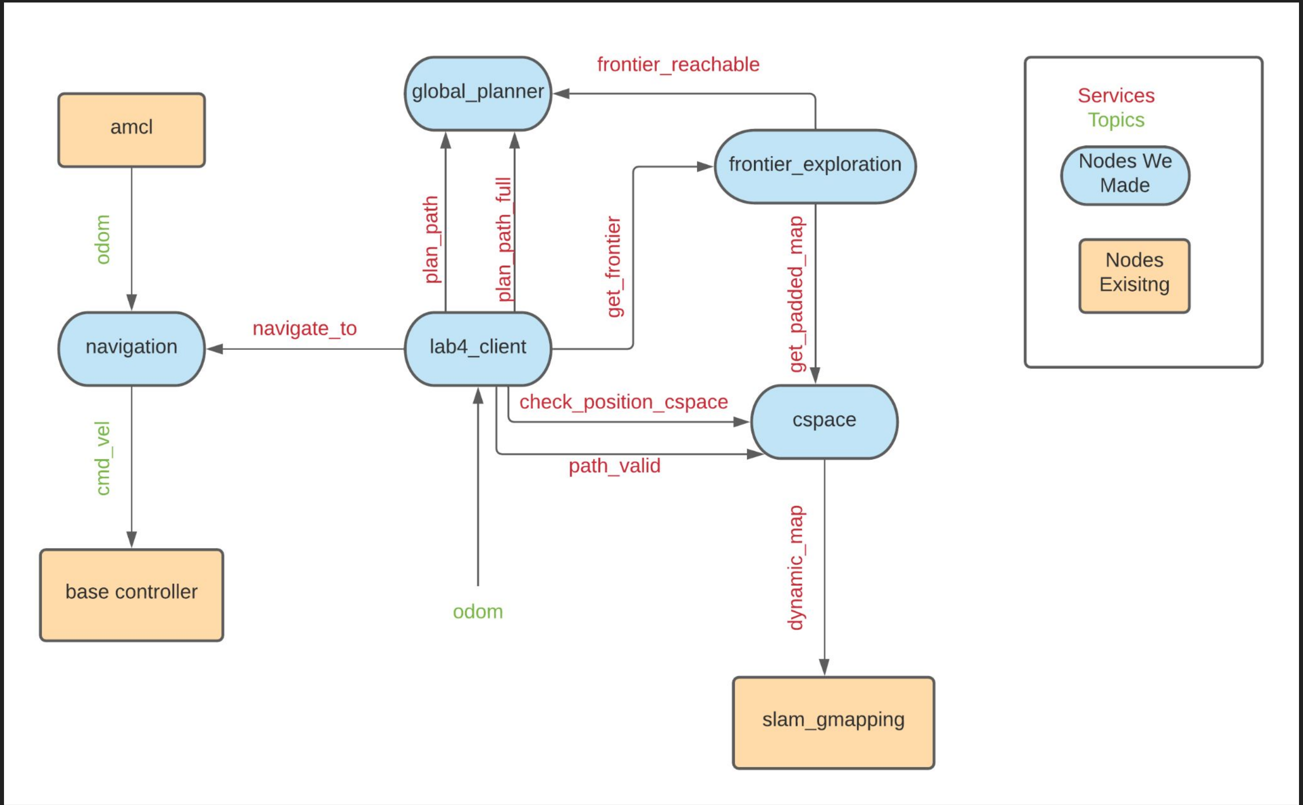
RBE 3002 CDR

Team 7

Logic Flowchart



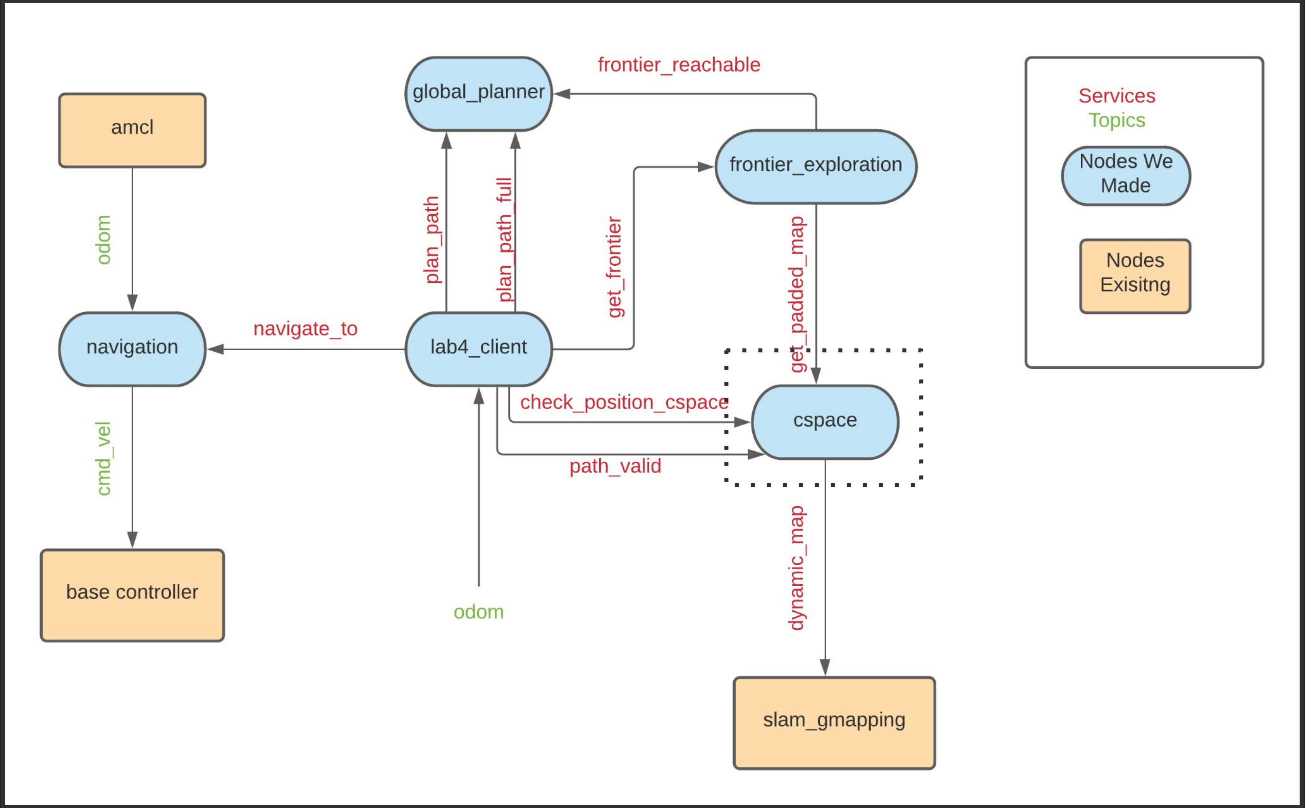
Lab 4 Client

- Contains the state machine that manages the first two phases of the final challenge
- Contains service requests for services offered by the other nodes to get the necessary information to fulfill tasks. Service proxies include for services:
    - *check_position_cspace*, *navigate_to*, *path_valid*, *plan_path*, *plan_path_full*, *get_frontier* and *frontier_reachable*,
- **The top level code for Phase 1 and Phase 2**

Lab 4 Client

- PHASE_1
    - STORE_START_LOCATION
        - Stores initial pose. This is the goal pose for Phase 2
    - CHECK_POSITION
        - Checks if current pose of robot is not in CSpace (aka current pose is valid), then switches to GET_FRONTIER.
        - Otherwise, navigates out of CSpace.
    - GET_FRONTIER
        - Gets goal frontier to navigate to next. If no more frontiers, sets phase_state to PHASE_2
    - PLAN_PATH
        - Plans path to goal frontier. Stores full path and optimized path in global variables.
    - NAVIGATE_PATH
        - Continuously checks to see that full_path is a valid path (i.e. that is not in CSpace).
        - If it is, navigates to incrementing optimized waypoints of path until no longer valid path or path is complete.
        - Goes back to CHECK_POSITION at the end.
- PHASE_2
    - Plans path to start pose (stored in Phase 1) and navigates to it.

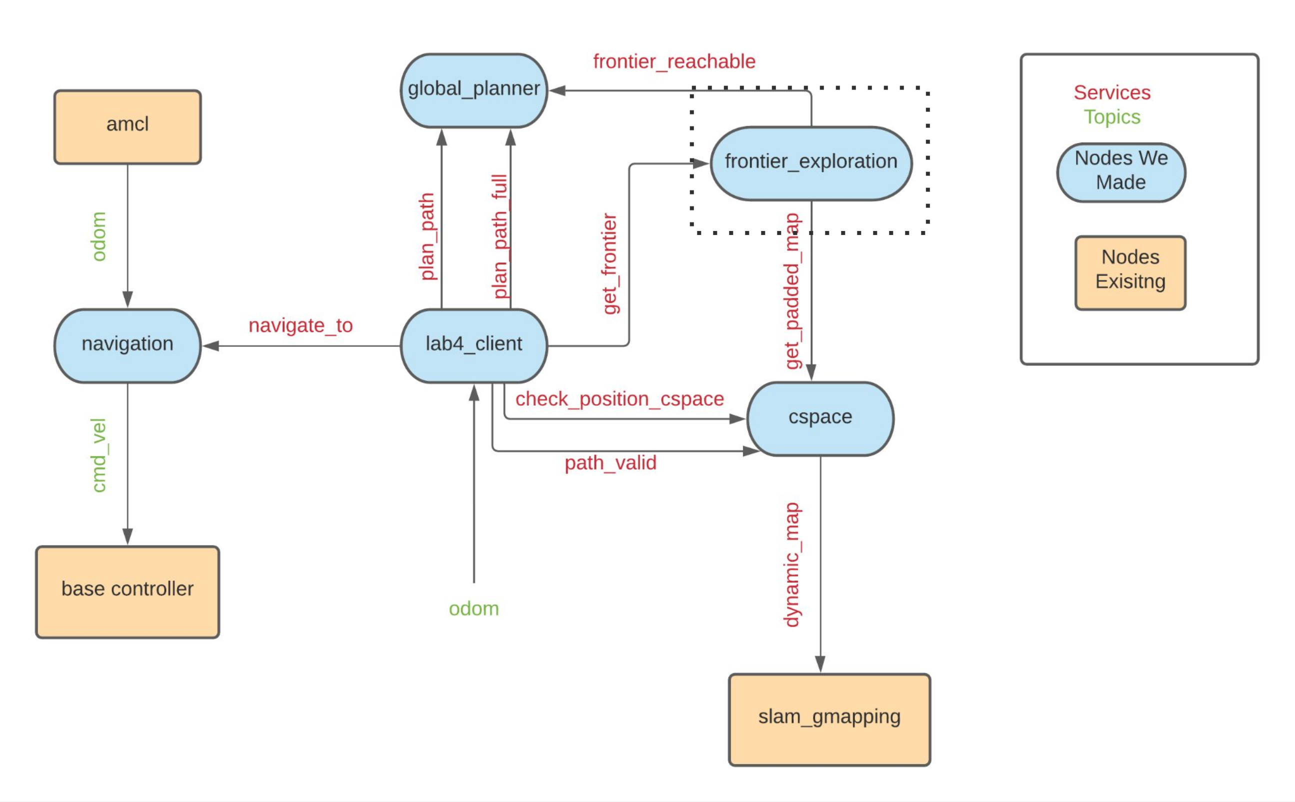
Logic Flowchart



CSpace Node

- Adds padding to raw map from slam_gmapping
- Requests */dynamic_map* service from slam_gmapping node
- Offers service */get_padded_map* with service handler **getCSpace()**
- Offers service */check_position_cspace* with service handler **pose_valid_callback()**.
    - Custom service type is CSpaceValid.srv.
    - Service callback determines whether pose sent (in request) is in the CSpace or not.
    - If pose sent (in request) is in CSpace, it finds the nearest valid (not in CSpace) position.
    - Response message is pose (valid pose) and isValid boolean (whether sent pose in CSpace or not).
    - Uses breadth-first search (BDS) to find nearest available position.
- Offers service */path_valid* with service handler **path_valid_callback()**.
    - Custom service type is PathValid.srv.
    - Returns boolean that indicates whether path sent is valid or not.

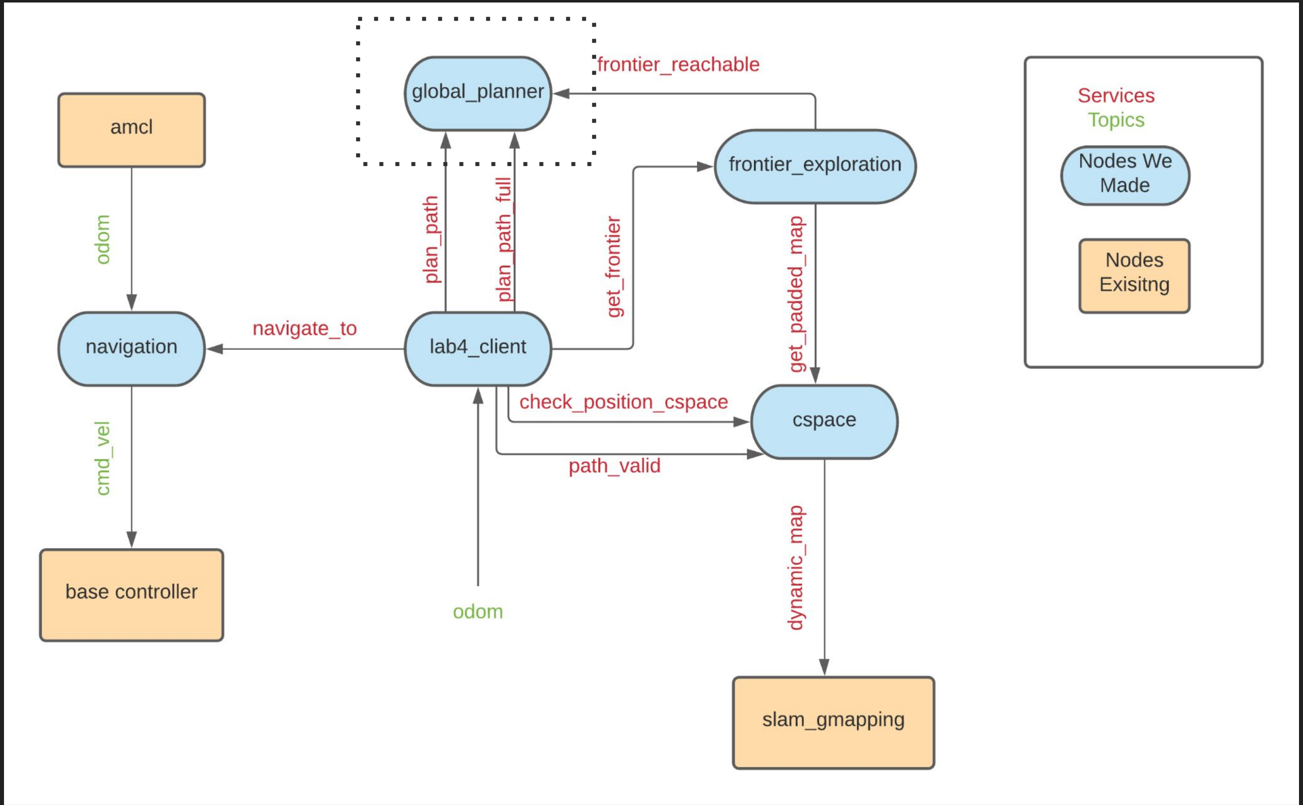
Logic Flowchart



Frontier_Exploration node

- Finds edges and segments edges to bins to group frontiers, and finds centroids and priorities.
- Requests service */frontier_reachable* from global_planner node
- Offers /get_frontier service, with service handler **getFrontier()**.
    - Uses custom GetFrontier.srv service type.
    - **getFrontier()**:
        - Calls **edgeDetection()**, **segmentFrontiers()**
        - Populates a priority queue with median of frontiers and their priorities based on length and distance.
            - Did not use centroids as per recommendation in PDR.
        - Returns goal frontier and boolean that indicates whether frontier exploration is finished or not.

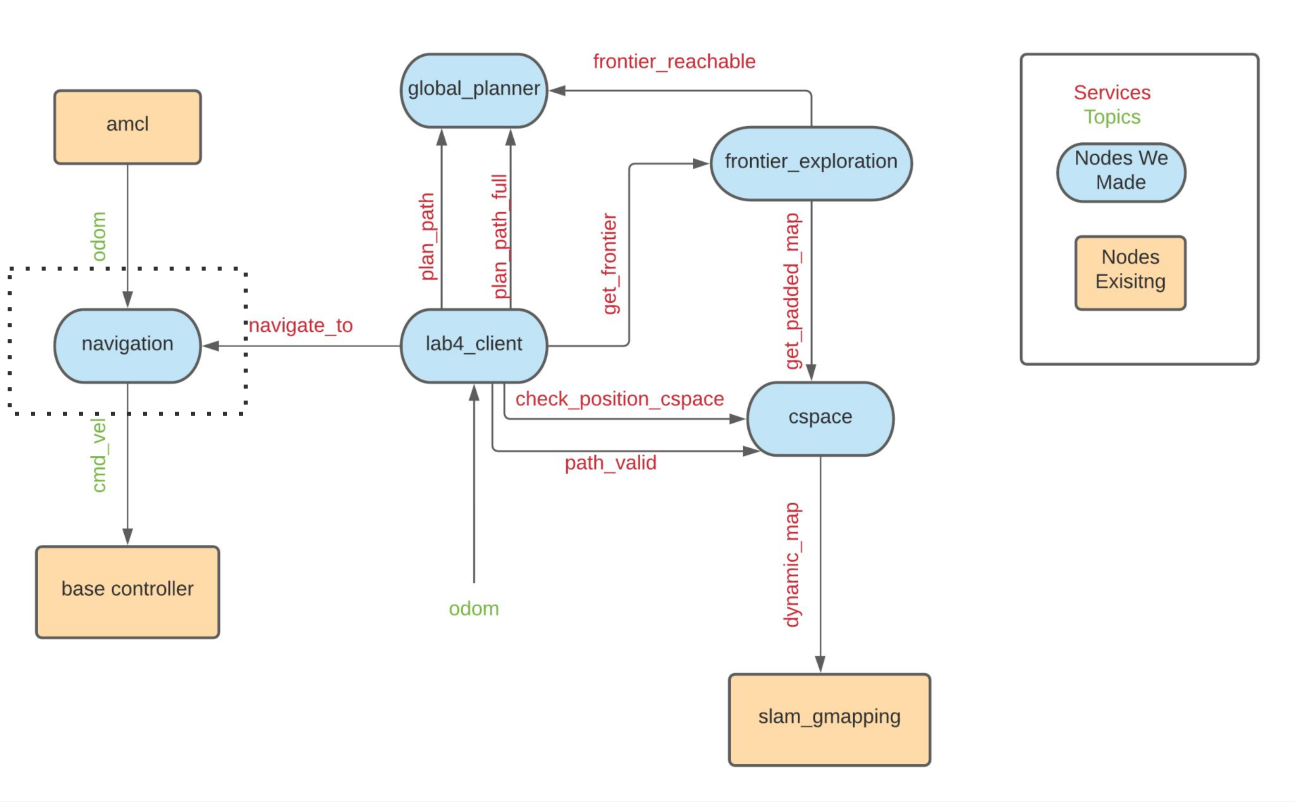
Logic Flowchart



Global_planner node

- Offers path_planner service, which calculates A* and returns an optimized path only.
- Offers plan_path_full service, which calculates A* and returns a full and an optimized path.
  - Custom service type GetFullPlan.srv created.
- Offers frontier_reachable service, which returns if a given point (i.e. frontier) is reachable or not.
  - Reachable means if there is a valid path towards it or not.
- A* algorithm adjustments
  - Added extra weighting to cells next to CSpace so robot does not plan path right next to CSpace. (Using neighbors of eight in CSpace)

Logic Flowchart



Navigation node

- Offers */navigate_to* service with service handler **go_to()**.
  - Custom service type is NavigateTo.srv
  - The service handler go_to() combines drive() and rotate() to reach the goal pose.

AMCL node

- Run node only for Phase 3.
- Updates odom based on map observations.