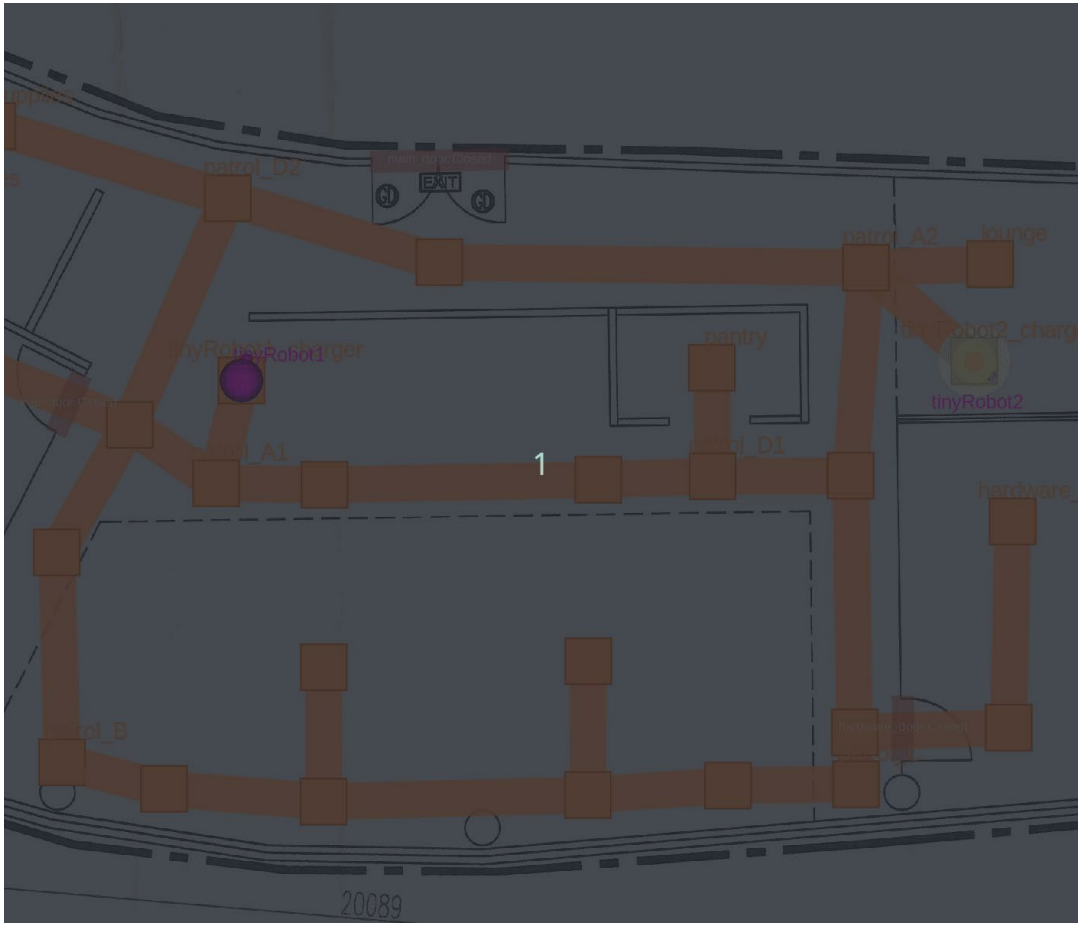
Open-RMF and the Challenge of Resource Contention in Large-Scale Robot Fleets

Arjo Chakravarty
arjoc@intrinsic.ai


patrol_D2
EXIT
patrol_A2
lounge
pantry
tinyRobot1
tinyRobot2
1
patrol_A1
patrol_D1
hardware
patrol_B
20089

Open-RMF

Currently available Open-RMF features

- ✅ Traffic Management
- ✅ Task allocation
- ✅ Task scheduling
- ✅ Charging Schedules
- ?? Resource allocation

Open-RMF

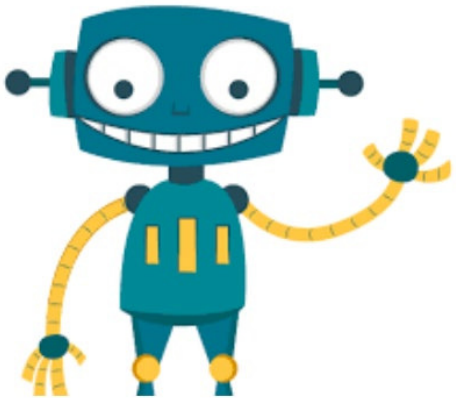
Resource Contention and Robots

- Lifts
- Parking spots
- Chargers
- Tool heads
- Carts

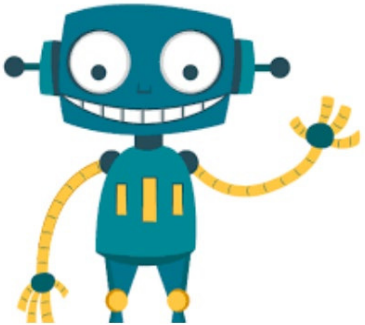
Open-RMF

Resource deconfliction?

I need to charge. I can charge at charging station 1 or station 2

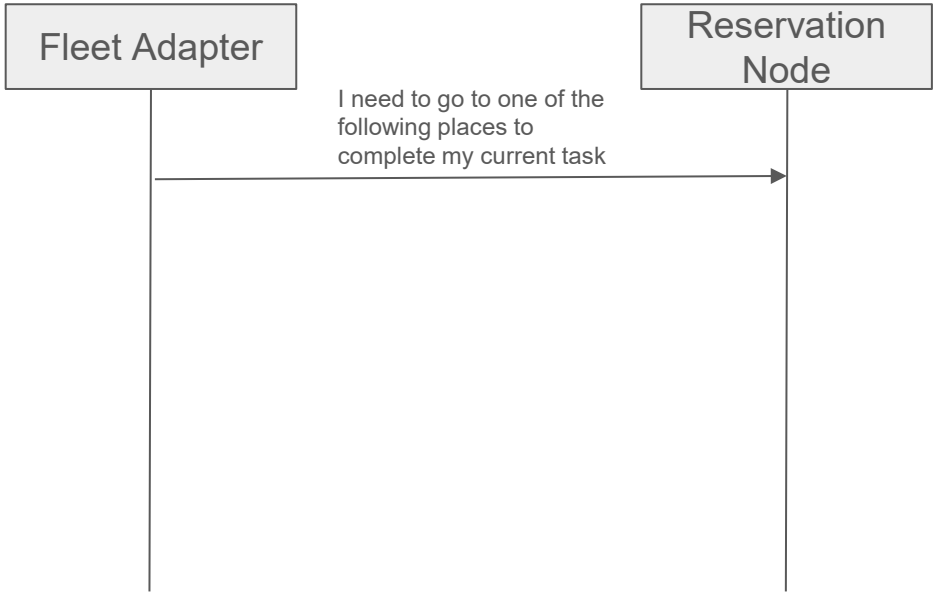


I need to charge. I can charge at charging station 2

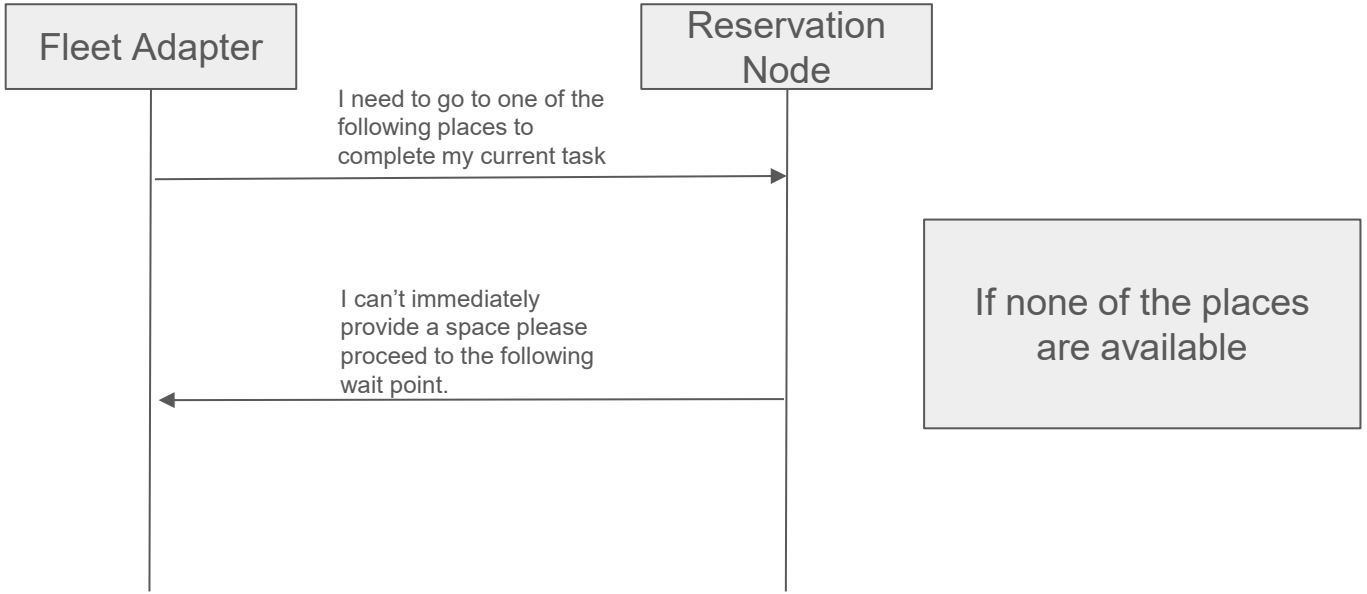


Open-RMF

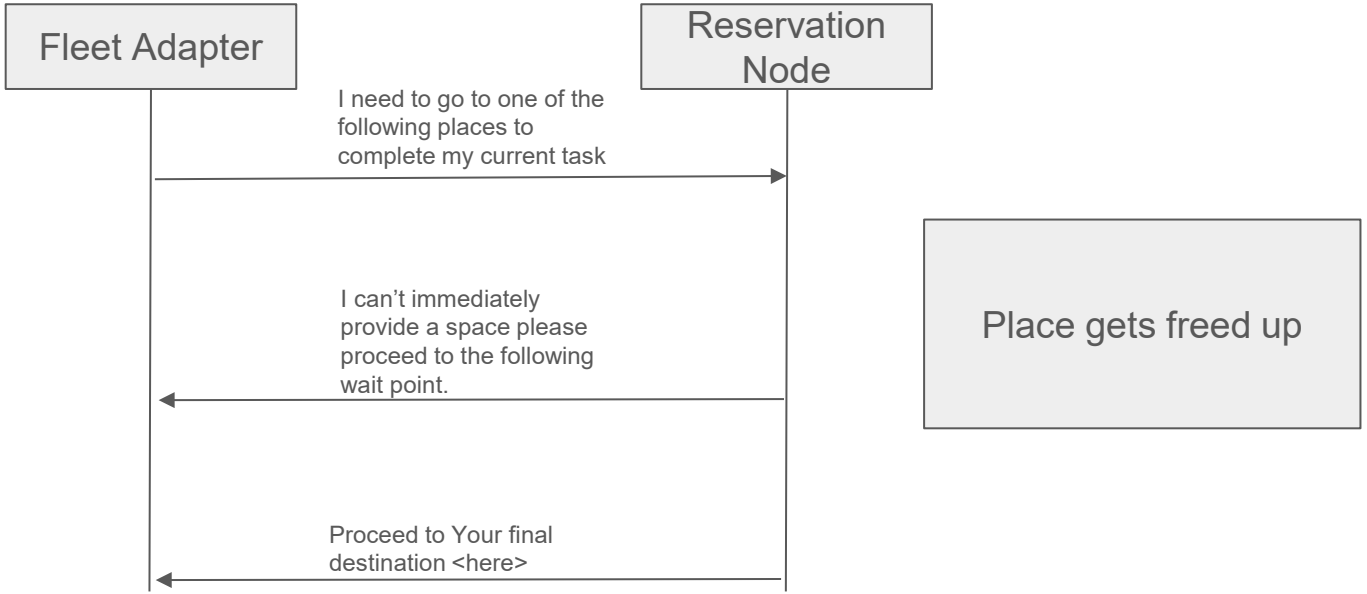
Reservation Node in RMF



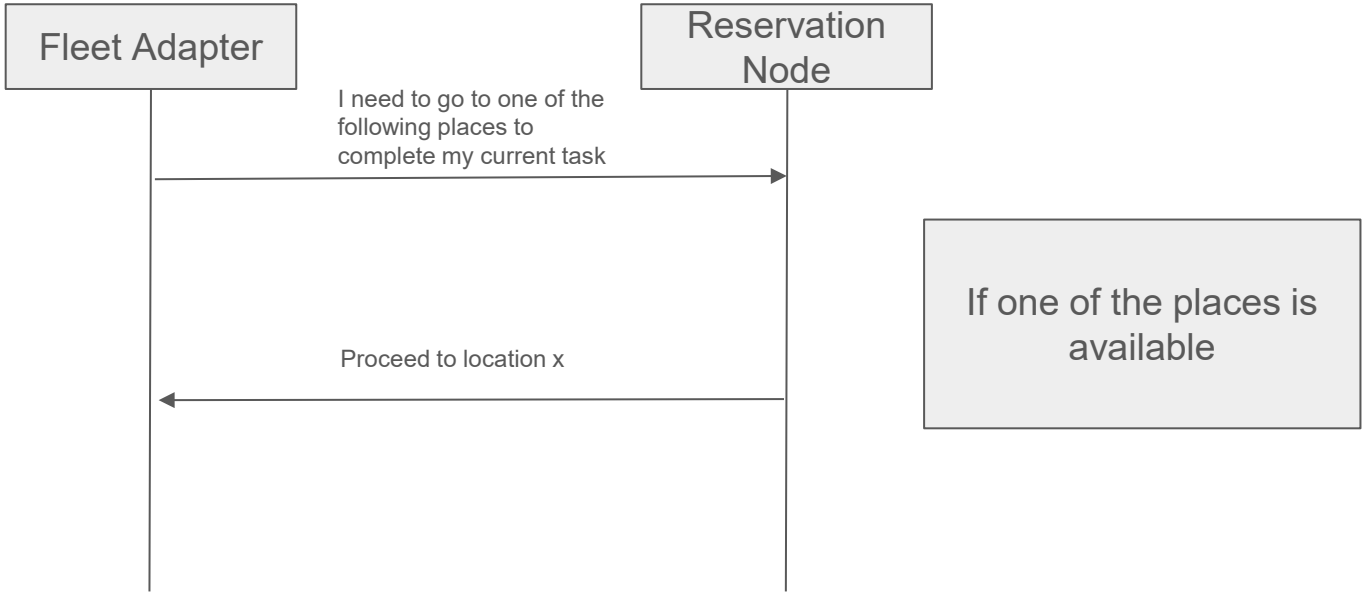
Reservation Node in RMF



Reservation Node in RMF



Reservation Node in RMF



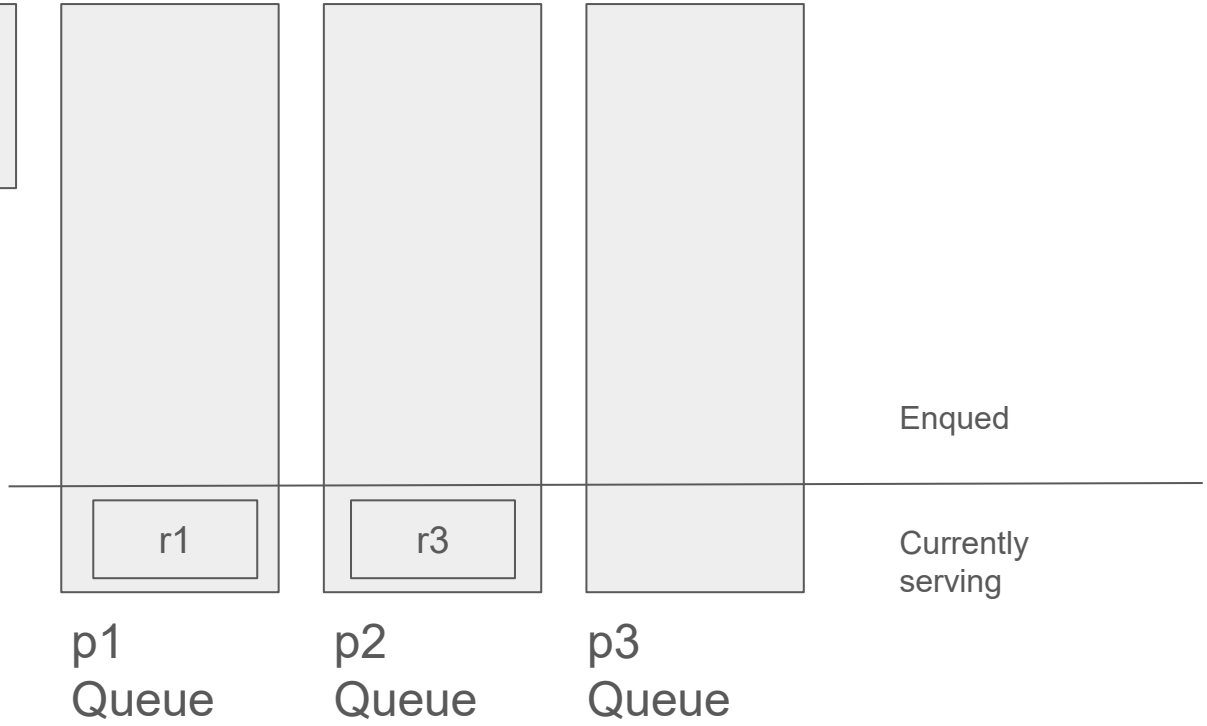
How it works internally

Robot 2 has task:

```
GoToPlace {
        one_of: [p1, p2]
}
```

How it works internally

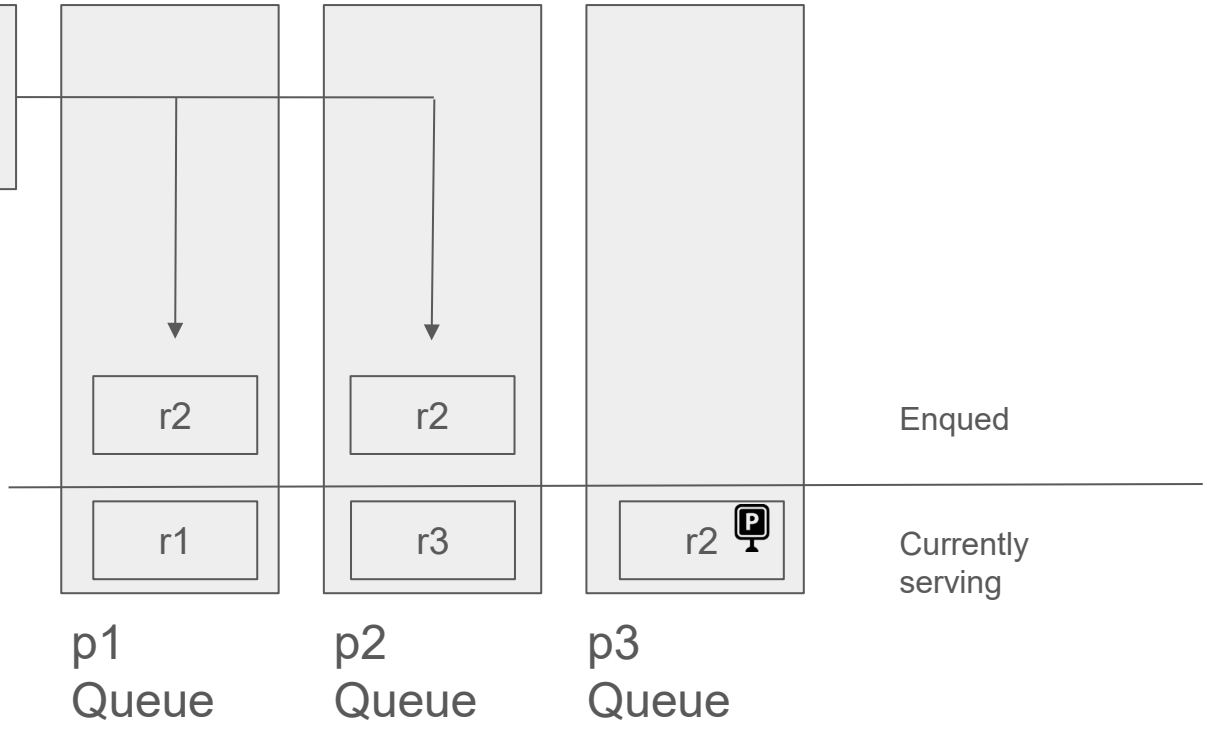
```
GoToPlace {
        one_of: [p1, p2]
}
```



How it works internally

```
GoToPlace {
        one_of: [p1, p2]
}
```

No free spot we add the request to the queue and assign a free spot.

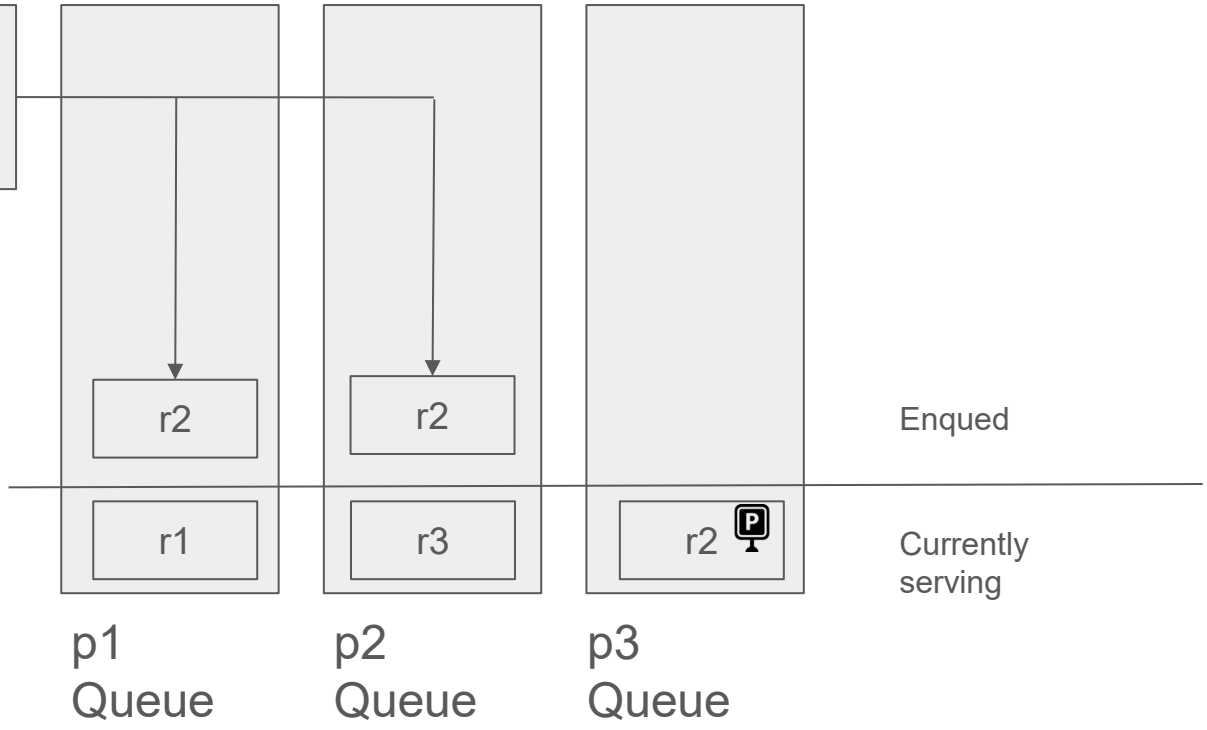


Open-RMF

How it works internally

```
GoToPlace {
        one_of: [p1, p2]
}
```

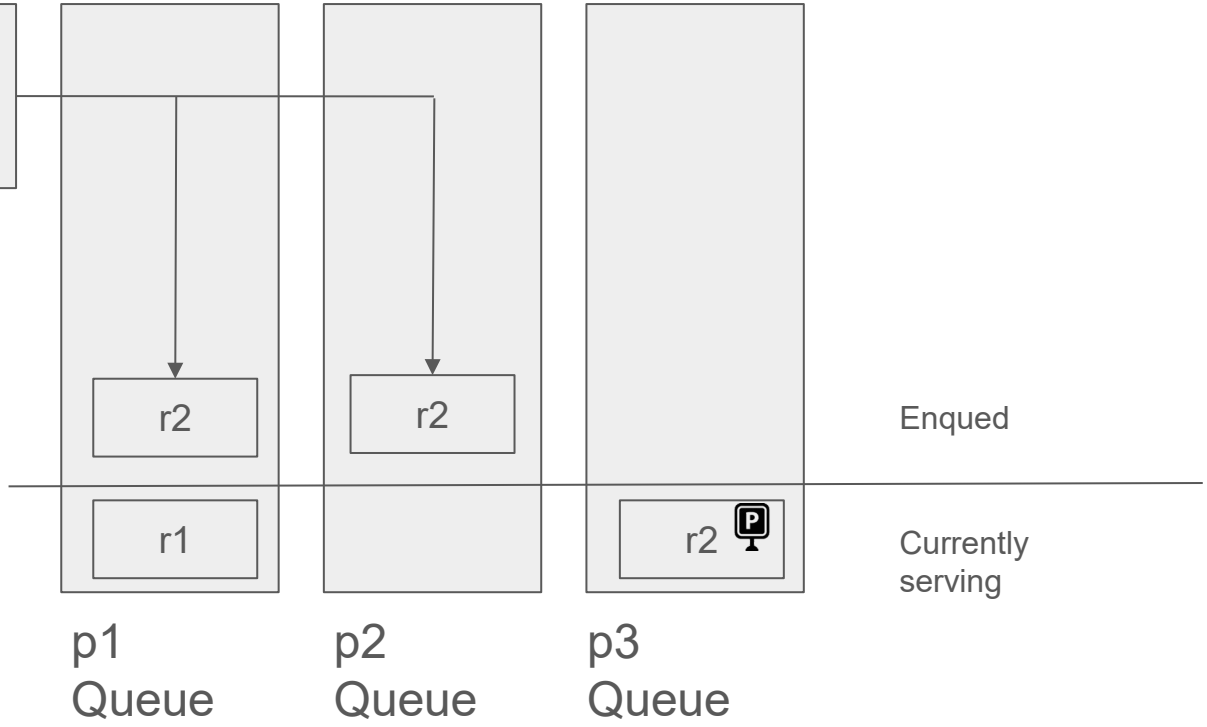
Lets say r3 finishes its task.



How it works internally

```
GoToPlace {
        one_of: [p1, p2]
}
```

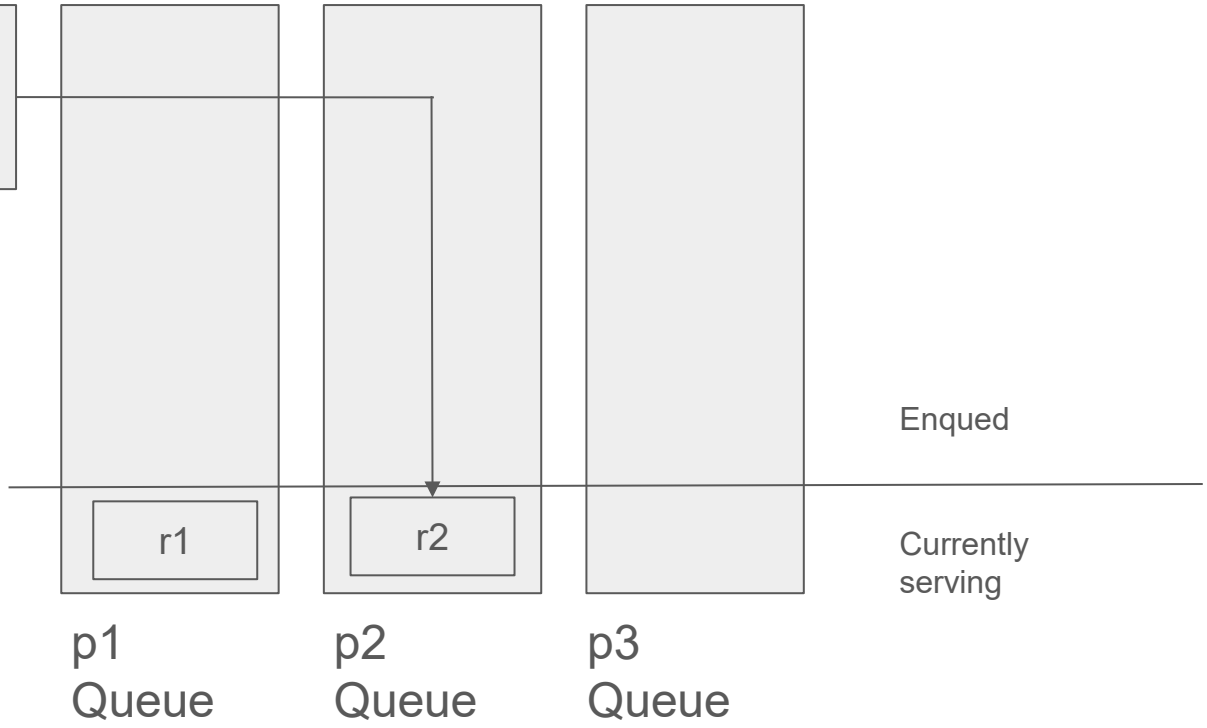
Lets say r3 finishes its task.



Open-RMF

How it works internally

```
GoToPlace {
        one_of: [p1, p2]
}
```

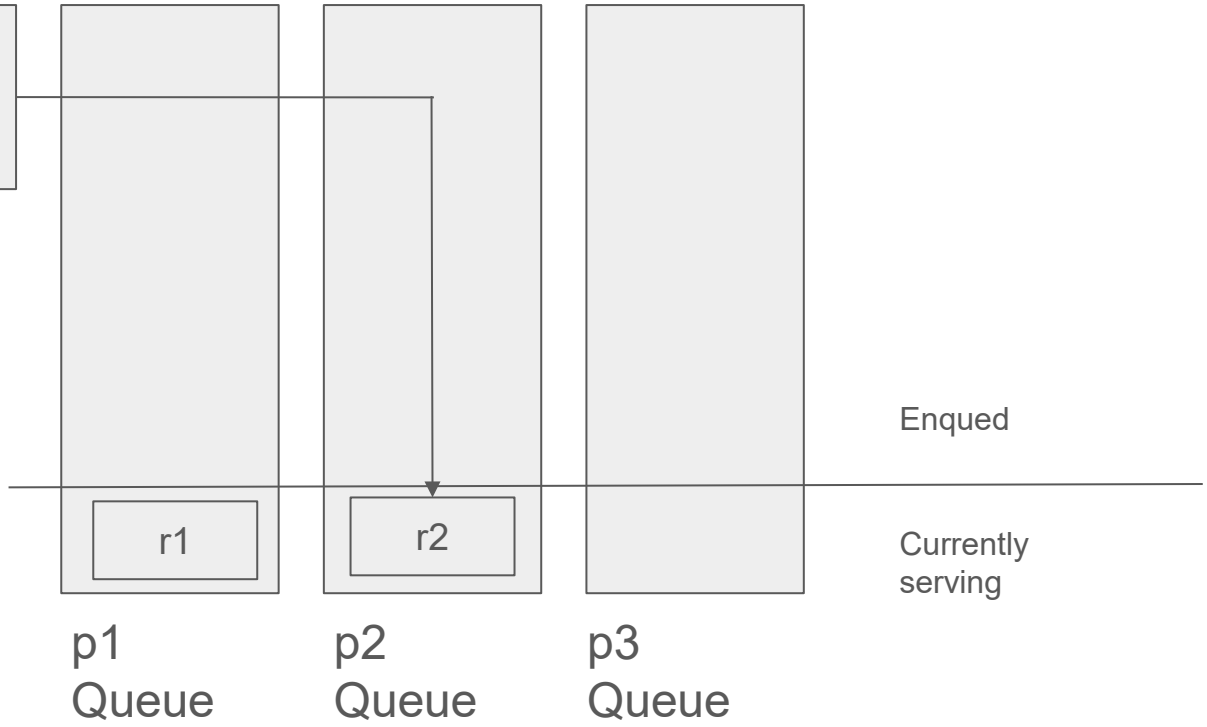


Lets say r3 finishes its task.

We remove all Pending r2 instances.

How it works internally

```
GoToPlace {
        one_of: [p1, p2]
}
```



Lets say r3 finishes its task.

We remove all Pending r2 instances.

Note: If there are dependencies, we will traverse the wait graph.

Open-RMF

Caveats

- There must be at least the same number of parking spots as robots available.
- The node must be explicitly enabled.
- The queue is FiFo

Open-RMF

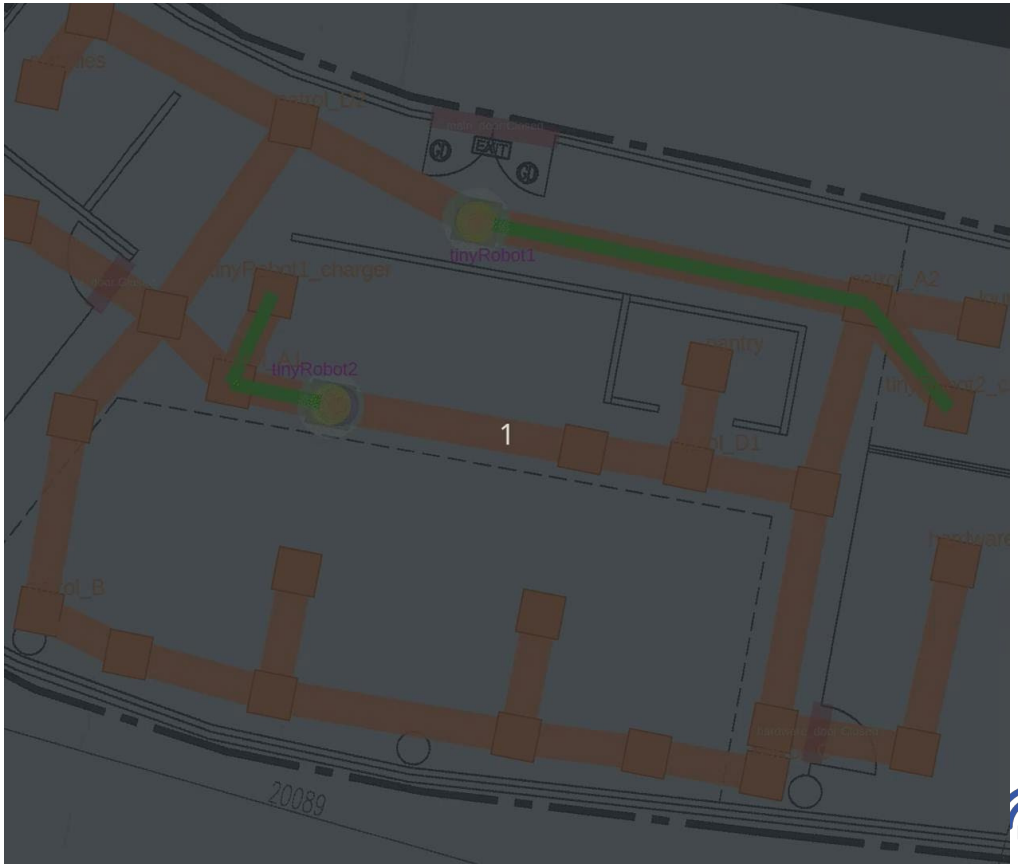
Video Demonstration

In this example tinyRobot2 is asked to go to the pantry first.

tinyRobot1 is told it can go to Either the lounge or the pantry.

The reservation node correctly allocates each robot to its final destination.

Video Demonstration



To Enable

In your fleet configuration:

```
rmf_fleet:
  name: "tinyRobot"
  limits:
    linear: [0.5, 0.75] # velocity, acceleration
    angular: [0.6, 2.0] # velocity, acceleration
  …
  use_parking_reservations: True
```

Open-RMF

To Enable

Also launch the reservation queue manager

```
<node pkg="rmf_reservation_node"
    exec="queue_manager"></node>
```

Example Config Available in this PR:

https://github.com/open-rmf/rmf_demos/pull/212/files

Open-RMF

Sneak Peek Of Next Gen Capabilities

Open-RMF

Motivation

- We can check resource constraints at runtime but what if there are tasks that have to be completed by some deadline?
- Example could be that there are robots waiting for their next charge blocked by a counterpart that is reusing the same space for some other task.

Introducing `rmf_reservations`

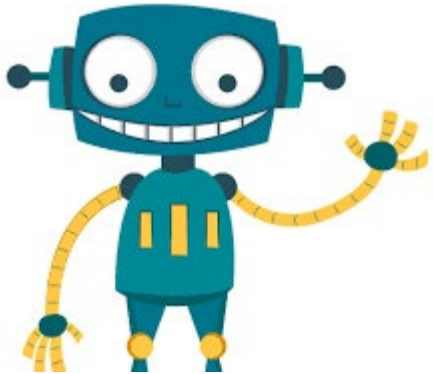
- Rust-based resource constrained schedule library
- Can be used at the task scheduling level
- Currently proof of concept is ready but exact APIs are unstable and **will change**
- Submit a set of resource constraints and requests. Get back feasibility immediately. Find optimal solution slowly.
- Paper presented at IROS 2024 outlining the algorithms.
- Unlike Google-OR tools our formulation does not need integer costs.

Open-RMF

Next generation capabilities

I need to charge. I can charge at charging station 1 from 3:00 to 4:00 or station 2 from 3:20 to 4:20

Alternatives



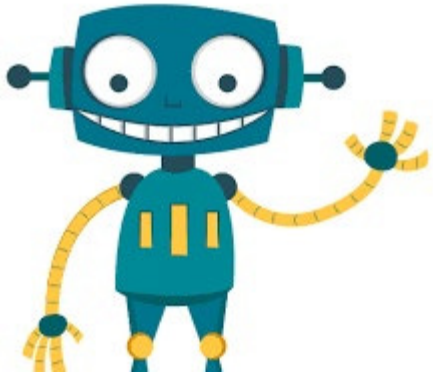
Alternatives can have costs. For instance we can use distance to charger as a cost function.

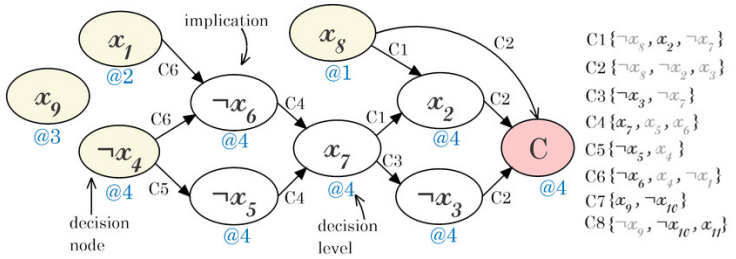
Open-RMF

Next generation capabilities

I need to charge. I can charge at charging station 1 from 3:00 to 4:00 or station 2 from 3:20 to 4:20

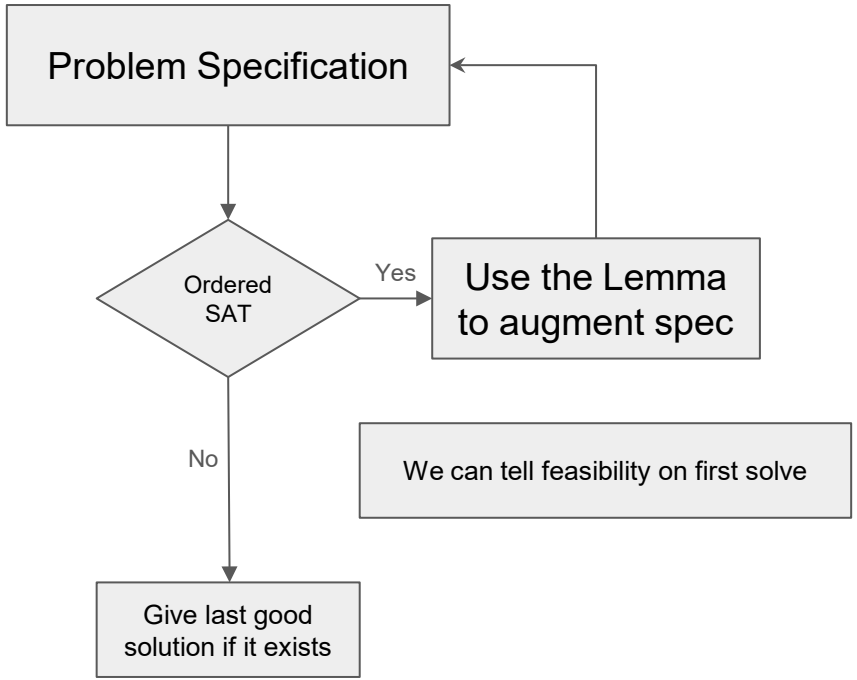
Alternatives





Open-RMF

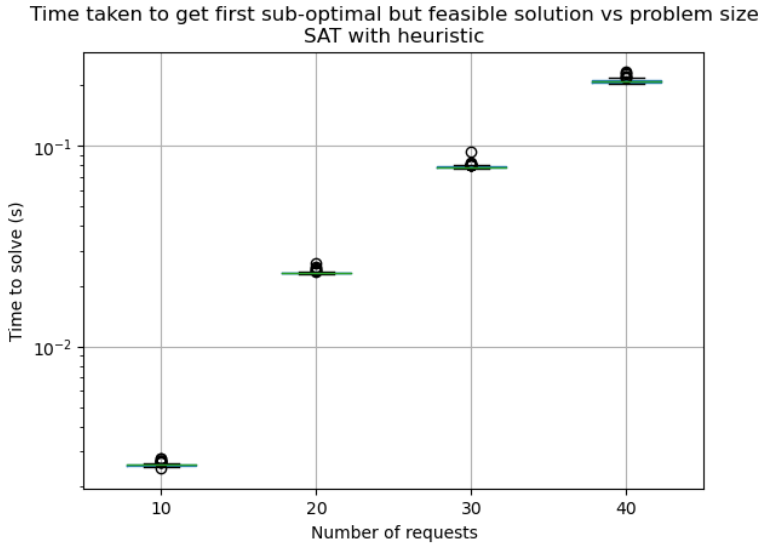
Next generation capabilities



Arxiv Link
To Paper

Open-RMF

Next generation capabilities



Feasibility via SAT For problems with obvious solutions. For a request size of n there are n alternatives. I.E for 40 requests there are 40 alternatives each

Questions?

PR
Documenting
How to use in
current
generation

RMF
Reservations
Source code

Arxiv Link
To Paper

Open-RMF