PRD  
VIRTUAL WALLS

short line

# API - TO/FROM ROBOX

## Input method:

1. Single Message - explicitly give away layer inputs and coordinates
   1. URI: base.command.traversability
   2. Topic: TRAVERSABILITY\_LAYER\_LIST\_IN ("/traversability\_layer\_list\_in")
   3. Input msg: TraversabilityLayerList.msg\*
2. Action command - start / stop drawing virtual wall using robot’s position
   1. Topic (temporary): “/syscommand
   2. Input msg - in main\_commnand:
      1. Start drawing virtual wall: *“draw\_wall\_start <id (int)>”*
         1. Example: *“draw\_wall\_start 1”* - will draw new wall with id 1 (or replace old one with the same id)
         2. *“draw\_wall\_start” will draw wall will id=”0”*
      2. Stop drawing virtual wall: *“draw\_wall\_stop”*

## Output method:

* Single msg - as a response
  1. URI: Base.info.traversability
  2. Topic: TRAVERSABILITY\_LAYER\_LIST\_OUT ("/traversability\_layer\_list\_out")
  3. Output msg:TraversabilityLayerList.msg\*

\* traversabilityLayerList.msg:

* status (int)
  + SET\_DESIRED = 0
  + GET\_CURRENT = 1
  + COMPLETE = 2
  + FAILED = 3
* status\_description (string)
* traversability\_layer\_list (Traversability layer[]\*\*) - all required virtual walls
* map\_id (string)

\*\* Traversability layer:

* layer\_creation\_universal\_time - universal time (int):
* TraversabilityCategory: (int):
  1. FEASIBLE = 0
  2. OBSTACLE = 1
  3. GOT\_STUCK = 2
  4. **VIRTUAL\_WALL = 3**
* Id - name (string)
* layer\_poses (geometry\_msgs::Pose2D[]\*\*\*)
* Thickness [meter] (float)

\*\*\* geometry\_msgs::Pose2D:

* float64 x
* float64 y
* float64 theta

# FLOW

Steps:

1. Robox get virtual wall input
   1. ‘Command Parser’ sends it to ‘navigation node’
   2. ‘Navigation Node’ gets the input and process it properly:
      1. \* Consider implement virtual wall both/only pure\_localiztion/slam modes
      2. ‘Navigation node’ publishing ‘traversability layer’ to ‘Local Map Node’
         1. Frequency: slam dependent - currently 1 [hz]
      3. Local map process information and project on map
      4. Both VC (velocity check) and ‘path plan node’ get information from local map

# Limitations

1. Consider cases:
   1. Does ROBOX allowed to get 1 point as virtual wall?
   2. Does ROBOX have a virtual wall minimum / maximum length?
   3. Priorities
      1. Virtual wall
      2. Green path
   4. Virtual wall in follower map? Skidjoy? moveby?
2. Does virtual wall work on SLAM mode? On pure\_localization mode
3. Should robox send out it’s virtual walls list as an interval? Or should it just draw it on the map and send out the map as usual?

# 

# Crucial Implementations notes

1. Virtual wall inflation on map should be parameter (maybe get as an input?)
2. Virtual wall we be implemented in insteration by ‘illegal state’ in ‘navigation node’ - which means it will be assigned to trajectory nodes that are closest to its location

# Tasks:

1. Implement virtual wall in SLAM (navigation node) - shir
2. Implement virtual wall in local map - shir
3. Implement API to and from ROBOX - Tal
4. Implement API inner ROBOX - shir/tal

# EXTRA Shopping List - TDL (check list):

* Local map:
  + Assign virtual wall with its own obstacle type and report correctly to VC
  + Paint virtual wall with its own color in map for debugging purposes
  + Create ON/OFF bottom (custom command) for virtual walls
  + Easy control from configuration files on virtual wall thickness [meters]
* Robot Manager:
  + Parse virtual wall obstacle to obstacle status
  + Update the launcher calculating msgs
* SLAM:
  + Create ON/OFF bottom (custom command) for virtual walls
  + Easy control from configuration files on virtual wall thickness
* ROBOX:
  + Start / stop virtual wall according to robot’s current position
* API
  + Update postman API
  + Add custom command for short list:
    - *draw\_wall\_start*
    - *draw\_wall\_stop*

short dash

\* Response with id