

自律移動ロボット（AMR）の Open-RMFへの対応方法

26st/Sep/2023

グエン ジュイヒン

Panasonic

Panasonic Asia Pacific Co. Ltd

目次

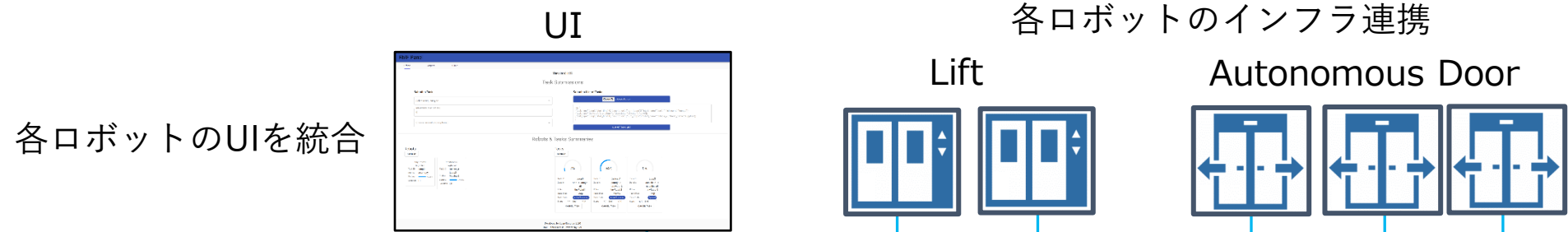
RMFの開発された背景

RMFの機能

RMFを使うにあたって

Robotics Middleware Framework (RMF)のコンセプト

全ての異業異社のAMRを一つの集中管理システムでコントロールし、施設内の全てのインフラ連携を可能にする



第2レイヤー

The RMF

Multi-Fleet Management System

ROS Health packages

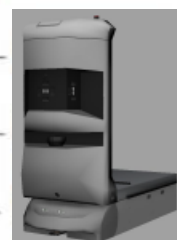
System Of Systems Synthesiser

第1レイヤー

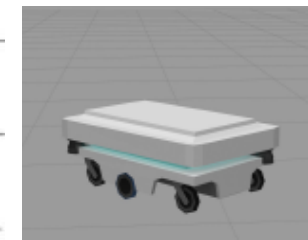
Fleet Management System



AGV System by
Robot Vendor A



AGV System by
Robot Vendor B

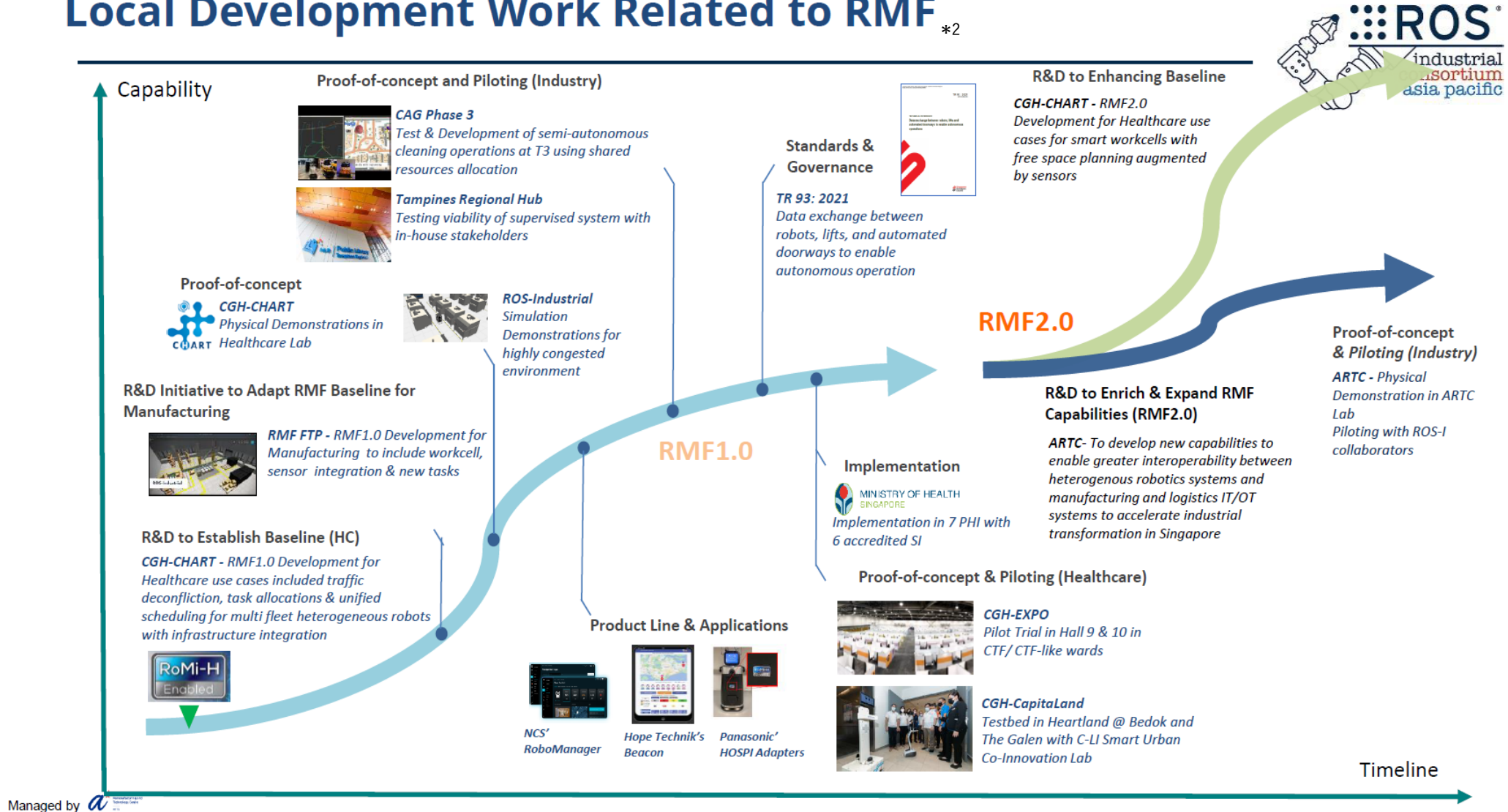


AGV System by
Robot Vendor C

各ロボットの交通整理

シンガポール政府によるRMFの開発ロードマップ^{※2}

Local Development Work Related to RMF ^{*2}



Managed by A-STAR

^{*2} : This roadmap was provided by Advanced Remanufacturing and Technology Centre <https://www.a-star.edu.sg/artc>

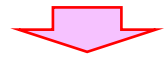
シンガポールにおけるロボット導入マップ



RMFの必要性

Changi General Hospitalでのケース

- 各ロボットは各々フリートマネジメントシステムを使用。
⇒各ロボットの現在位置は共有化されていないため、ロボットの交通渋滞が発生する。
- ⇒各ロボットはそれぞれ別々のエレベーターを占有。
各社のロボットは互いのエレベーターには乗れない。
新たに導入したロボットにエレベーターを割り振れないため、そのロボットはフロア間移動ができない。



ロボットの社会実装の障壁になっている

ロボットの交通渋滞



Panasonic



Swisslog



お掃除ロボット



各ロボットは各専用のエレベーターを使用している*1

お掃除ロボットは自動で階移動できず、マニュアルで移動

*1 : Swisslog uses an AGV lift, and Panasonic uses a staff lift for each AGV.

目次

RMFの開発された背景

RMFの機能

RMFを使うにあたって

RMFの詳細

- RMFはROS2で構成されており、誰でも使用可能

RMFのソース

<https://github.com/open-rmf>

RMFのデモ環境

https://github.com/open-rmf/rmf_demos

評価環境の作成

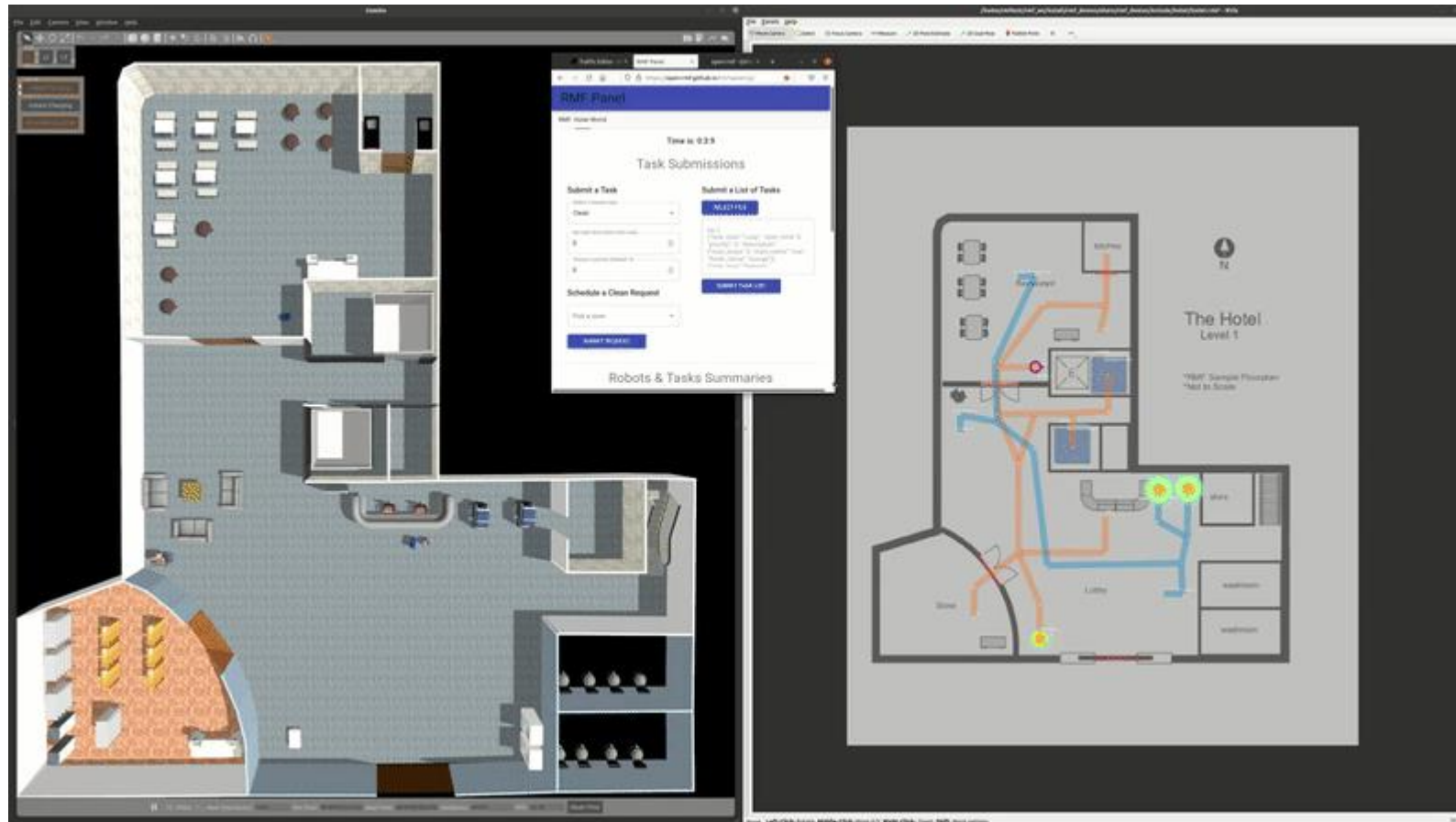
<https://osrf.github.io/ros2multirobotbook/traffic-editor.html>

RMFのドキュメント

<https://osrf.github.io/ros2multirobotbook>

- 実際のロボットの動き
(Gazebo)

- RMFで制御されているロボットの様子
(Rviz)



(参考) RMFを使った3次元シミュレーション環境の作成

RMFではロボットを実際の現場に導入する前に事前に仮想環境でシミュレーションをするためのツールが用意されている※1

※1:評価環境はGazebo、検証用のロボットは自前で用意する必要有

環境地図※2

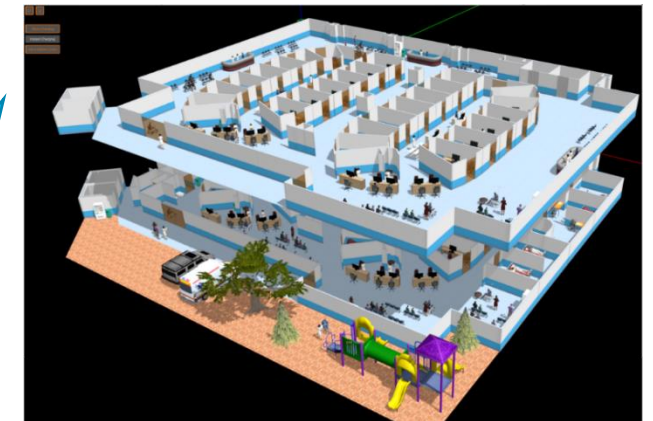


※2:寸法と位置関係が分かれば何でも良い
ex:CAD図、SLAMで作成した地図、etc

専用のエディタ

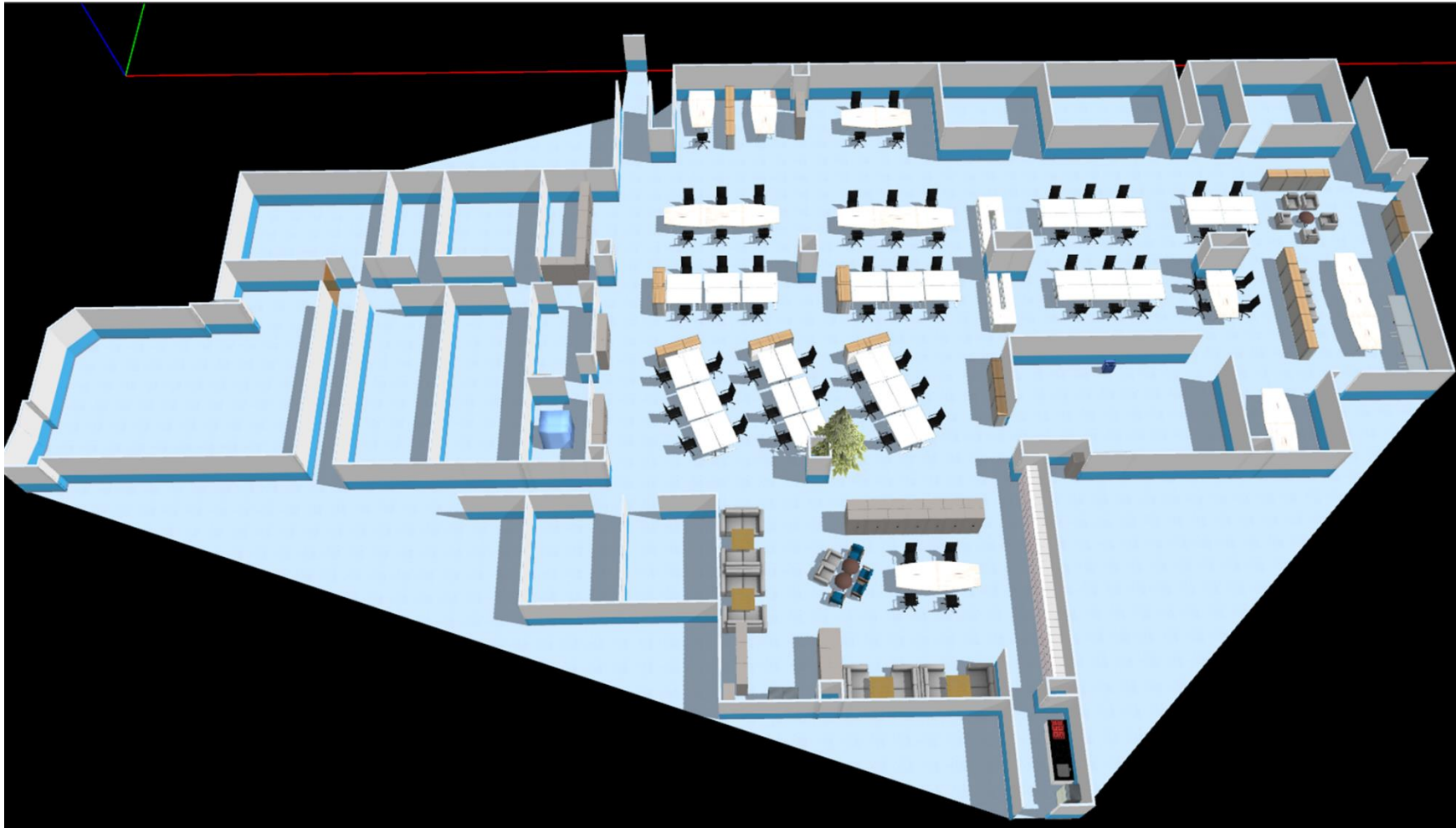


シミュレーション環境



(参考) RMFを使った3次元シミュレーション環境の作成

(例) シンガポールの自社オフィスを模擬したシミュレーション環境
我々のオフィスでSLAMで作成した地図をベースに評価環境を作成



3時間程度で作成が可能です。もちろん、複数階の建物 エレベーター 自動ドア も設置可能です。

(参考) RMFを使った3次元シミュレーション環境方法

<https://www.jmfri.gr.jp/document/library/4319.html>



ホームページ > オンラインライブラリ > オンラインライブラリ > 「移動ロボットROS2シミュレーション資料・環境構築ソフトウェア」一般公開

オンラインライブラリ

「移動ロボットROS2シミュレーション資料・環境構築ソフトウェア」一般公開

移動ロボット評価指標調査検討委員会で作成した「移動ロボットシミュレーションツール、資料」を一般公開します。また、説明会のビデオについても併せて掲載します。

移動ロボット評価指標調査検討委員会は、ロボットイノベーションWGに属する委員会で、移動ロボットの社会実装に向けた検討を実施しています。2021年度の活動においてはgazeboを使用した移動ロボットのシミュレーションに関するツール、資料、ビデオを作成しました。

移動ロボット評価指標調査検討委員会は、今回公開する走行環境に加え、2022年度はロボット側の設定環境についても整備し、シミュレーションツールの拡充を図る計画です。これらのシミュレータを皆様の開発・商談などで活用して頂けたら幸いです。

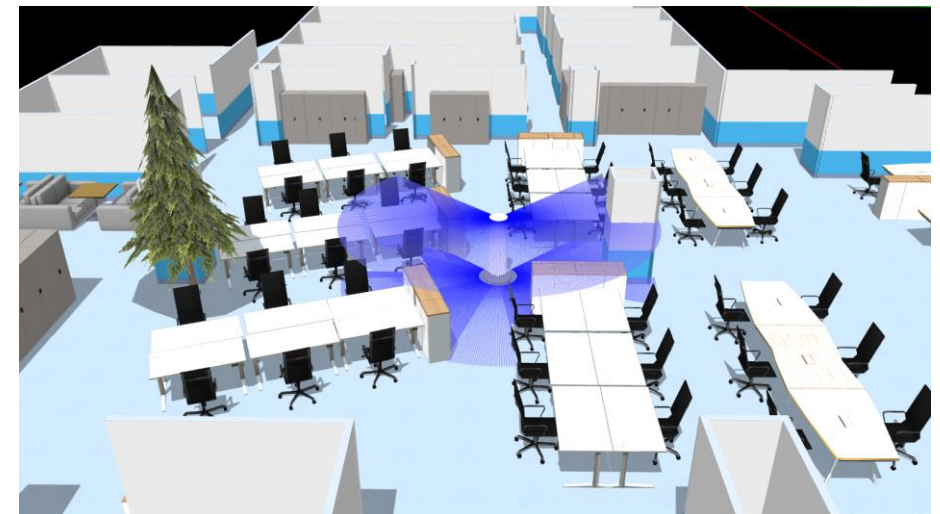
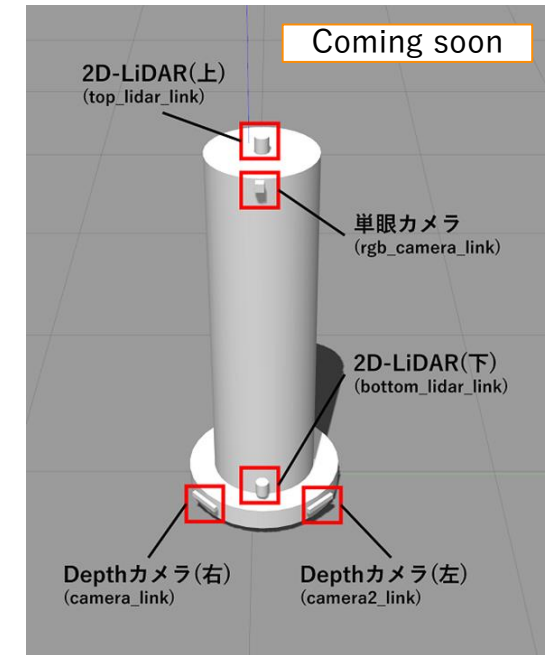
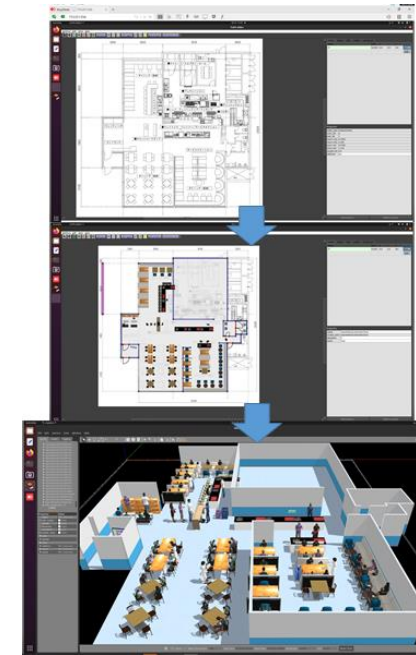
●移動ロボットシミュレーションツール

【ダウンロードファイル】

- ① : gazebo.zip
- ② : rmf_edit_ws.tar.gz
- ③ : turtlebot3_RRI.launch.py.zip ←py形式にてアップできないことからzip形式でアップしています。
- ④ : 【資料1】評価環境作成手順及び起動方法.pdf
- ⑤ : 【資料2】移動ロボットシミュレーション.pdf
- ⑥ : 説明会用サンプルデータ.zip

●説明会 ビデオ

ユーチューブに掲載しています。⇒ 【リンク <https://youtu.be/jrTvjV-WY28>】



RMFの特徴である3つの対応レベル

RMFでは様々なロボットを想定し、ロボットの対応レベルを以下の3つの分類し、それぞれのレベルに応じて制御方法を変更している

注意！！
RMFでインフラ連携
させたい場合は
Full Control 一択！！

対応レベル	Fleet Manager API features	Potential benefits
高	“Full control” : robot waypoint control	Reduce stoppages, increase interaction efficiency. Deal gracefully with the unexpected.
中	"traffic light": robot pause, resume	Spatio-temporal separation of robots can often be achieved.
低*	"read-only": robot locations, destinations	Unified dashboard for operators. Data can allow other robots to avoid conflicts with this fleet.
無対応	no fleet API 😞	None. Fleets cannot coordinate with each other at all to avoid or resolve conflicts.

* ロボット以外の障害物になりうるものも含む

“Full control”: RMF側でロボットの経路生成を行う。ロボットは、RMFから経路を与えられた際に、**与えられた経路に沿って**目的地まで自律移動を行う。

“Traffic Light”: ロボット自身が経路生成を行い、目的地まで自律移動を行う。しかし、ロボットはRMFから“一時停止”と“移動再開”の指示だけは受ける。
また、常時RMFに**走行予定の経路をRMF**に通知する義務がある。

“Read only”: ロボット自身が経路生成を行い、目的地まで自律移動を行う。
RMFの指示には何も従わない。

RMFにおける自律移動しない移動障害物の取り扱い方

- RMFでは病院内でベットや運搬用のコンテナがロボットの進路を塞ぐ可能性も考慮し、これらの機器にBeaconを取り付けて位置を管理する機能がある。
- ロボットの経路生成時にはこれらの障害物も考慮されて経路が生成される。



RoMi-H integrates robotic systems to building infrastructure such as lifts and doors

AMRのすれ違い制御の方法

RMFでは各ロボットの経路情報をもとに、ロボット同士の衝突を事前に検出し、衝突を回避する経路を再生して各ロボットをコントロールする。

注意！！：良く勘違いされるが、同じ廊内ですれ違いを制御するわけでは無い、同じ廊下にAMRが2台来ないようにする



(例)

各ロボットA、Bがそれぞれ地点AからBへ、地点BからAに同時に移動する時、各ロボットが一番最短のコースで経路まで行く経路が生成される。しかし、左の様に経路がバッシングしてしまう。

そこで、RMFはいずれか一方のロボットの経路（ここではロボットBの経路）をバッシング（通過時間も考慮して）しないように経路を再生成しロボットBに指示を出す。

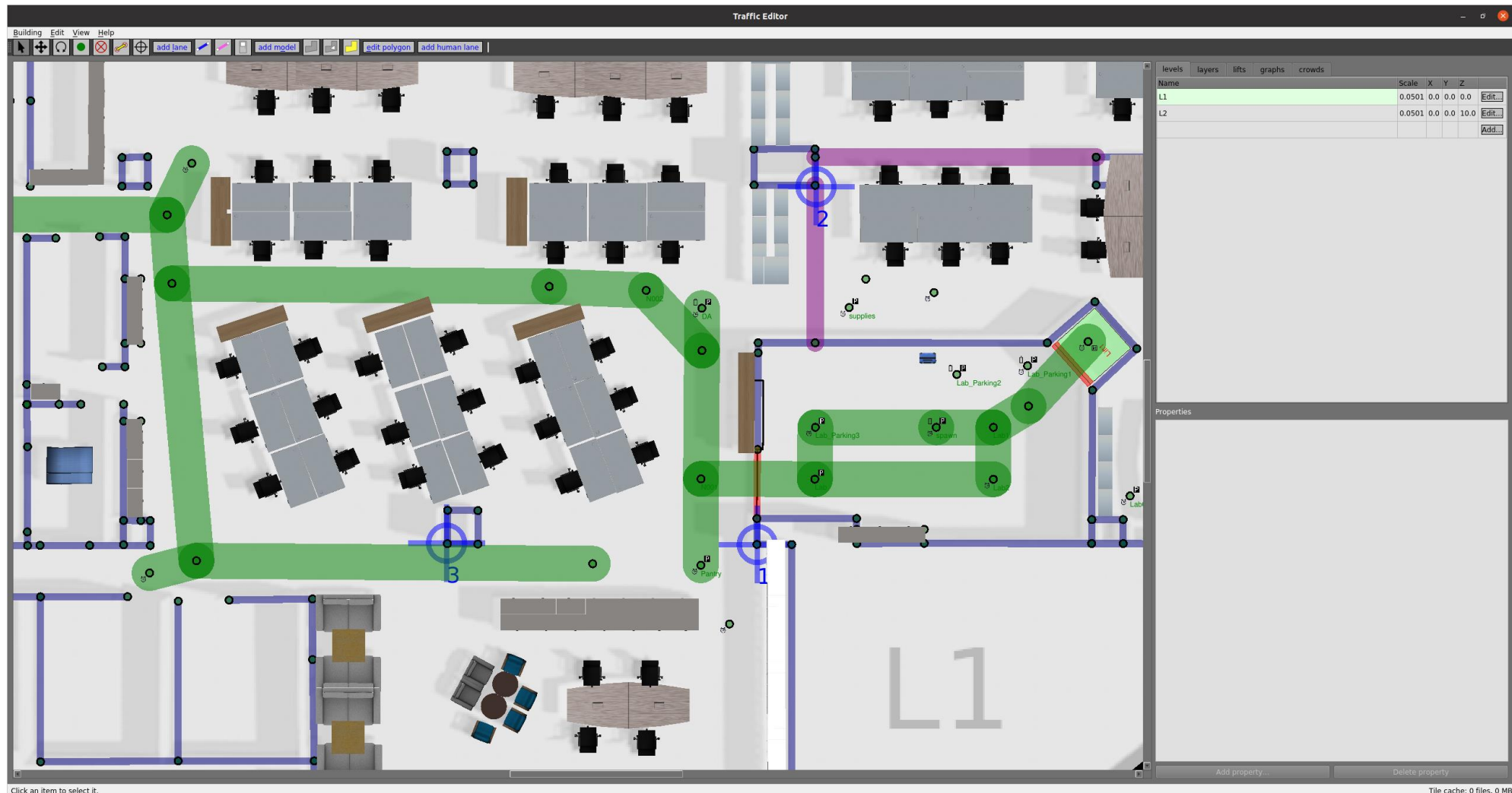
回避する経路が生成できない場合は、衝突しない位置まで一度回避する。

RMFの経路設定方法

RMF内の経路はTraffic editorで簡単に設定が可能
ロボットが通過するWayPoint・ターゲットポイント・走行レーン・自動ドア・エレベーターを設定する

評価環境の作成

<https://osrf.github.io/ros2multirobotbook/traffic-editor.html>



RMF制御時の各レーンでの速度制御

RMFで各レーンごとに最高速度を設定できる事が可能
ただし、データのみが配信される状態なので、実際に利用する場合はロボット側の対応が必須

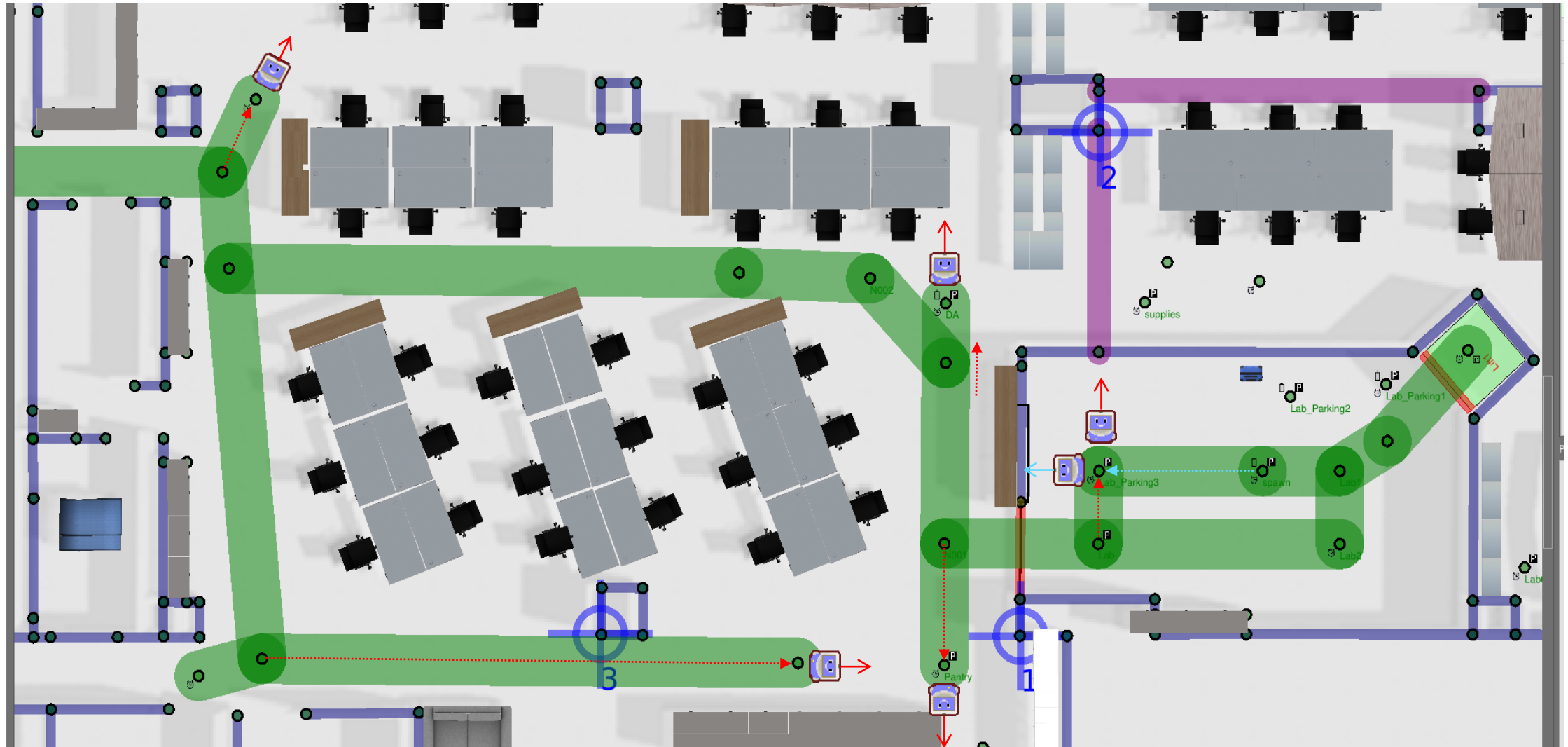
The screenshot displays the Traffic Editor interface. The main workspace shows a 2D environment with various lanes (green, purple, blue) and robot positions. A red box highlights a specific lane, and a callout box shows its properties. A smaller inset also shows the properties for the same lane.

Properties	
edge_type	lane
start_idx	198
end_idx	203
start x (m)	49.2800
start y (m)	18.8165
end x (m)	50.9404
end y (m)	17.0241
length (m)	2.4434
bidirectional	true
demo_mock_floor_name	
demo_mock_lift_name	
graph_idx	0
orientation	
speed_limit	0.3

Setting of Lane

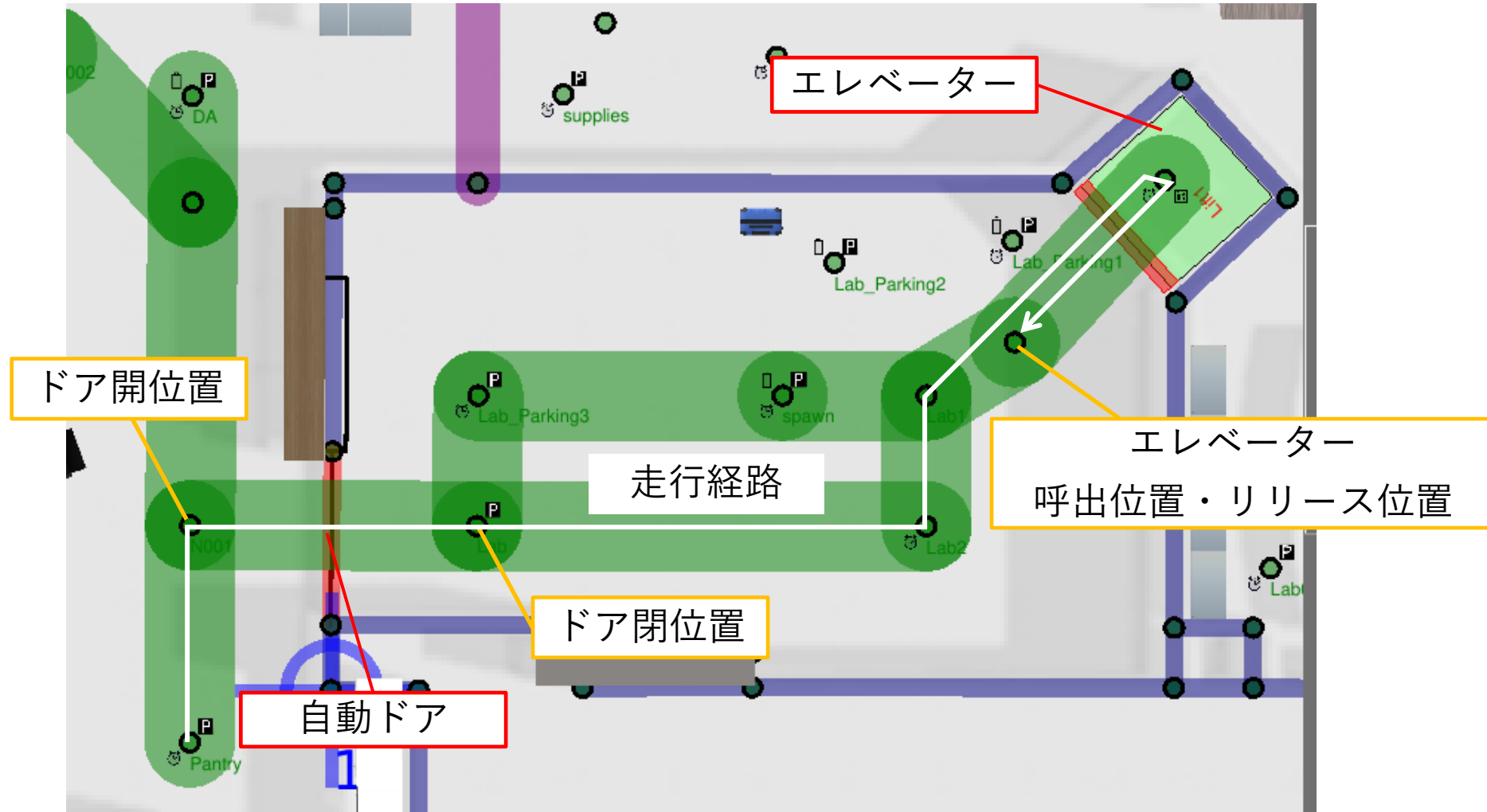
RMF制御時のAMRの各ポイントでの角度制御

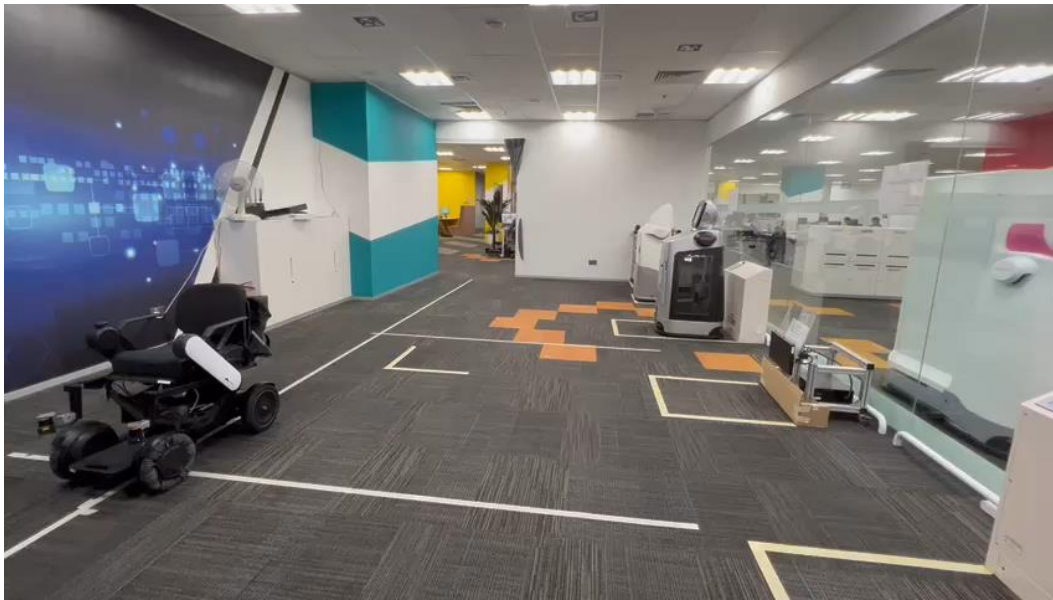
AMRの各ポイントでの目標角度は過去の経路の角度に依存
各ポイントで任意の角度を設定不能



RMF制御時の自動ドア、エレベーター連携

RMF制御時はRMFが各AMRのインフラ連携のタイミングを司る
各AMRがインフラ直前直後のWaypointに到着時にRMFがインフラを制御します





Time is: 0:47:25

Task Submissions

Submit a List of Tasks

SELECT FILE

```
eg [
  {"task_type": "Loop", "start_time": 0, "priority": 0, "description": {"num_loops": 5, "start_name": "cove", "finish_name": "lounge"}},
  {"task_type": "Delivery", "start_time": 0, "priority": 0, "description": {"option": "coke"}},
  {"task_type": "Loop", "start_time": 0, "priority": 0, "description": {"num_loops": 5, "start_name": "cubicle_2", "finish_name": "supplies"}}
]
```

SUBMIT TASK LIST

SUBMIT REQUEST

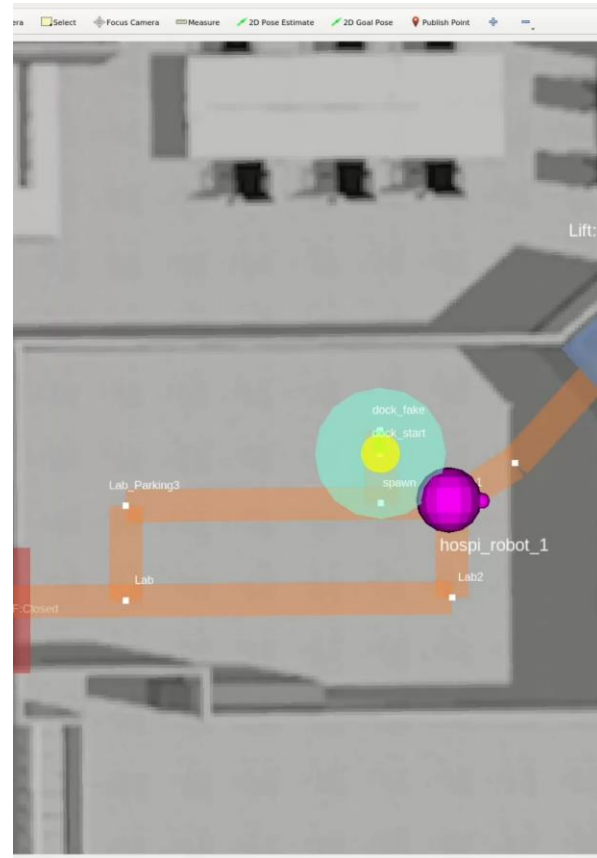
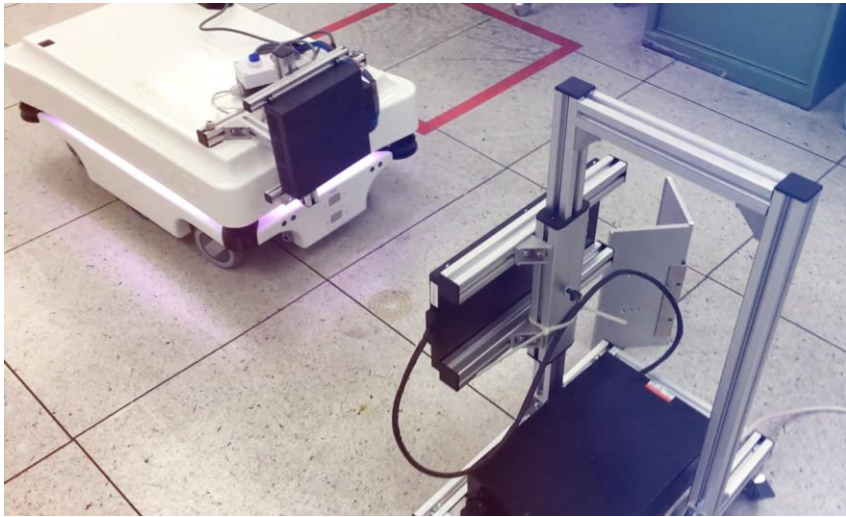
Robots & Tasks Summaries

Tasks

REFRESH

RMFの充電動作への対応

RMFの目的地の種類でドッキングポイントとゆう設定があり、目的地到着後特定の動作を開始できるようにタスクを割り当てることが可能



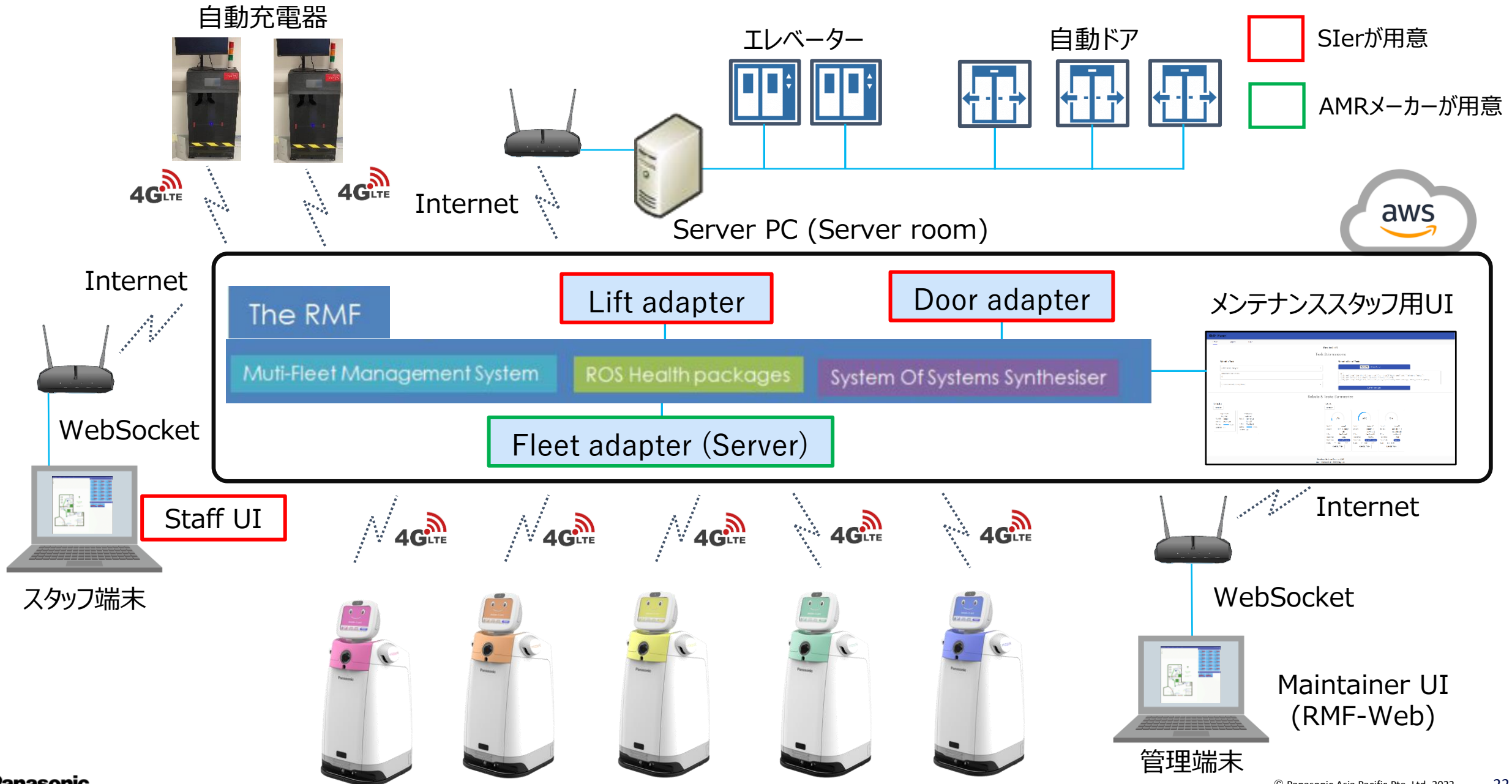
目次

RMFの開発された背景

RMFの機能

RMFを使うにあたって

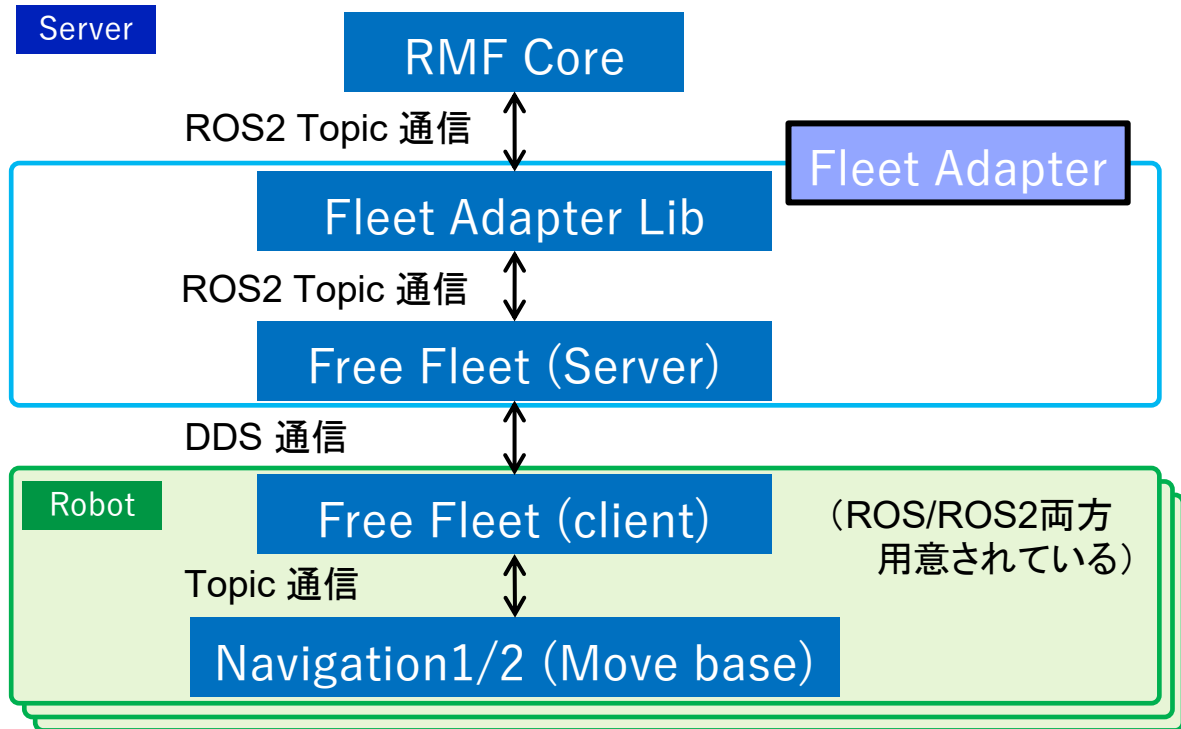
RMFを用いたAMRの制御システム構成 (例)



Fleet adapterの作成方法

Fleet adapterは以下の2つのサンプルを使う事で簡単に作成が可能

Free Fleet を使う場合



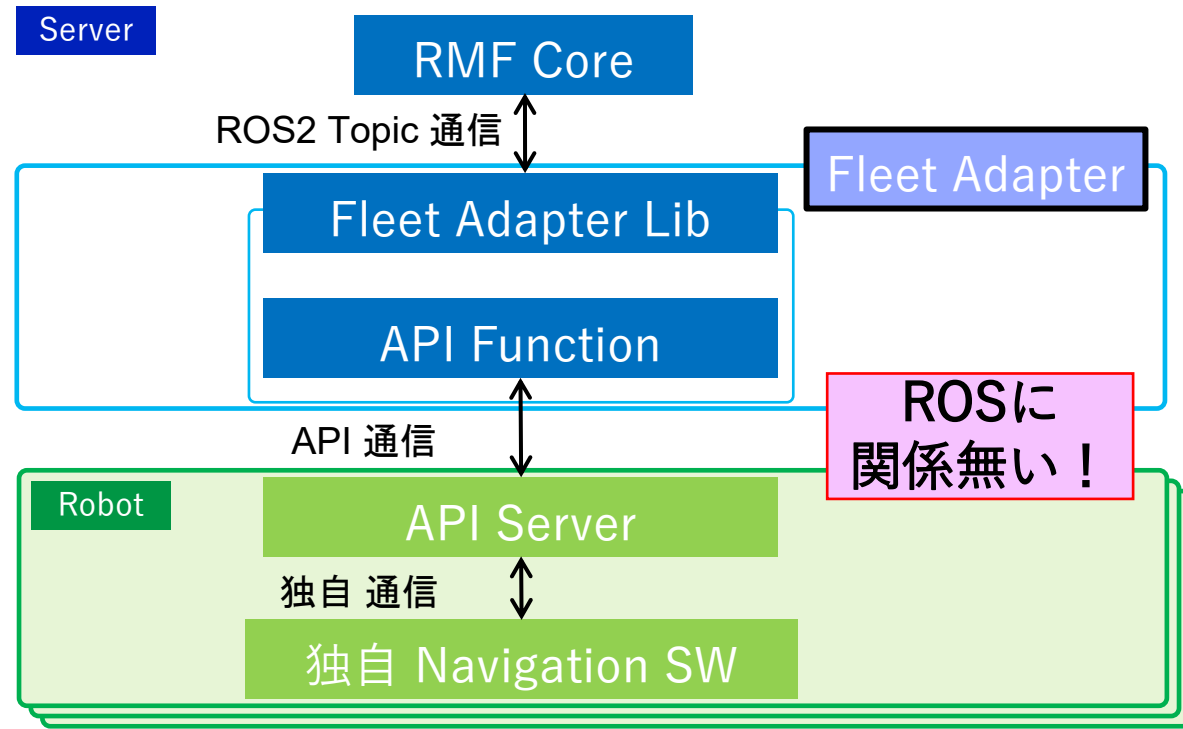
メリット

ROSで動作するロボットであれば、すぐに動作可能

デメリット

DDS通信の縛りがある
ROSで動作しないロボットはカスタマイズが必要

Fleet Adapterのひな形を使う場合



メリット

独自の通信方式が採用可能
ROSを採用していないロボットでも動作可能

デメリット

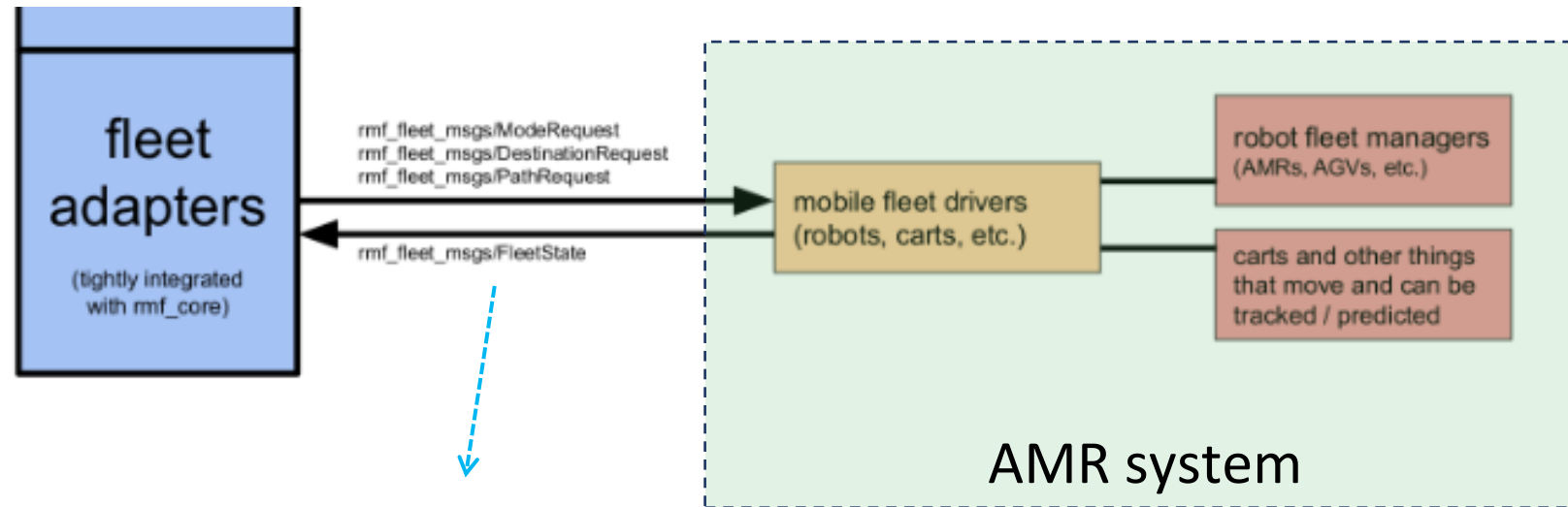
APIを自分で用意する必要がある

Free Fleet: https://github.com/open-rmf/rmf_ros2/tree/main/rmf_fleet_adapter/src/full_control

C++: https://github.com/open-rmf/rmf_ros2/tree/main/rmf_fleet_adapter
Python: https://github.com/open-rmf/rmf_ros2/tree/main/rmf_fleet_adapter_python

AMRのRMFのインターフェース

- RMFでは基本的に3つのリクエストと、1つの状態通知を要しており、各対応レベル毎に必要なAPIを用いる



From RMF to AMR Fleet Manager

- ModeRequest --- pause/resume motion
- DestinationRequest --- set destination
- PathRequest --- set complete paths

From AMR Fleet Manager to RMF

- FleetState --- state reporting

”Full control”: PathRequest と FleetState

”Traffic Light”: ModeRequest と FleetState

”Read only”: FleetState

ステータス情報 (FleetState) のメッセージタイプ

- RMFの中で各ロボットは、**目的地までの経路情報**を求められる

理由：

- ①：自動ドアやエレベーターに搭乗する意味は、経路情報から汲み取るため
- ②：各ロボットの経路を事前に把握し、それぞれのロボットに対して最適な経路を再設計するため

RobotState message:

https://github.com/open-rmf/rmf_internal_msgs/blob/main/rmf_fleet_msgs/msg/RobotState.msg

Datatype	Data Name	Data to be Sent
<i>string</i>	name	<i>HOSPI's ID</i>
<i>String</i>	model	HOSPI
<i>string</i>	task_id	<i>Task ID</i>
<i>float32</i>	battery_percent	<i>Battery Level</i>
<i>Location</i>	location	<i>HOSPI's current Location</i>
Location[]	path	List of waypoints for HOSPI
<i>RobotMode</i>	mode	<i>HOSPI's status</i>

Location message:

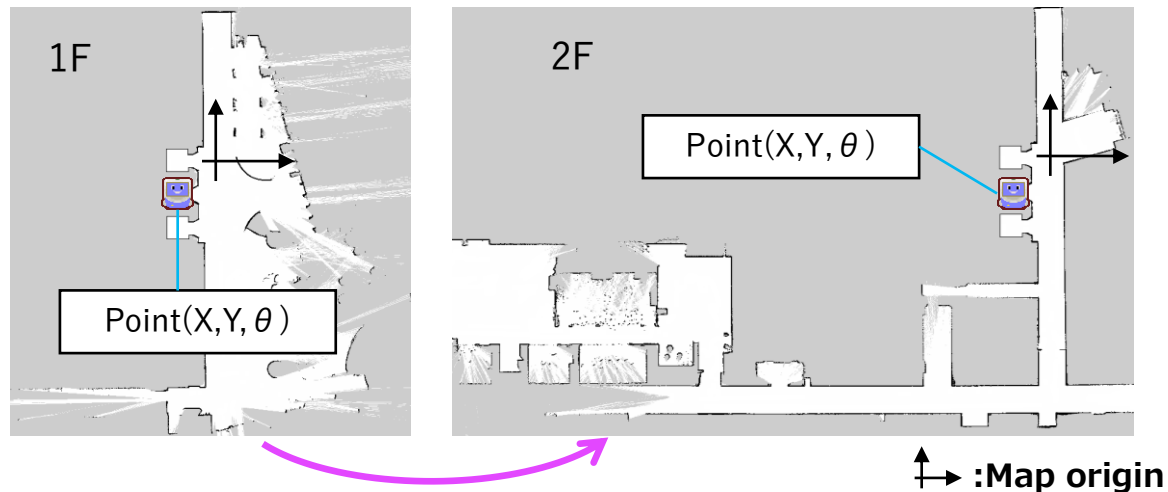
<i>builtin_interfaces/Time</i>	t	<i>unknown</i>
<i>Float32</i>	X	<i>X coordinate</i>
<i>float32</i>	Y	<i>Y coordinate</i>
<i>float32</i>	Yaw	<i>Yaw angle (radians)</i>
<i>string</i>	level_name	<i>Level number</i>

RMFを使う前のAMR側の準備

RMFを使ってフロア間移動をする場合は、ROS Navigationにフロア切替機能を追加が必要

・実現するための代表的な方法

Navigation用の地図を物理的に切り替える



Change Floor = Change map = Republish new map topic

■ Send coordinates to RMF

$(X,Y,\theta) \Rightarrow (X,Y,\theta)$ in 1F

$(X,Y,\theta) \Rightarrow (X,Y,\theta)$ in 2F

メリット

地図を統合する必要なし

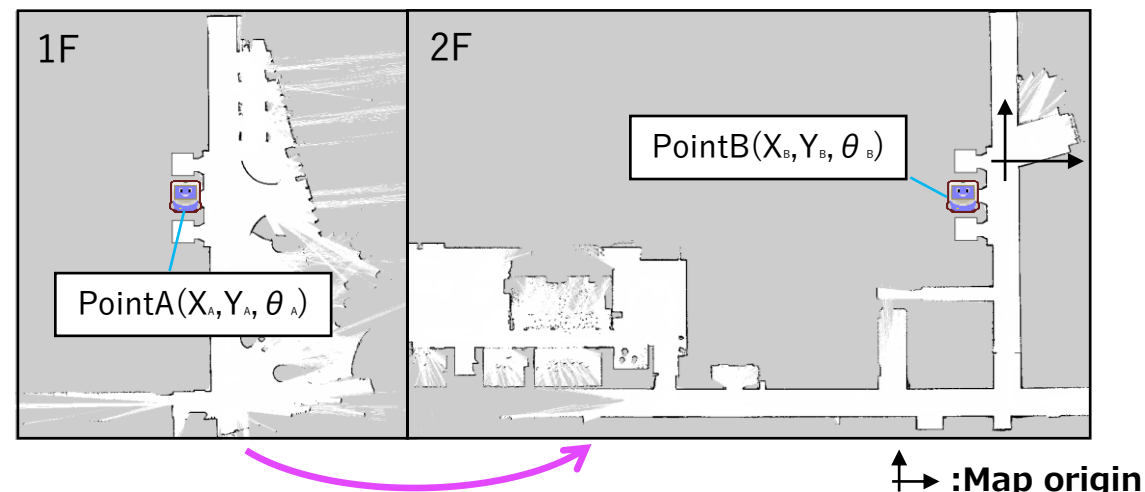
地図内で座標変換が1回で済む

デメリット

Mapsever の改造が必要

各階の座標の統一が必要

同一地図内でロボットの位置を移動させる



Change Floor = Change position = Resend initial pose again

■ Send coordinates to RMF

$(X_A, Y_A, \theta_A) \Rightarrow (X,Y,\theta)$ in 1F

$(X_B, Y_B, \theta_B) \Rightarrow (X,Y,\theta)$ in 2F

メリット

Mapsever の改造が必要なし

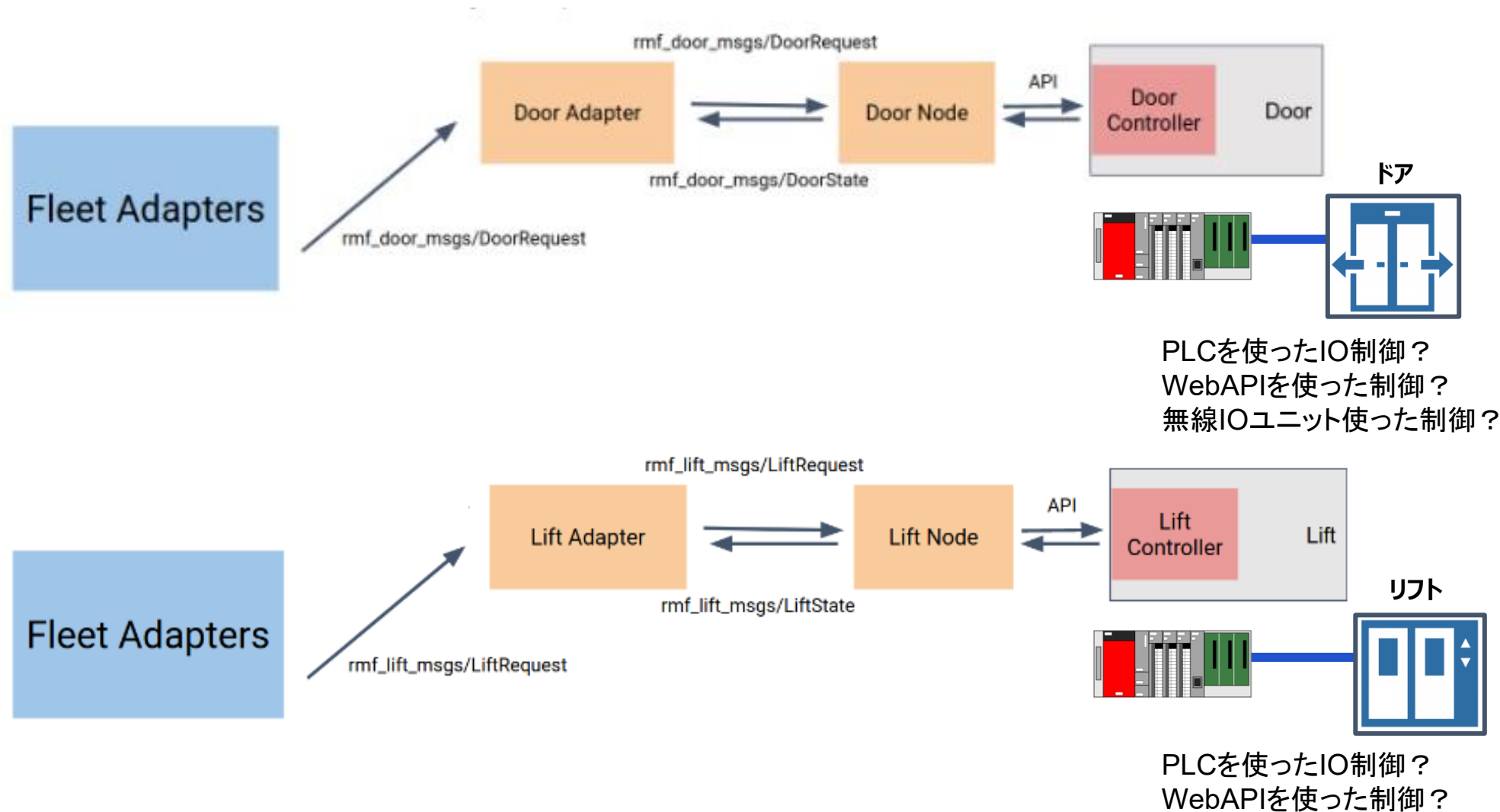
デメリット

各階毎に座標変換が異なる

地図の統合が必要

RMFを使ったインフラ連携

各インフラに合わせてSIerによってノードのカスタマイズが必要





The image features the Panasonic logo in white, bold, sans-serif font, centered horizontally. The background is a dark blue gradient with several geometric elements: a large, semi-transparent light blue circle on the left side, a smaller solid light blue square on the right side, and a grid of thin white lines. The overall aesthetic is clean and modern.

Panasonic