you see data, robot state and maps

Why Use RViz?

- Visualize sensor data (LiDAR, Camera, IMU)
- See robot model & transformations
- Debugging robot motion planning
- Learn ROS 2 concepts interactively

Prerequisites

- ROS 2 Humble Desktop installed
- Workspace created and sourced (source install/setup.bash)
- Verify RViz installed:

```
ros2 pkg list | grep rviz2
```

Installing & Launching RViz2

- Install (if missing):

```
sudo apt install ros-humble-rviz2
```

- Launch with:

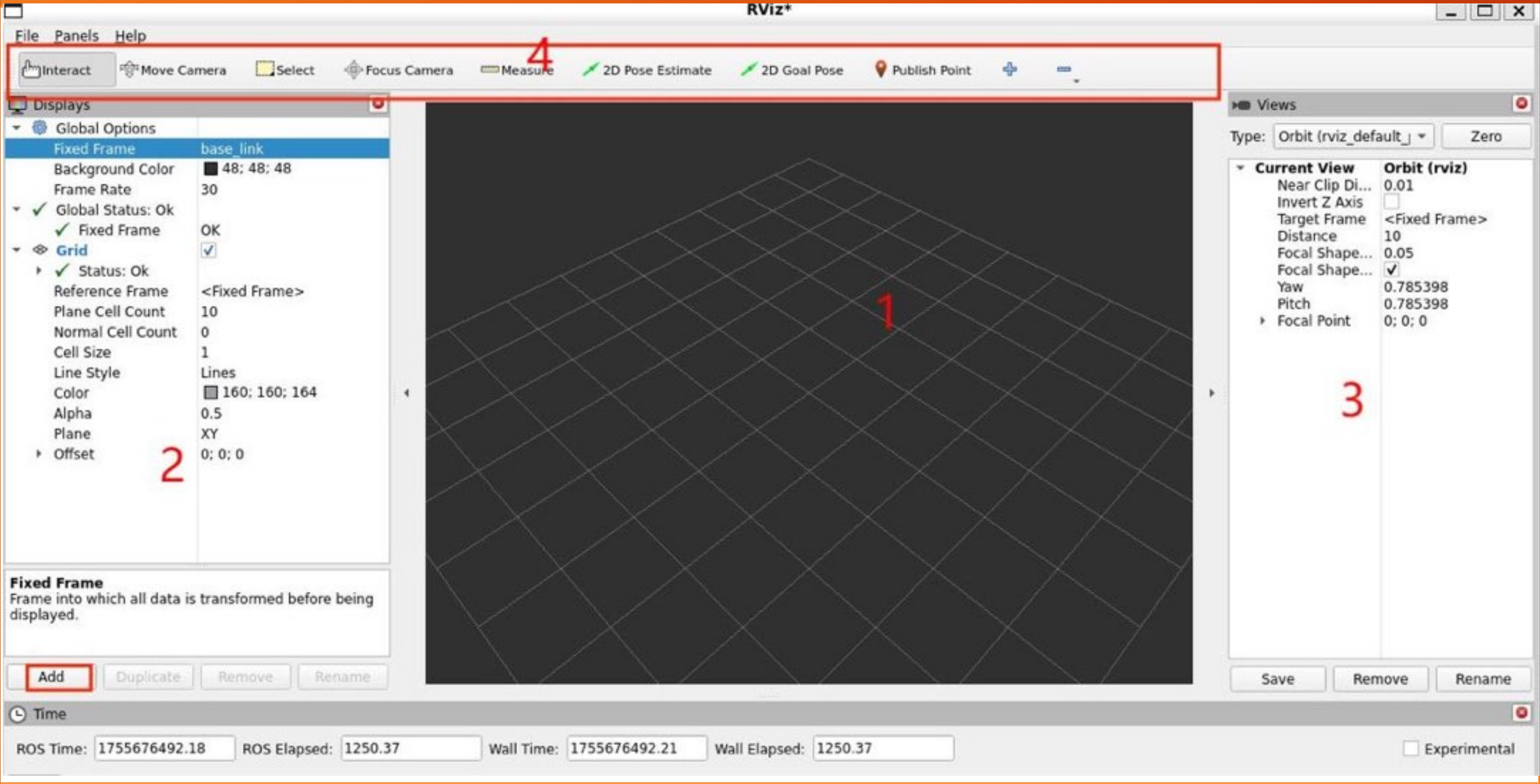
```
rviz2
```

- Common error: *Command not found* → fix by sourcing ROS2

RViz Interface Overview

1. **3D View (center)** – visualization area
2. **Displays Panel (left)** – manage robot model & sensors
3. **Properties Panel (right)** – adjust settings
4. **Toolbar (top)** – tools for navigation & interaction

Rvizs Window



Global Options	
Fixed Frame	base_link
Background Color	■ 48; 48; 48
Frame Rate	30
✓ Global Status: Ok	
✓ Fixed Frame	OK
Grid	☑
✓ Status: Ok	
Reference Frame	<Fixed Frame>
Plane Cell Count	10
Normal Cell Count	0
Cell Size	1
Line Style	Lines
Color	■ 160; 160; 164
Alpha	0.5
Plane	XY
Offset	0; 0; 0

Fixed Frame
Frame into which all data is transformed before being displayed.

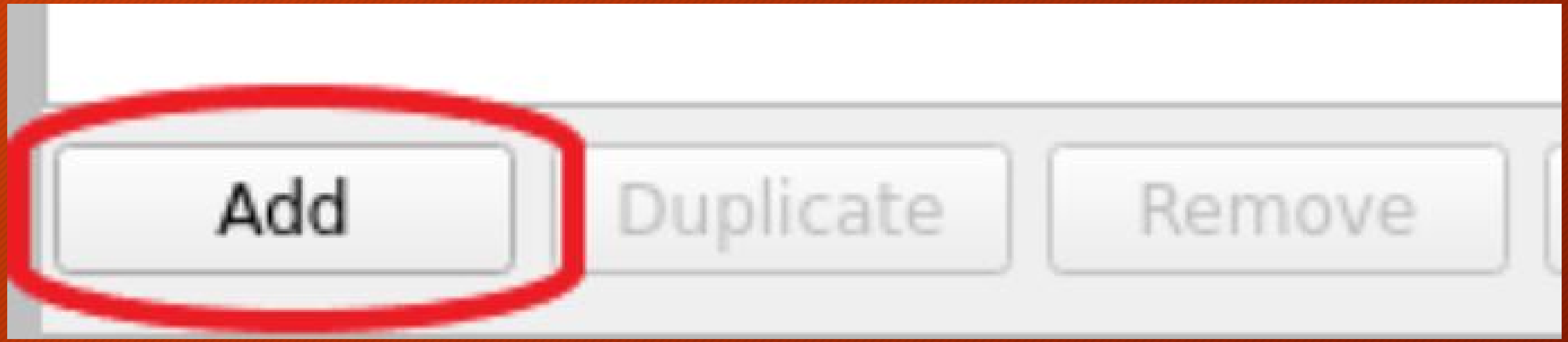
Type: Orbit (rviz_default_

Current View	Orbit (rviz)
Near Clip Di...	0.01
Invert Z Axis	☐
Target Frame	<Fixed Frame>
Distance	10
Focal Shape...	0.05
Focal Shape...	☑
Yaw	0.785398
Pitch	0.785398
Focal Point	0; 0; 0

ROS Time: 1755676492.18 ROS Elapsed: 1250.37 Wall Time: 1755676492.21 Wall Elapsed: 1250.37 ☐ Experimental

Adding a Display

- Click **Add** → Choose type
- Example: RobotModel (to load URDF robot)
- Check **Fixed Frame** matches (base_link or chassis)
- Robot now appears in 3D view



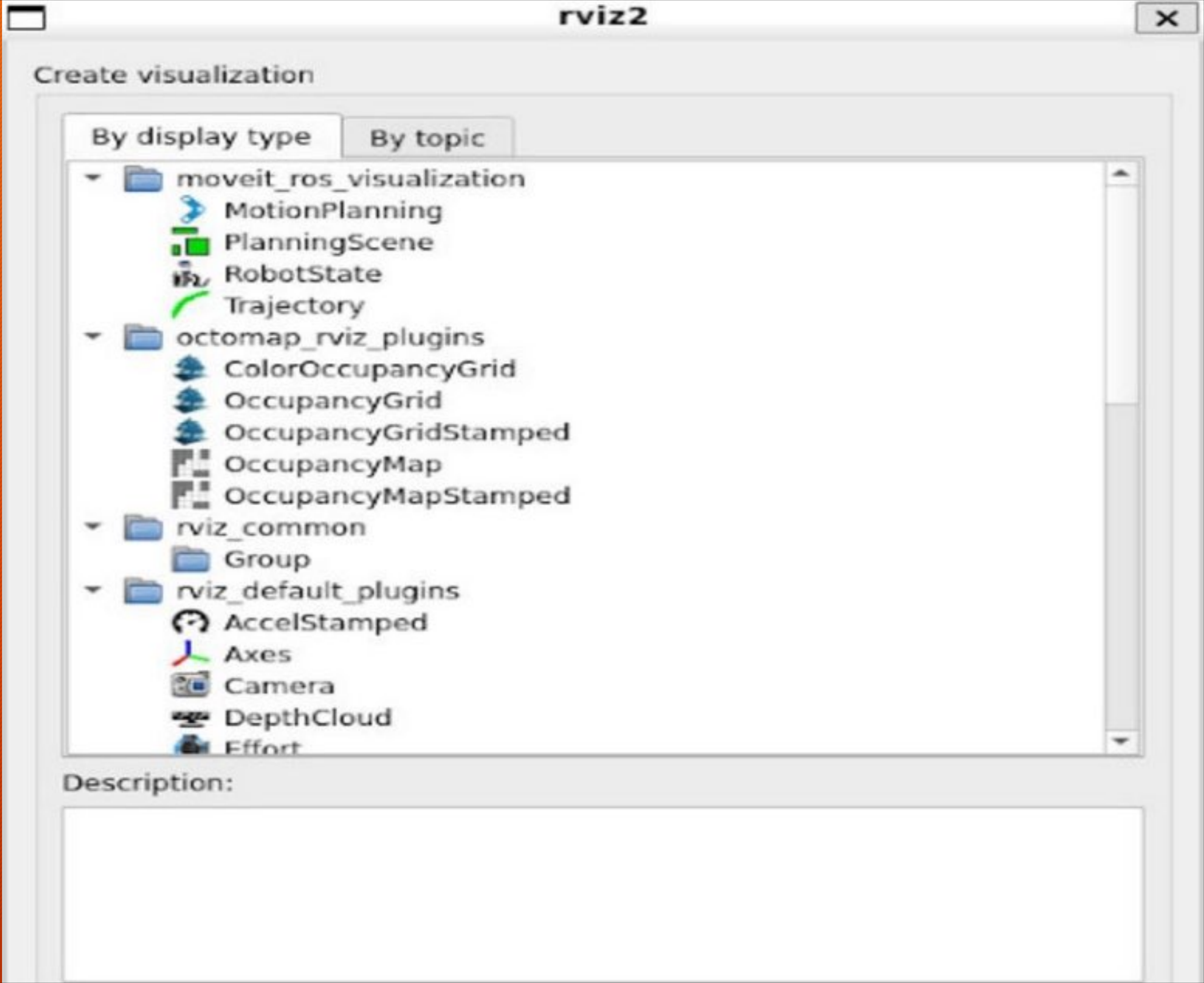
The window shows a list of plugins (tools) you can add to visualize different information.

For example:

- **octomap_rviz_plugins →** shows motion planning things like the robot's planned path

moveit_ros_visualization → shows 3D maps of the environment (what the robot "sees").

- **rviz_default_plugins →** basic tools like axes, camera view, depth clouds, etc.



rviz2

Create visualization

By display type | By topic

- moveit_ros_visualization
  - MotionPlanning
  - PlanningScene
  - RobotState
  - Trajectory
- octomap_rviz_plugins
  - ColorOccupancyGrid
  - OccupancyGrid
  - OccupancyGridStamped
  - OccupancyMap
  - OccupancyMapStamped
- rviz_common
  - Group
- rviz_default_plugins
  - AccelStamped
  - Axes
  - Camera
  - DepthCloud
  - Effort

Description:

Useful Tools in RViz

- **Camera Controls** → Orbit, FPS, Top-down
- **Measurement Tool** → check distances
- **2D Pose Estimate & 2D Goal** → simulate navigation
- **Select Tool** → interact with objects

Summary

- RViz = robot visualization tool in ROS2
- Helps debug, test, and learn
- Add displays for sensors, robot model, TF
- Key step before hardware testing

References

1. https://roboticsdojo.substack.com/p/getting-started-with-rviz-in-ros

2. https://www.youtube.com/watch?v=6pep5xB4pEU
3. https://www.youtube.com/watch?v=WA3ynlo30v
4. https://www.youtube.com/watch?v=WA3ynlo30vw

Thank you