

ROS2 Navigation stack and
study for use in our robot

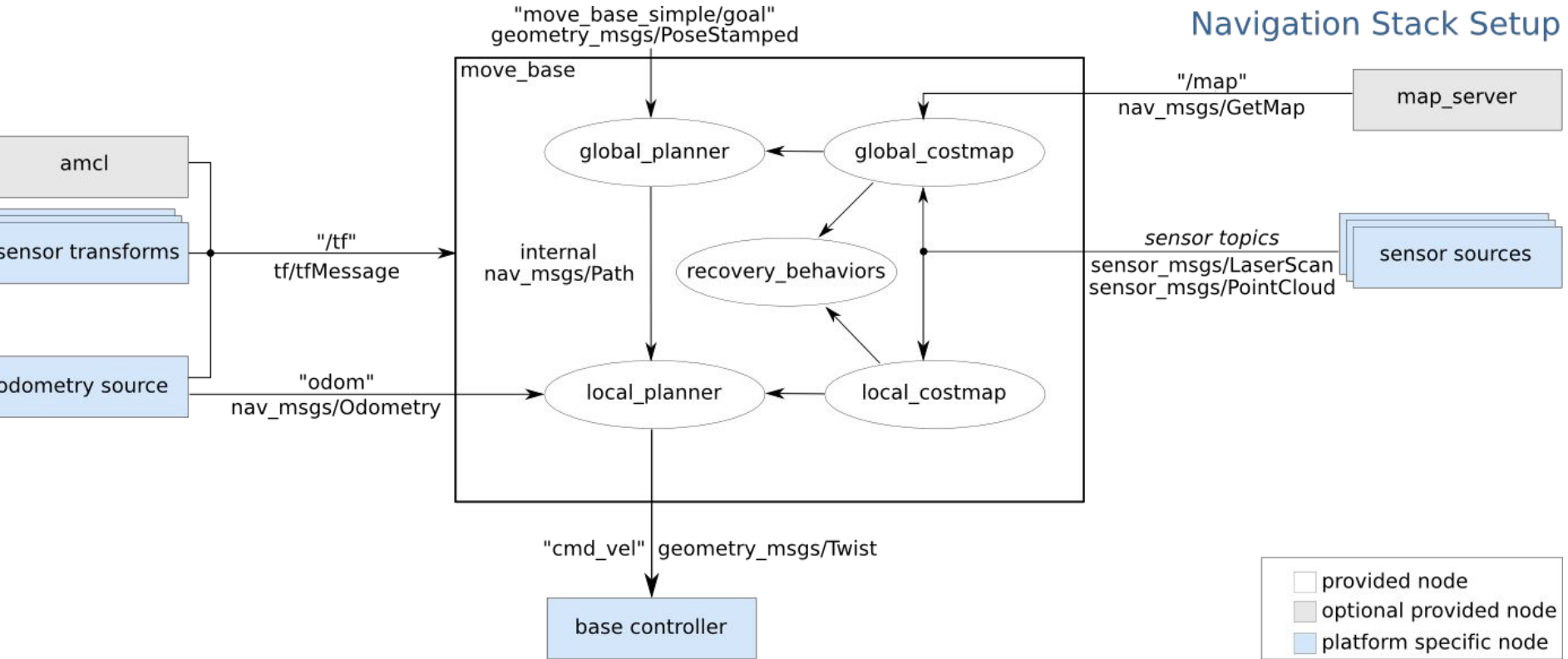
Some key points from previous discussion

- It is important to study navigation2 in detail, to see what is going in and what else is required, ex- behavior trees
- Having messages being exchanged and controllers in process would help us in finalizing the roadmap
- Even though we have Roswan drivers with us, it is not necessary to use previous r2c2 architecture
- See what else functionalities are provided with ros2 for use in future
- Good to look into everything and then decide !

ROS to ROS2 Navigation stack comparison

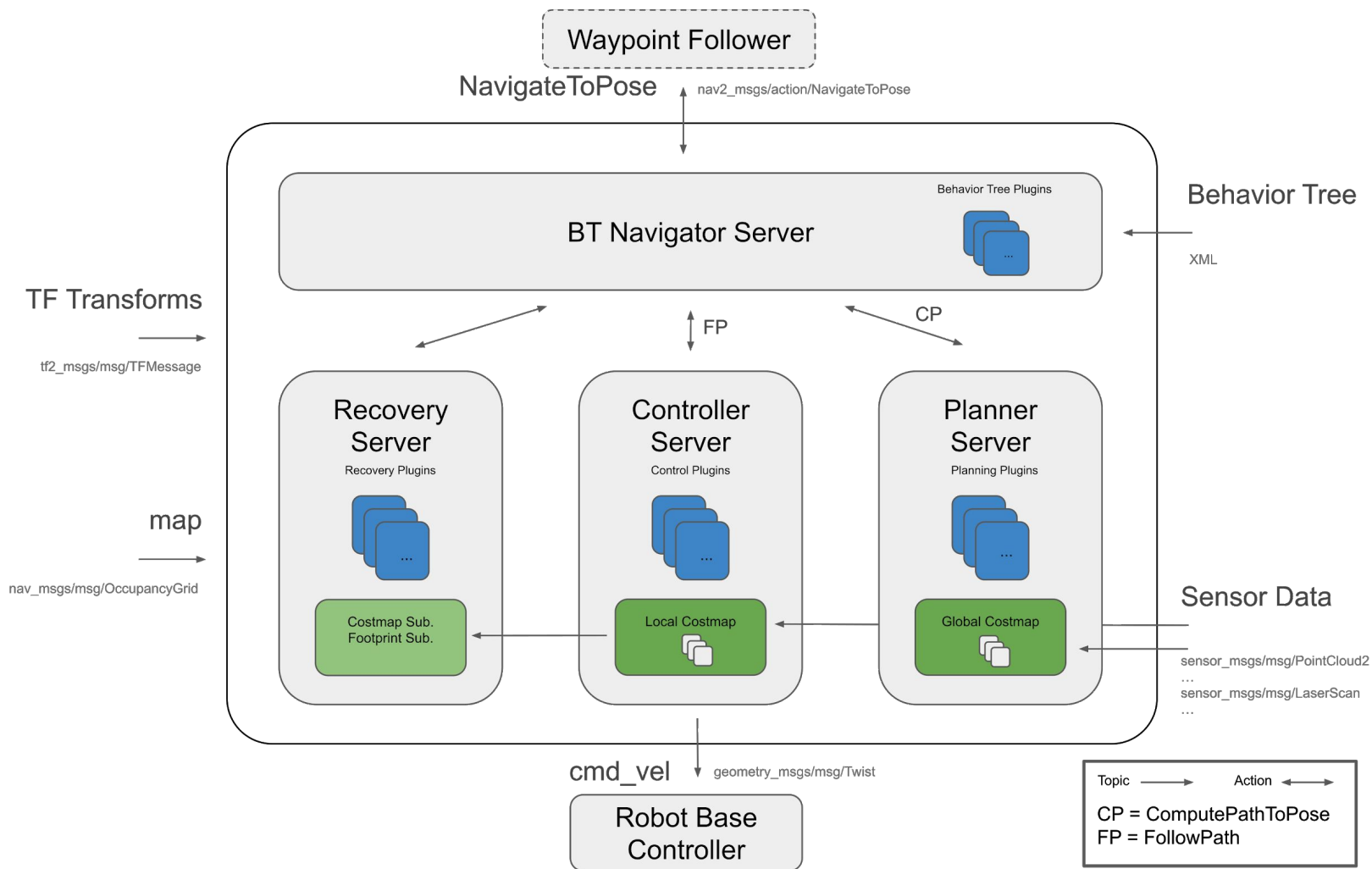
- **Move_base** splitted into multiple components
- Ported packages:
 - amcl: Ported to nav2_amcl
 - map_server: Ported to nav2_map_server
 - nav2_planner: Replaces global_planner
 - nav2_controller: Replaces local_planner
 - Navfn: Ported to nav2_navfn_planner
 - DWB: Replaces DWA and ported to ROS2 under nav2_dwb_controller metapackage
 - nav_core: Ported as nav2_core with updates to interfaces
 - costmap_2d: Ported as nav2_costmap_2d

Navigation Stack Setup



contd...

- New packages:
 - nav2_bt_navigator: replaces move_base state machine
 - nav2_lifecycle_manager: Handles the server program lifecycles
 - nav2_waypoint_follower: Can take in many waypoints to execute a complex task through
 - nav2_system_tests: A set of integration tests for CI and basic tutorials in simulation
 - nav2_rviz_plugins: An rviz plugin to control the Navigation2 servers, command, cancel, and navigation with
 - nav2_experimental: Experimental (and incomplete) work for deep reinforcement learning controllers
 - navigation2_behavior_trees: wrappers for the behavior tree library to call ROS action servers



ROS2 goals (Foxy) (Can we use something?)

- Support for navigating multi-story buildings #1226
- Behavior tree plugin support #958
- Add support for map zones #404, #401
- Traffic lanes and routes #1263
- Docking support #1225
- Voxel layer support #955
- Support multiple sensor inputs #1227
- Support for changing maps #239
- Waypoint follower demo #803

contd...

- Ensure a stable, functional SLAM implementation is available for ROS 2 (outside the nav stack)
- Ensure a stable, functional IMU/Odom fusing solution (eg [Robot Localization] (https://github.com/cra-ros-pkg/robot_localization)) is available for ROS2. (outside the nav stack) #40
- Provide an option/example for launching nav2 using composed nodes #1244
- Add Auto-localization Action server #1254

ROS2 roadmap (Visit the website)

<https://index.ros.org/doc/ros2/Roadmap/#port-of-existing-ros-1-functionality>

Behavior trees

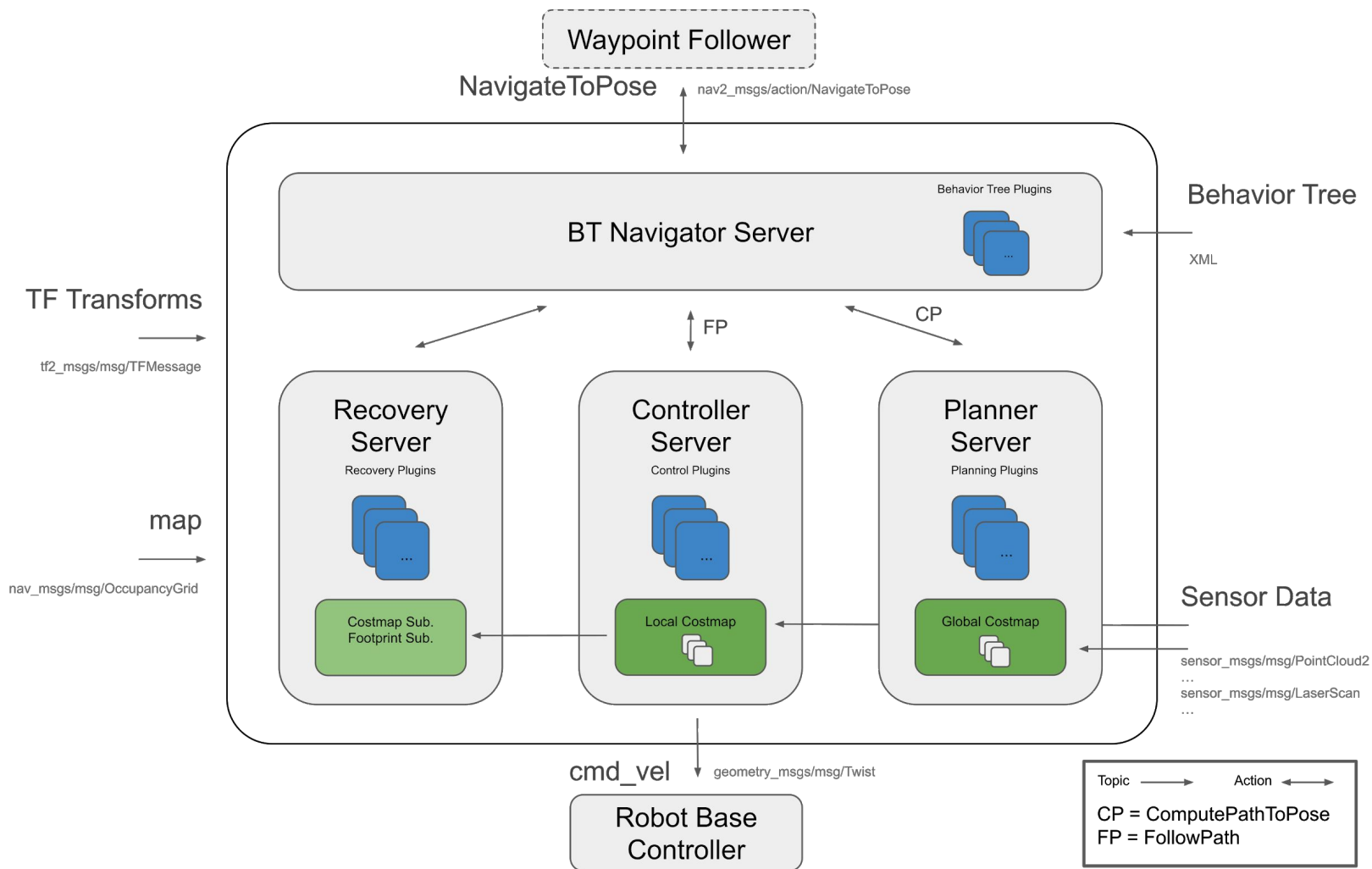
- Mathematical model of plan execution
- Graphically represented as a **directed tree**
- Describe switchings between a finite set of tasks in a modular fashion
- Create complex task; **without worrying how the simple tasks are implemented**
- How different from Finite state machines?
 - Main building block of a behavior is a **task** rather than a state.
 - **Easier to add other controls** without disturbing previous architecture

Detailed ROS2 Navigation

- Navigation 2 makes use of **action servers** and ROS2's low-latency, reliable communication to separate ideas
- Behavior tree is used to **orchestrate** these **tasks**: allows Navigation2 to have **highly configurable navigation behavior** without programming by rearranging tasks in a behavior tree xml file.
- Nav2_bt_navigator:
 - replaces move_base at the top level, with an Action interface to call the planner and controller with a configurable tree-based action model
 - Action Server and can also be **replaced** with other implementations
 - Behavior Trees to make it possible to have more complex state machines and to add in recovery behaviors as **additional Action Servers**

contd...

- planning, recovery, and controller servers are also action servers that the BT navigator can call to compute.



Ros2 navigation stack- what parameters it can take?

What else can we put?

Nav2_amcl: probabilistic localization module

amcl takes -> laser-based map, laser scans, and transform messages

Outputs -> pose estimates.

(future plan - Extending AMCL to work with different type of Sensors

Improving overall localization performance with fusing data from different sensors such as IMU, Sonar, Radar, Laser, Depth camera, and etc.)

Nav2_behavior_tree:

integrating ROS2 actions into Behavior Trees

Navigation-specific behavior tree nodes

Nav2_controller:

designed to be loaded with plugins for path execution

plugins need to implement functions in the virtual base class

Nav2_msgs:

[BehaviorTreeLog.msg](#)

[BehaviorTreeStatusChange.msg](#)

[Costmap.msg](#)

[CostmapMetaData.msg](#)

[VoxelGrid.msg](#)

contd...

System Tests:

Test the integration of several components and subsystems.

- Support modules (Mapping, Perception, Prediction, Localization)
- Navigation core (Navigator, Planner, Controller)
- Support modules and navigation core
- Command chain (Mission Planning, Mission Execution, Navigation System, Robot Interface)

Messages exchanged

Sensor, actuator messages

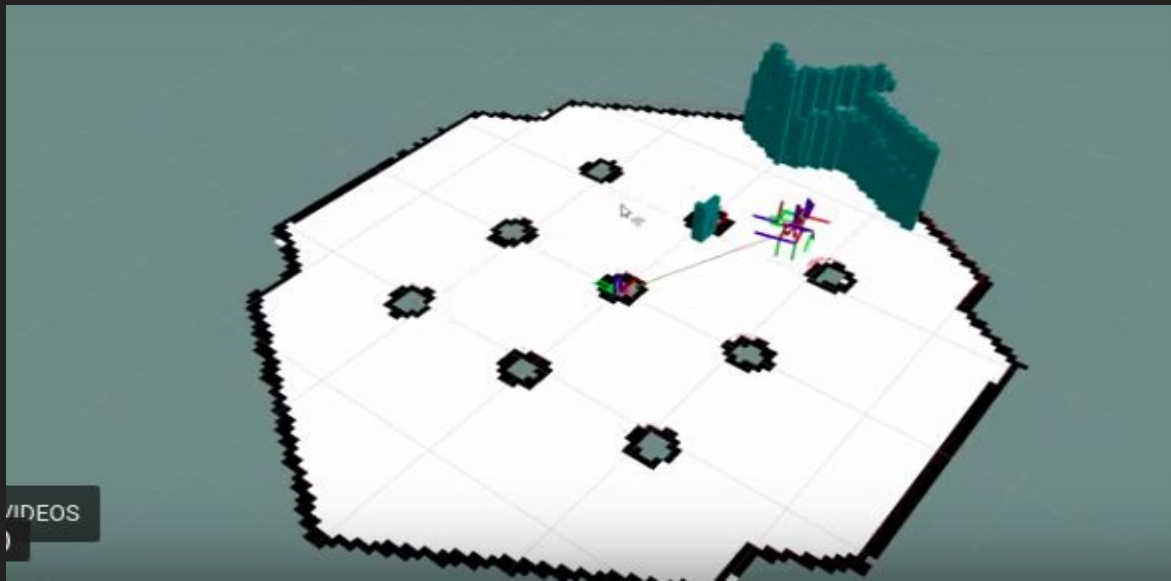
- sensor_msgs/Range
- sensor_msgs/NavSatFix
- sensor_msgs/Imu
- sensor_msgs/FluidPressure
- sensor_msgs/Temperature
- sensor_msgs/BatteryState
- sensor_msgs/RelativeHumidity
- /pc_cylinder/water_detected
[std_msgs/Bool]
- geometry_msgs/PoseStamped
- sensor_msgs/Joy
- nav_msgs/Odometry]
- geometry_msgs/Vector3Stamped
- cola2_msgs/Setpoints
- cola2_msgs/BodyForceReq
- diagnostic_msgs/DiagnosticArray
- cola2_msgs/DVL
- cola2_msgs/Float32Stamped
- cola2_msgs/VehicleStatus
- cola2_msgs/RecoveryAction
- cola2_msgs/BodyVelocityReq
- cola2_msgs/NavSts
- cola2_msgs/CaptainStatus
- cola2_msgs/CaptainStateFeedback
- visualization_msgs/Marker
- Barometer, joystick, leak sensor, depth, maestro polulo motors

Add ons with ROS2

- SLAM
- Reinforcement Learning in obstacle avoidance and navigation
- Vision based DL plugins
- Robot Security

(STVL) Using an External Costmap Plugin

- STVL (another 3D perception plugin similar to the Voxel Layeris), a demonstrative **pluginlib plugin** and the same process can be followed for other costmap plugins as well as **plugin planners, controllers, and recoveries**.



Conclusion

With behavior trees in navigation2, we have flexibility to add plugins without disrupting the current architecture

We may utilize these with ROS2 Navigation stack

What Next?

We have study of various marine robots, ros messages used in them, our own rosswan drivers which could be used with ros2 and full understanding of ros2 navigation stack