ROSCon JP 2024

群管理における分散協調方式と集中管理方式にまつわる実証実験の取り組み紹介

2024年9月25日

ソフトバンク株式会社 ROS-SI推進課

SoftBank

ソフトバンク ROS-SI推進課
活動内容紹介

ROS-SI推進課の紹介

Cuboidをはじめとした内製ROSロボットを開発して運用
国プロ実証実験やグループ内ソリューション・案件のPOCを実施

社内法人案件のPOC実施[1]

高精度測位サービス「ichimill」[2]の活用

内製ロボット
Cuboid・SOARの開発と運用

実証事業の受託[3]

[1] https://www.softbank.jp/sbnews/entry/20230929_01
[2] https://www.softbank.jp/biz/services/analytics/ichimill/
[3] https://www.walkingspacedx.go.jp/post-619/

国プロへの参画

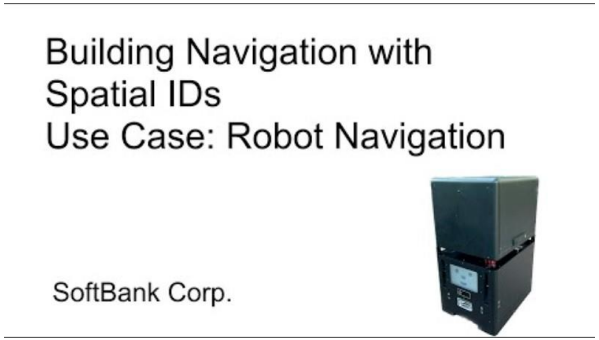
各種実証事業に内製ロボットCuboidを用いて参画

（FY19）屋内配送ロボットと エレベーターとの通信連携

令和元年度予算IoTの安心・安全かつ適正な利用環境の構築
https://www.softbank.jp/sbnews/entry/20200109_01
https://youtu.be/0SNI1tObTPM

（FY21）屋外配送ロボットと 信号機との通信連携

自動走行ロボットを活用した新たな配送サービス実現に向けた技術開発事業
https://www.softbank.jp/corp/news/press/sbkk/2021/20210615_01/
https://youtu.be/CyYP8fSry-I



（FY22）共通インデックス 空間IDを介したデータ連携

デジタル庁「デジタルツイン構築に関する調査研究」事業
https://www.softbank.jp/corp/news/press/sbkk/2023/20230425_01/
https://youtu.be/h7Y9QNBfTUg

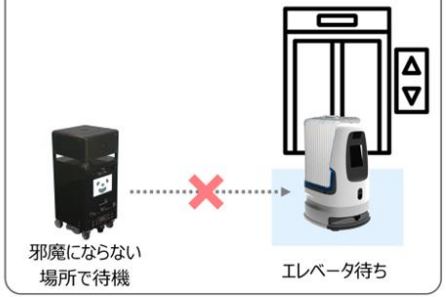
複数のモビリティの協調運行に関する実証調査研究

SoftBank

FY23の事業にて、マルチベンダーロボットでの各種検証実験を羽田イノベーションシティにて実施

経路の調停

エレベータや自動ドア出入口などの
経路の占有情報を共有することで
邪魔にならない場所で待機



ロボット位置の可視化

施設内でマルチベンダーのロボットの
位置情報をリアルタイムに可視化し
運行状況の監視を実施

ロボット用地図や運行ルールの共有

ロボット用地図や進入禁止エリアを
共有することで地図の作成工数を削減

https://www.digital.go.jp/assets/contents/node/basic_page/field_ref_resources/9f4e70e2-2335-4181-8293-258c12549d31/3045743d/20240508_policies_mobility_report_02.pdf

複数のモビリティの協調運行に関する実証調査研究

SoftBank

FY23の事業にて、マルチベンダーロボットでの各種検証実験を羽田イノベーションシティにて実施

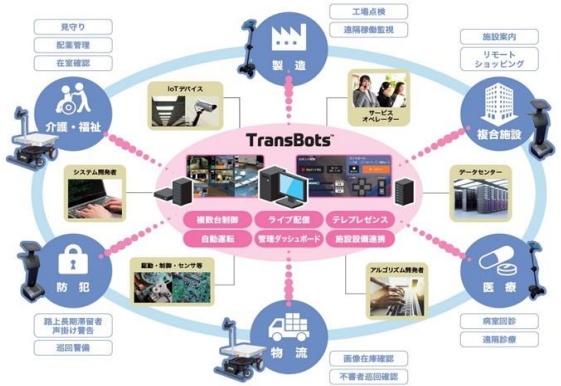
今日の話題

https://www.digital.go.jp/assets/contents/node/basic_page/field_ref_resources/9f4e70e2-2335-4181-8293-258c12549d31/3045743d/20240508_policies_mobility_report_02.pdf

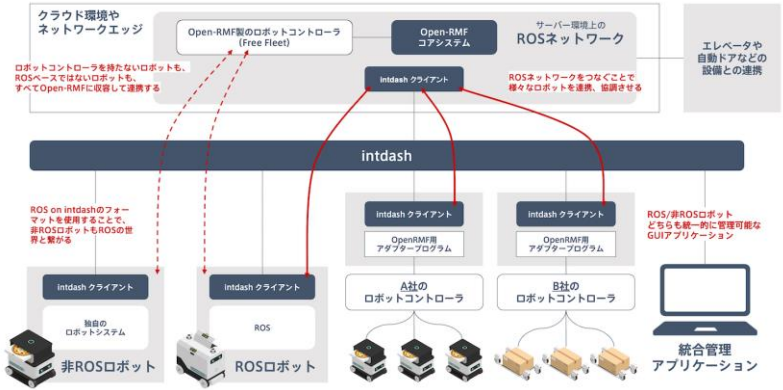
群管理における集中管理と分散協調

集中管理という考え方

- 中央集権的な考え方に基づく群管理の仕組み
- 経路生成・干渉回避、位置・センサー情報の可視化などの機能を保有
- 各ベンダーが独自で保有している群管理（運行管理）システムを指す

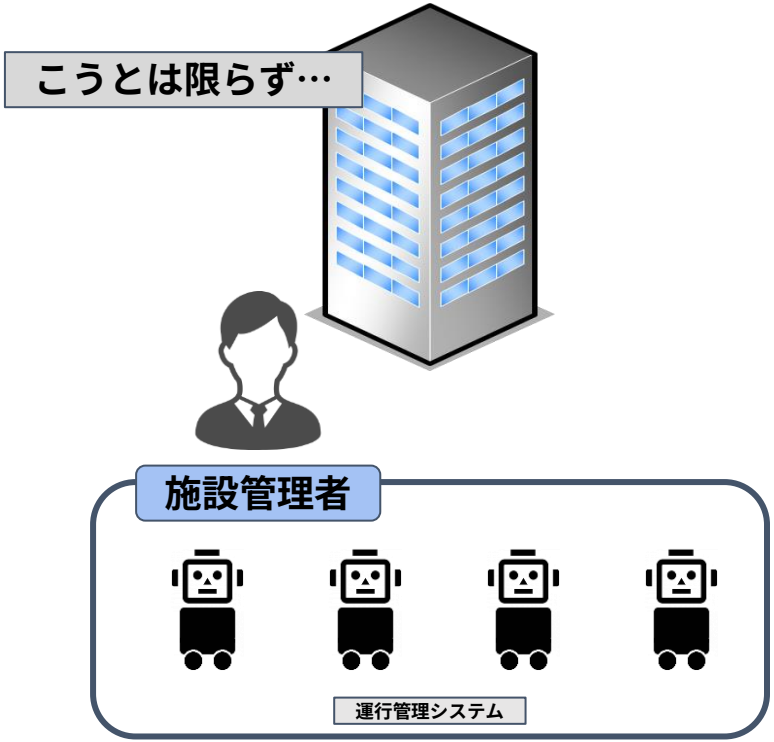


TOPPANホールディングス『TransBots』
https://www.holdings.toppan.com/ja/news/2021/09/newsrelease210909_1.html



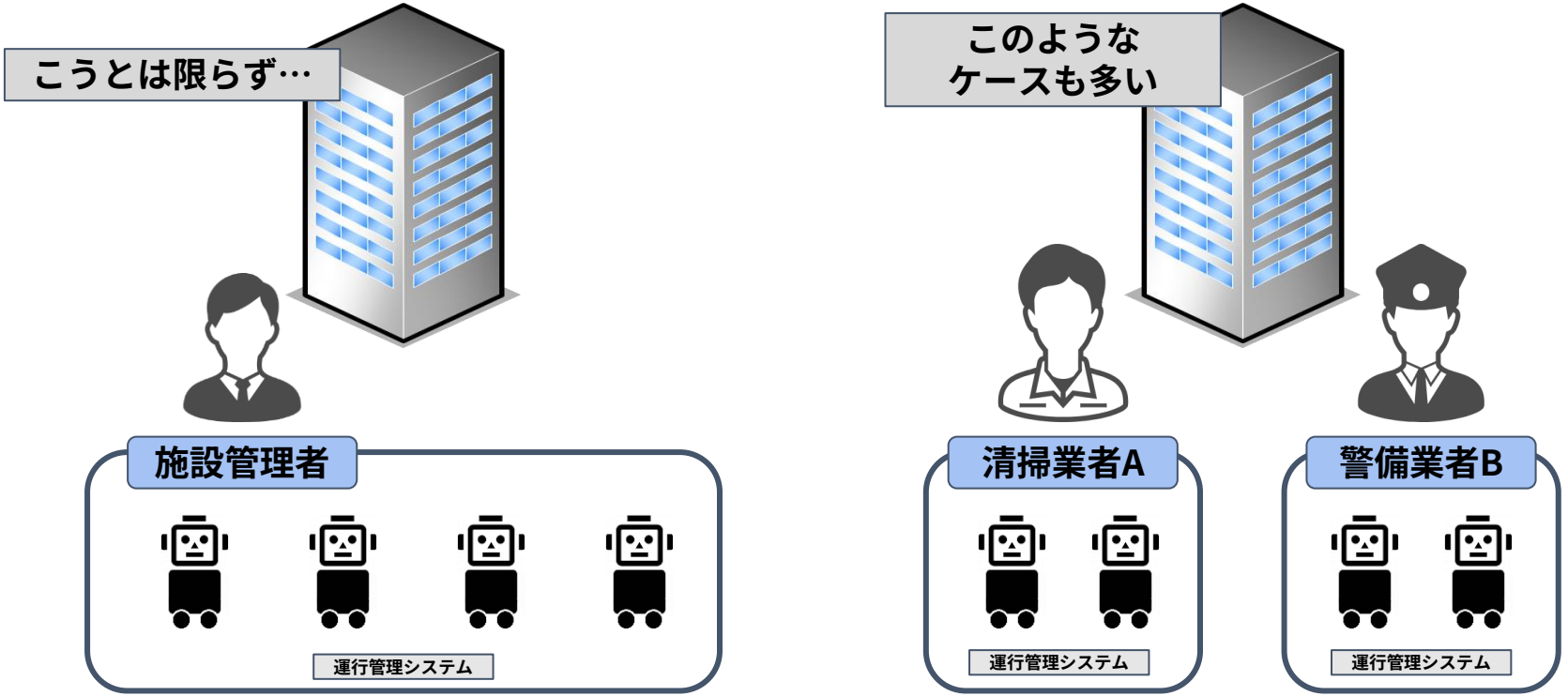
aptpod『intdash CONTROL CENTER』
https://www.aptpod.co.jp/solutions/control-center/robops/

集中管理の問題点



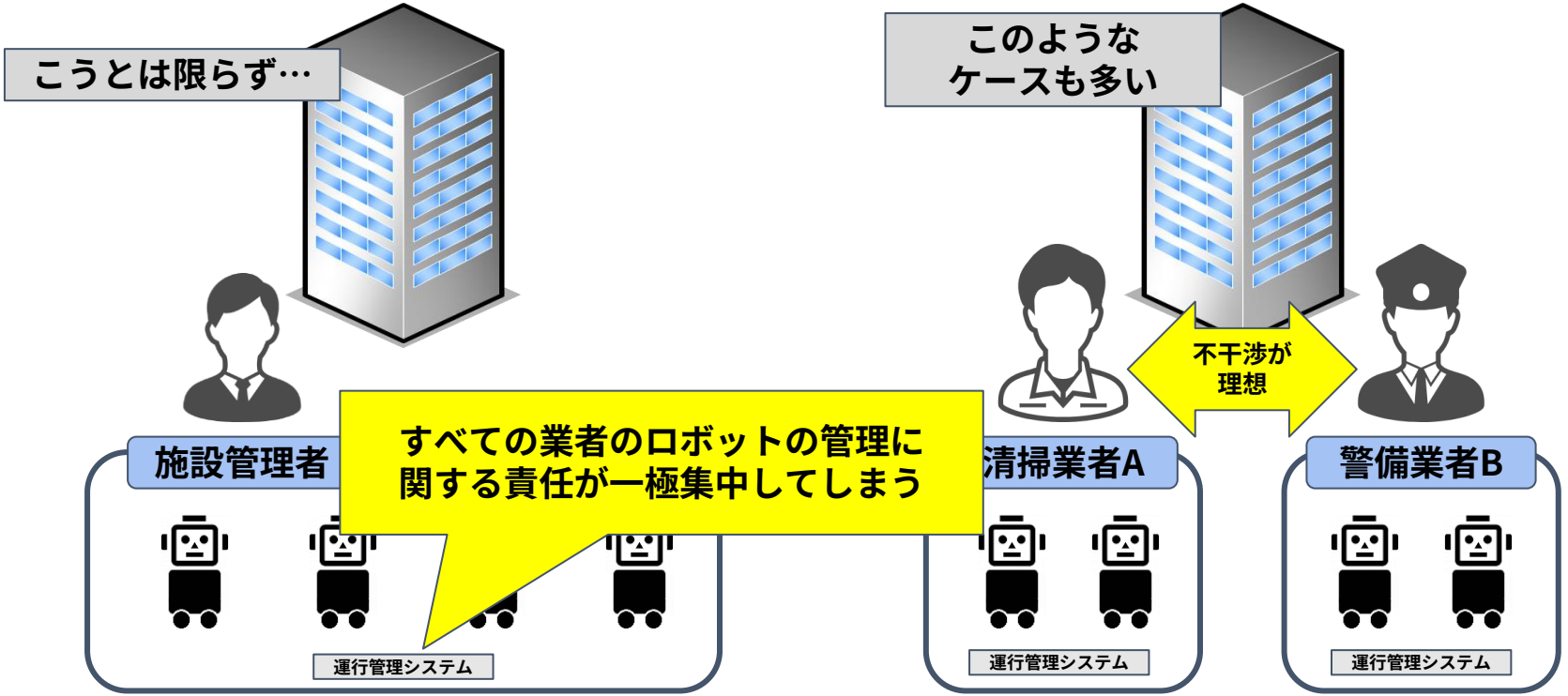
現状の施設は全てのロボットを同じシステム配下に置けるような体制になっていないケースがある

集中管理の問題点



現状の施設は全てのロボットを同じシステム配下に置けるような体制になっていないケースがある

集中管理の問題点



現状の施設は全てのロボットを同じシステム配下に
置けるような体制になっていないケースがある

集中管理の問題点

こうとは限らず…

東京ポートシティ竹芝の例

清掃

https://www.softbank.jp/sbnews/entry/20200910_02?page=03#page-03

警備

https://www.softbank.jp/sbnews/entry/20200910_02?page=03#page-03

配送

https://www.softbank.jp/corp/news/press/sbkk/2021/20210420_01/

このような
ケースも多い

現状の施設は全てのロボットを同じシステム配下に置けるような体制になっていないケースがある

分散協調の定義・考え方

SoftBank

モビリティ運行における (1)モビリティ機体への機体割当、(2)モビリティの運行計画及び制御の二点に関して、集中的な管理でなく個社のベンダー運行システムに機能が分散配置されること

分散協調の実現によって汎用性・拡張性の獲得、責任集中の回避、共助による経済性の向上といったメリットを享受することができると考える

分散協調のメリット	最小限の共助機能を有する共通基盤を、多様なサービスの提供者が一定の責任のもと活用することにより、共通基盤・インフラ提供者／サービスの提供者双方の経済性を確保しながら、安全に対する責任や拡張性といった集中管理における課題への対処が可能となる
汎用性・拡張性	多様なモビリティが連携可能な汎用性の高い共通技術・インフラ基盤を活用することにより、**多様なサービス・モビリティの乗り入れが容易**
責任分散	分散協調のもと、サービスの提供者（モビリティ運行者）が一定の責任を担うことで共通基盤・インフラ提供者への**責任集中を防ぐことが可能**
経済性	分散協調により、共通基盤・インフラへの機能依存度を低減することで、共助を最小化し、共通基盤提供者の**投資を最小限に抑える**ことが可能

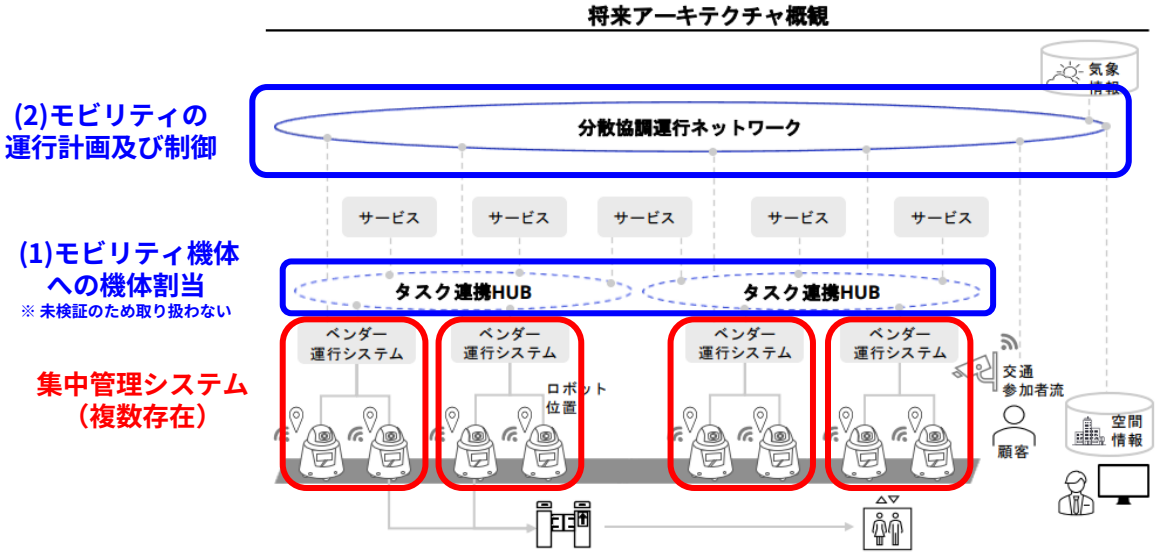
上記のメリットがある一方で、分散協調の仕組みにおいては、a) アーキテクチャの複雑性、b) 多様なI/Fの必要性、c) 集中管理としない認証の実現といった難しさがあると想定され、それら課題の解決に向けた検討が必要と考えられる

複数のモビリティの協調運行に関する実証調査研究 最終報告書（詳細版） p.6, 23
https://www.digital.go.jp/assets/contents/node/basic_page/field_ref_resources/9f4e70e2-2335-4181-8293-258c12549d31/3045743d/20240508_policies_mobility_report_02.pdf

分散協調の仕組みを取り入れた将来アーキテクチャ

赤枠部分は集中管理システム、各ベンダーの責任領域

青枠部分は分散協調の導入、今回の主題となるシステム間の経路調停は(2)に含まれる



*NW：ネットワーク

名称	定義
分散協調運行ネットワーク	分散協調による運行を実現するために必要な情報の収集・加工などを担う
サービス	最終顧客から受注したタスクのモビリティ運行者側とのやり取りを担う
タスク連携HUB	複数のサービスシステムとベンダー運行システム間のタスクの中継を担う
ベンダー運行システム	タスク指示に基づき、ロボット機体を運行・制御する役割を担う

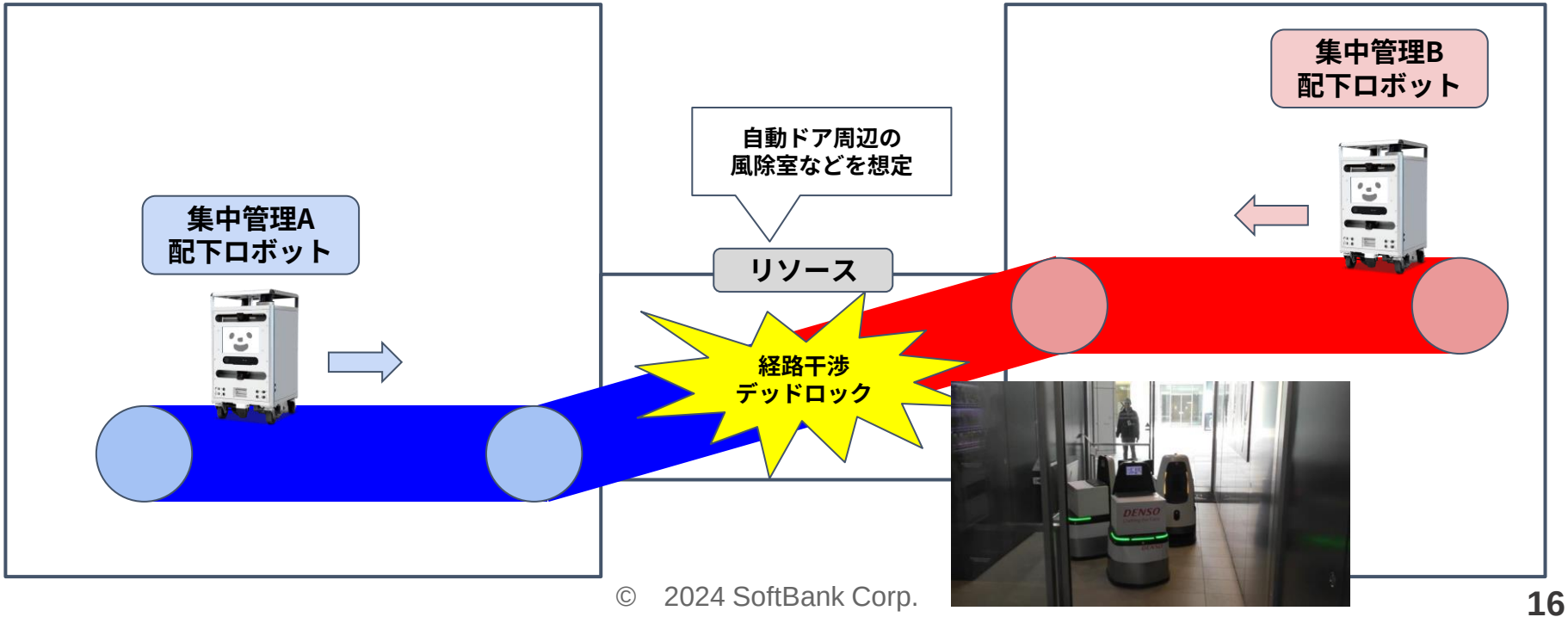
複数のモビリティの協調運行に関する実証調査研究 最終報告書（詳細版） p.22 一部改変
https://www.digital.go.jp/assets/contents/node/basic_page/field_ref_resources/9f4e70e2-2335-4181-8293-258c12549d31/3045743d/20240508_policies_mobility_report_02.pdf

分散協調の思想に基づく経路調停検証

占有情報管理を介した分散協調型経路調停の検証

SoftBank

2つの集中管理システム間で分散協調の考えに則った経路調停を実施
以下のような状況を「第三者による占有情報管理」の導入により未然に防ぐ



検証の前提

SoftBank

各ロボット間で「走行ルール」が共有されている前提に立つ

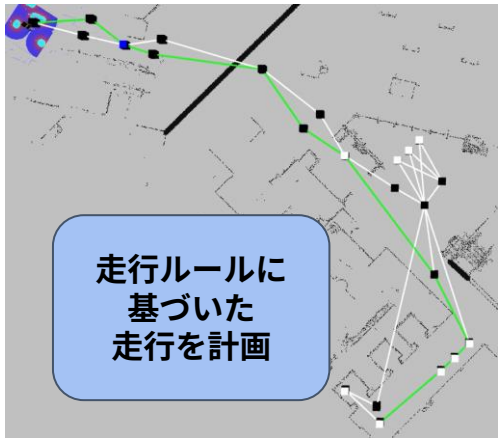
- 共有されたトポロジカル・グラフ地図に基づく経路生成
  - エッジ情報とノード情報から成り、ロボットは移動する際のおおよその経路（グローバルパス）をこれに基づき生成する
- 経路調停に関するルール
  - 調停対象となっているノードについては、サーバー（DWH）に対して占有を申請し、占有に成功してから移動を行う

```
K1N1_K1N2:
  edge_id: K1N1_K1N2
  source_node: K1N1
  destination_node: K1N2
  connection_type: 0
  length: 1.09
```

エッジ情報の例

```
K2N1:
  node_id: K2N1
  map_id: HICity_ZoneK_2F
  floor_name: 2F
  latitude: 35.548521204
  longitude: 139.755288913
  altitude: 10.22
  mediation_target: true
  spatial_id: 26/20/59606706/26456426
  type: 0
```

ノード情報の例

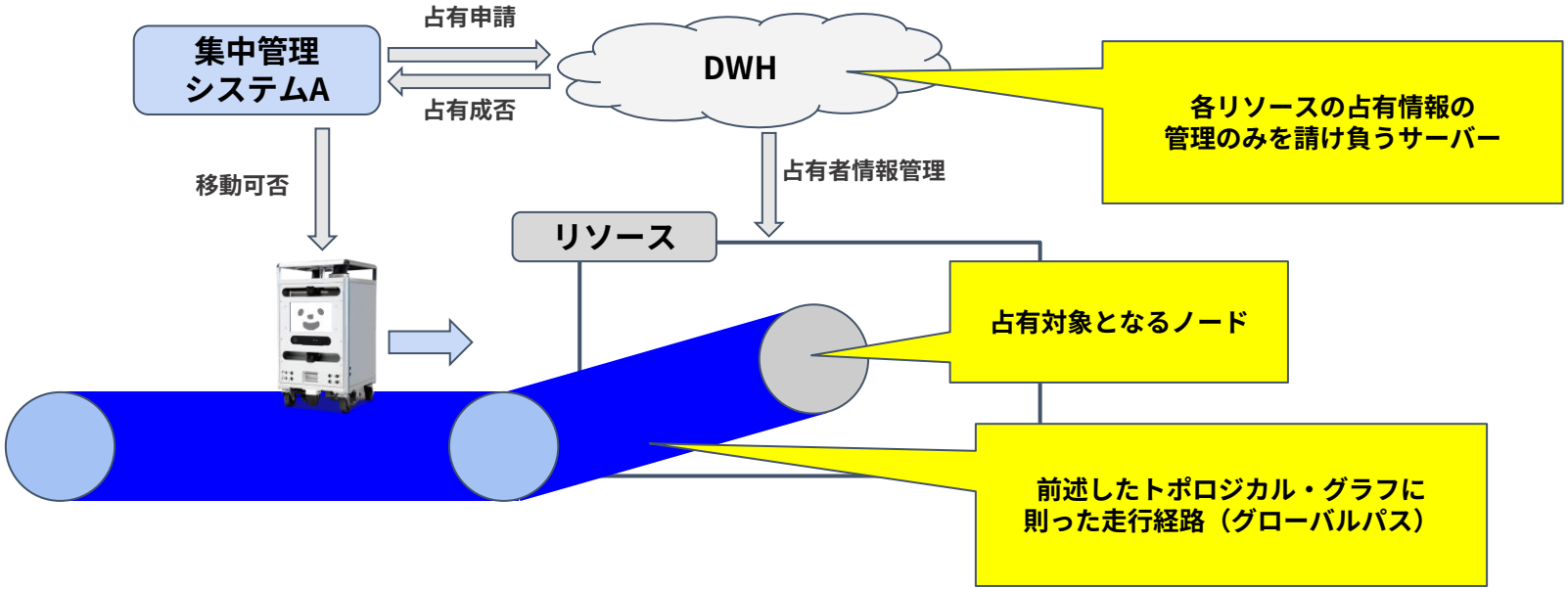


占有情報管理を介した分散協調型経路調停の検証

SoftBank

実施フロー

1. 狭路やエレベーター付近などの特定リソースを事前に定義
2. リソース近辺のノードにロボットが近づいたタイミングでリソース占有申請

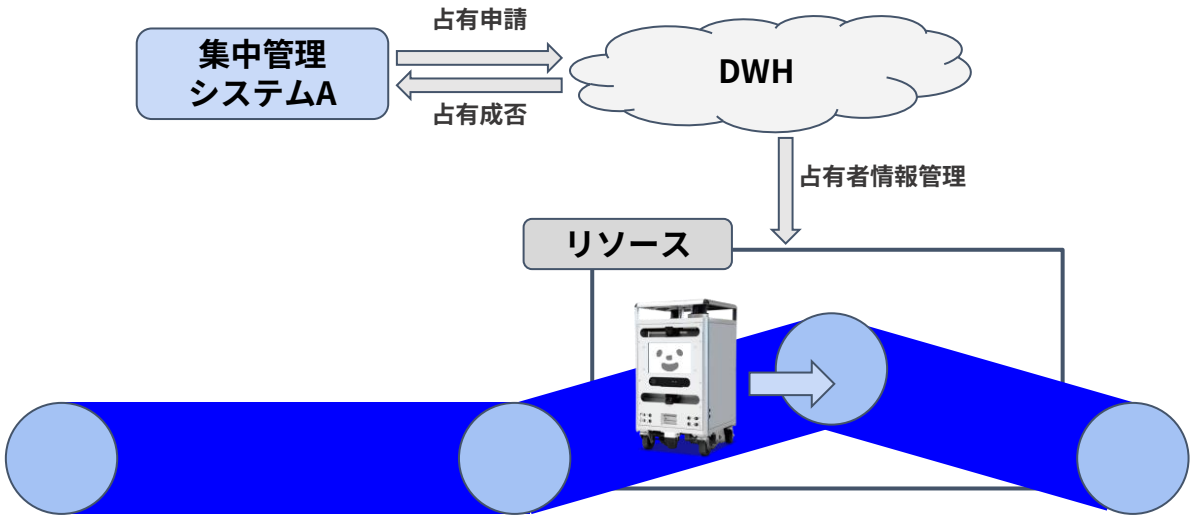


占有情報管理を介した分散協調型経路調停の検証

SoftBank

実施フロー

3-A. リソースの占有取得に成功した場合のみ進行



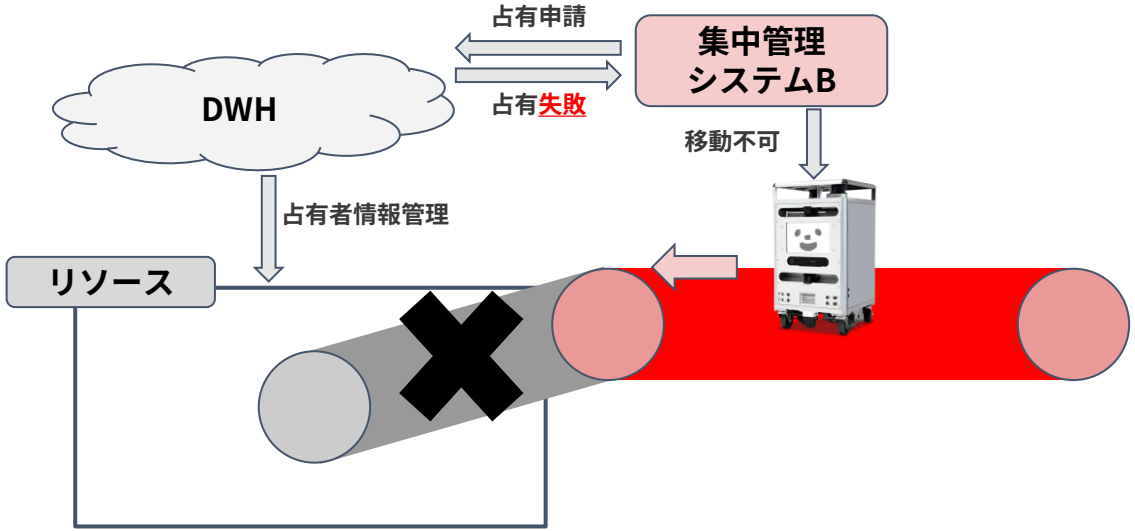
占有情報管理を介した分散協調型経路調停の検証

SoftBank

実施フロー

3-B. リソースの占有取得に失敗した場合は進行を止める

4. 占有が成功するまで待ってから進行を再開



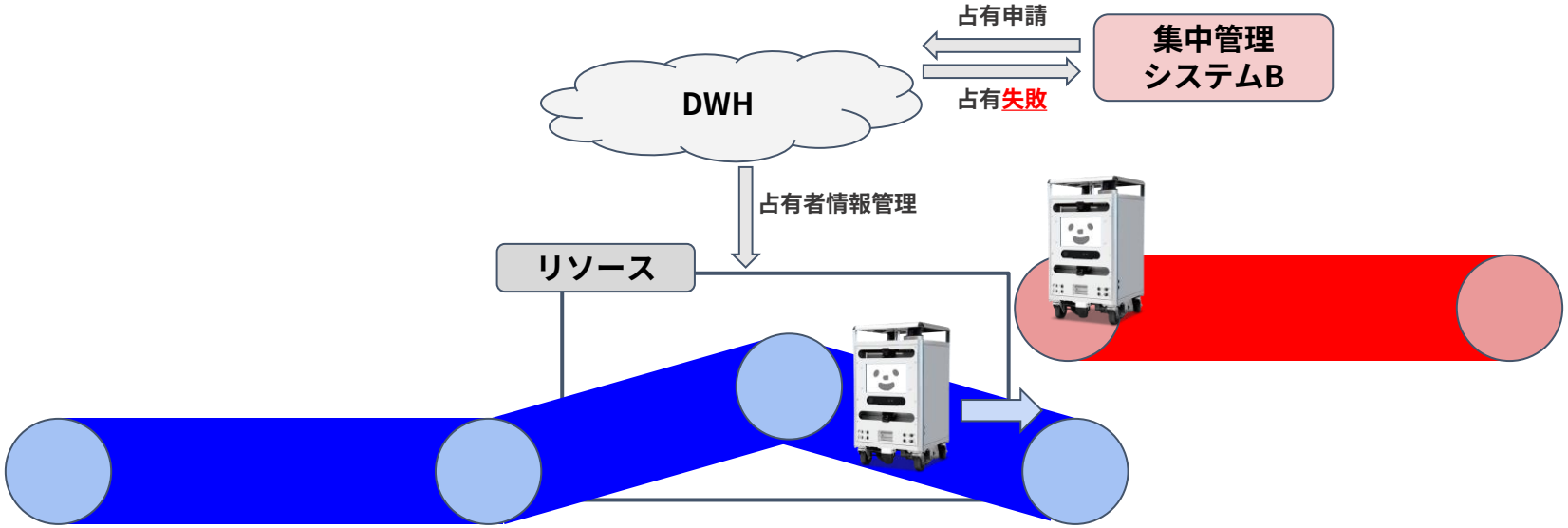
占有情報管理を介した分散協調型経路調停の検証

SoftBank

実施フロー

3-B. リソースの占有取得に失敗した場合は進行を止める

4. 占有が成功するまで待ってから進行を再開

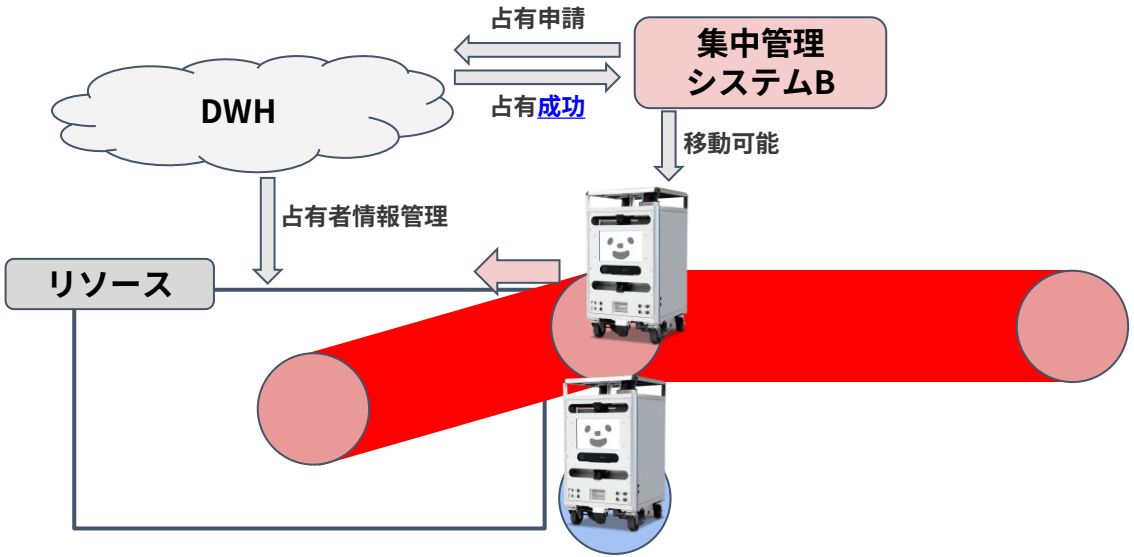


占有情報管理を介した分散協調型経路調停の検証

SoftBank

実施フロー

5. 通行が完了したら占有解除を申請（他のロボットが通行可能となる）



占有情報管理を介した分散協調型経路調停の検証

SoftBank

トポロジカル・グラフの共有と占有申請の仕組みにより
異なるシステム配下のロボット間の経路調停を実現

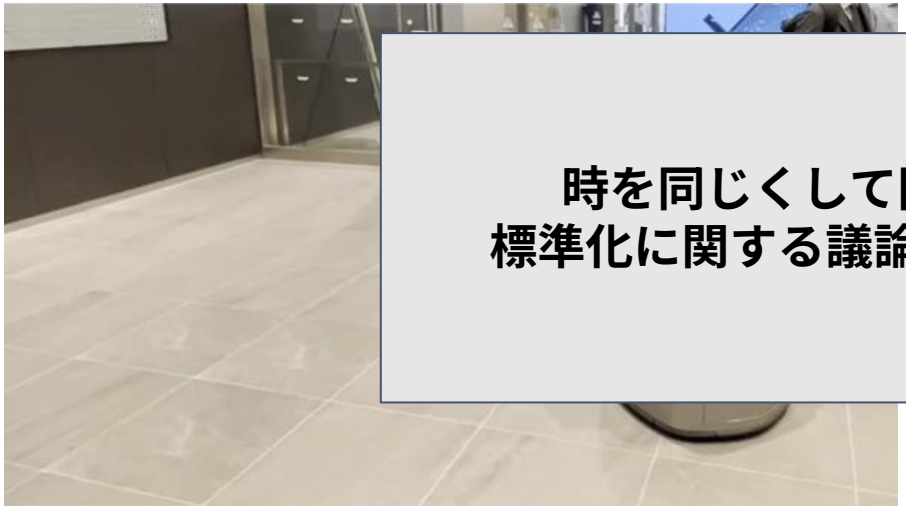
隘路（風除室）付近の経路調停の様子
手前の機体が奥の機体の占有解除を待ってから進行

エレベーター待機時の経路調停の様子
後続のロボットが占有解除を待っている

占有情報管理を介した分散協調型経路調停の検証

SoftBank

トポロジカル・グラフの共有と占有申請の仕組みにより
異なるシステム配下のロボット間の経路調停を実現



時を同じくして同様の内容の
標準化に関する議論が始まっていた

隘路（風除室）付近の経路調停の様子
手前の機体が奥の機体の占有解除を待ってから進行

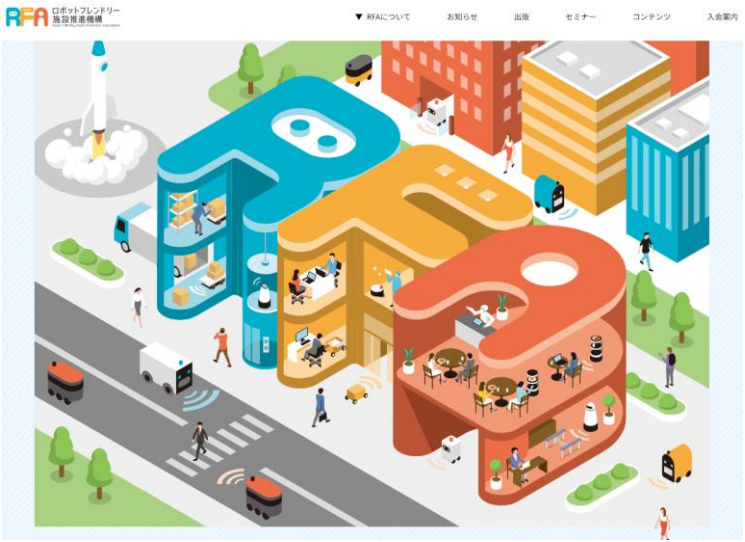
エレベーター待機時の経路調停の様子
後続のロボットが占有解除を待っている

RFA標準

RFA（ロボットフレンドリー施設推進機構）

SoftBank

あらゆるタイプの施設においてロボットの導入を実現するため、
ロボットフレンドリーな環境の構築を目指す



4つのTCから構成

- エレベーター連携TC
- セキュリティTC
- 物理環境特性TC
- ロボット群管理TC

目指すべき世界観

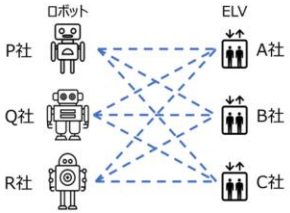
- **メーカーフリー**なロボットと設備の連動を実現すること
- **レトロフィット**を実現できること(既存の施設に導入可能)
- **可能な限りシンプルな実現**を目指すこと

https://robot-friendly.org/

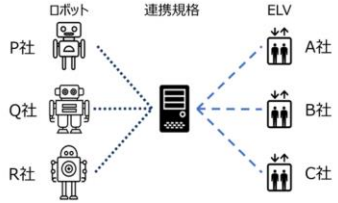
RFA規格の発行

エレベーター連携TC

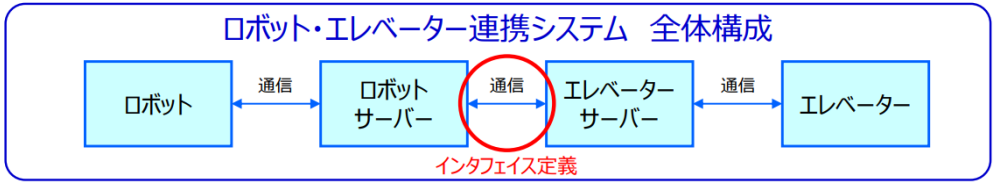
- RFA規格（ロボット・エレベーター連携インタフェイス定義 RFA B 0001 :2022）
- RFAマニュアル（ロボット・エレベーター連携 導入・運用マニュアル RFA MN B 0201 : 2024）



各連携において個別の開発が必要



各社ロボット／エレベーターに対応する共通連携基盤



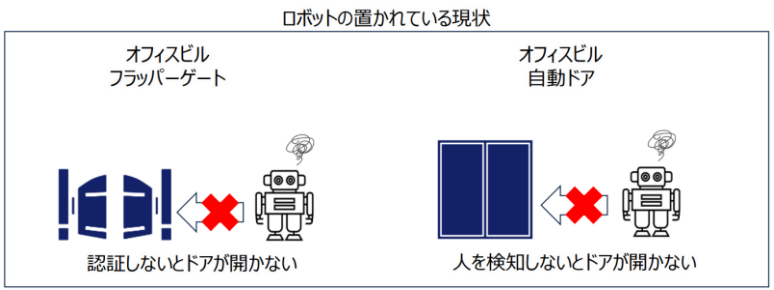
これまでのRFA全体の活動実績についてはこちらの対外発表資料参照

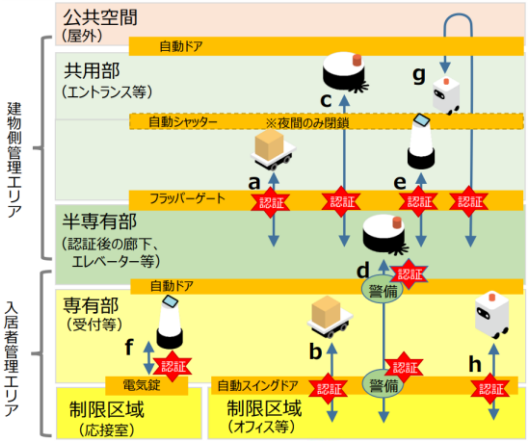
https://robot-friendly.org/wp/wp-content/uploads/2024/07/20240717_RFA_Presentation_2024.pdf

RFA規格の発行

セキュリティTC

- RFA規格（ロボット・セキュリティ連携インタフェイス定義 RFA B 0002 :2023）
- RFAガイドライン（ロボット・セキュリティ連携 ガイドライン RFA GL B 0101 :2023）





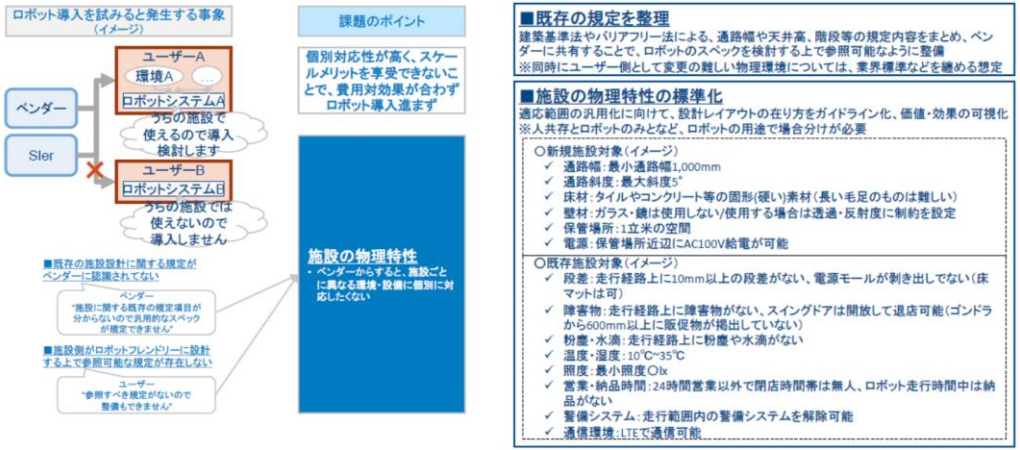
これまでのRFA全体の活動実績についてはこちらの対外発表資料参照

https://robot-friendly.org/wp/wp-content/uploads/2024/07/20240717_RFA_Presentation_2024.pdf

RFA規格の発行

物理環境特性TC

- RFA規格（サービスロボットの移動の円滑化 ― 物理環境の分類と指標 ―建築物およびその敷地内 RFA B 0003 : 2024）
- RFA 規格(サービスロボットの移動の円滑化 ― ロボットが共有して利用する画像標識 RFA B 0005 : 2024)



これまでのRFA全体の活動実績についてはこちらの対外発表資料参照

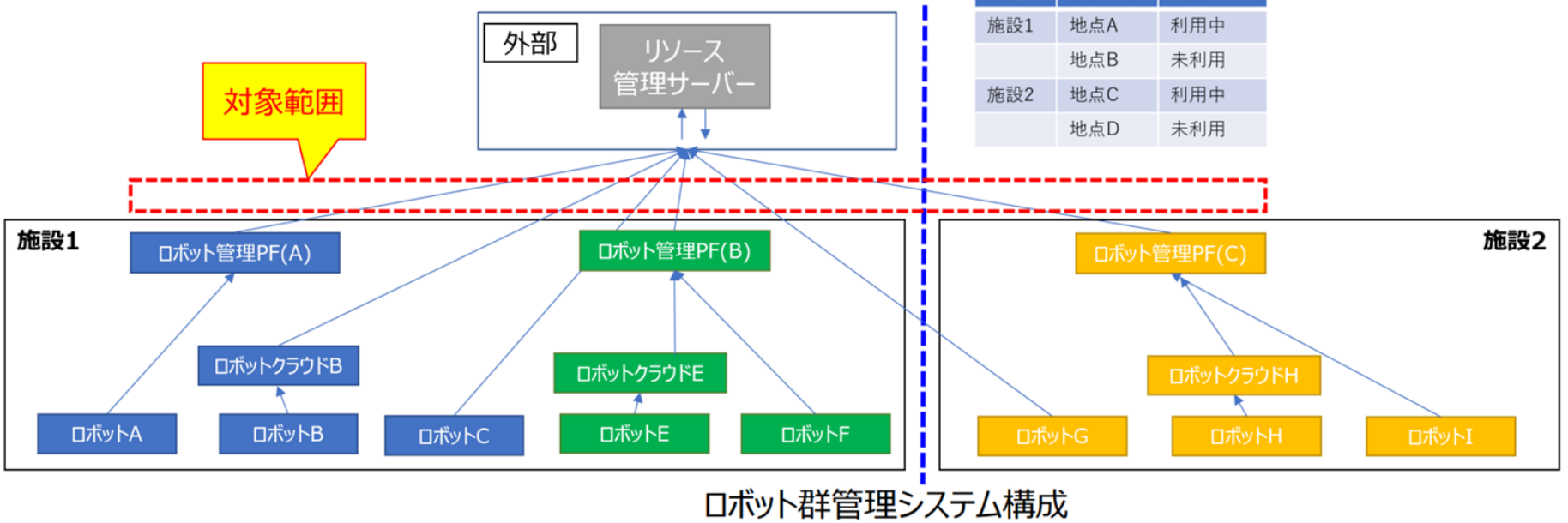
https://robot-friendly.org/wp/wp-content/uploads/2024/07/20240717_RFA_Presentation_2024.pdf

RFA群管理TCで議論中の内容

SoftBank

隘路などの1台のロボットしか通過ができない地点や領域を**リソース**として定義
その利用状態を管理する「**リソース管理サーバー**」とそのインタフェースの仕様を決定

建物	リソース	状況
施設1	地点A	利用中
	地点B	未利用
施設2	地点C	利用中
	地点D	未利用



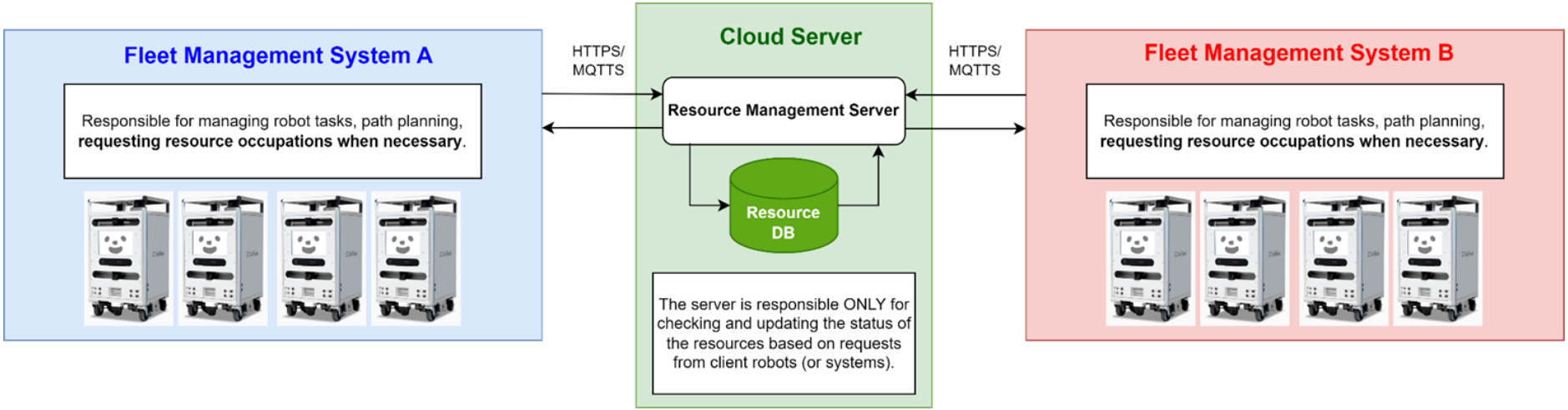
ロボット群管理システム構成

https://robot-friendly.org/wp/wp-content/uploads/2024/07/20240717_RFA_Presentation_2024.pdf

リソース管理サーバー

SoftBank

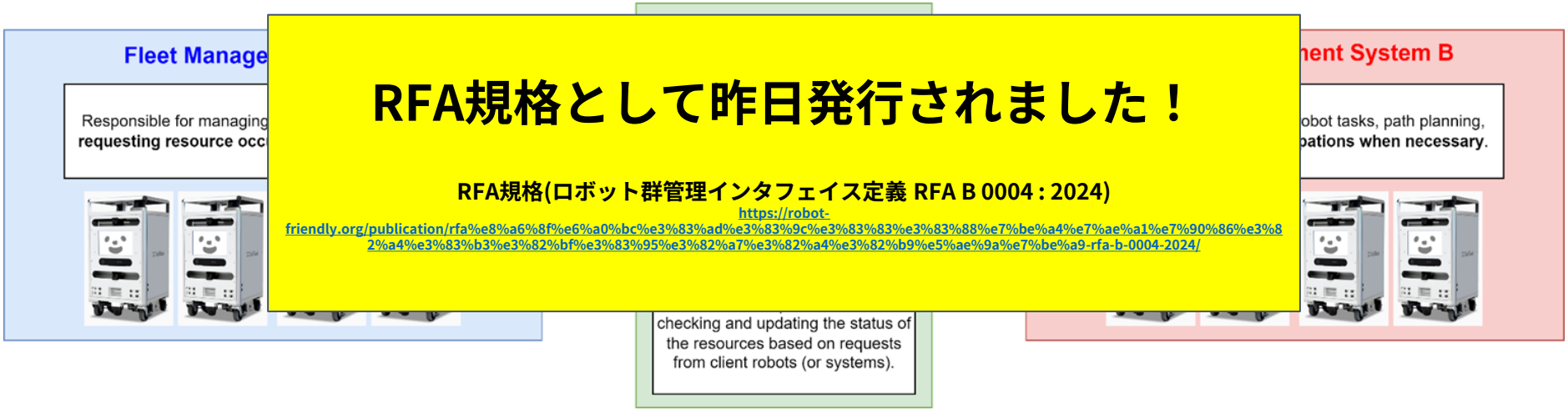
リソース管理サーバーは**リソースの状態の管理に関してのみ**責任を持つ
位置情報管理や経路計画やその修正は各集中管理システムが責任を持つ



リソース管理サーバー

SoftBank

リソース管理サーバーは**リソースの状態の管理に関してのみ**責任を持つ
位置情報管理や経路計画やその修正は各集中管理システムが責任を持つ



RFA規格として昨日発行されました！

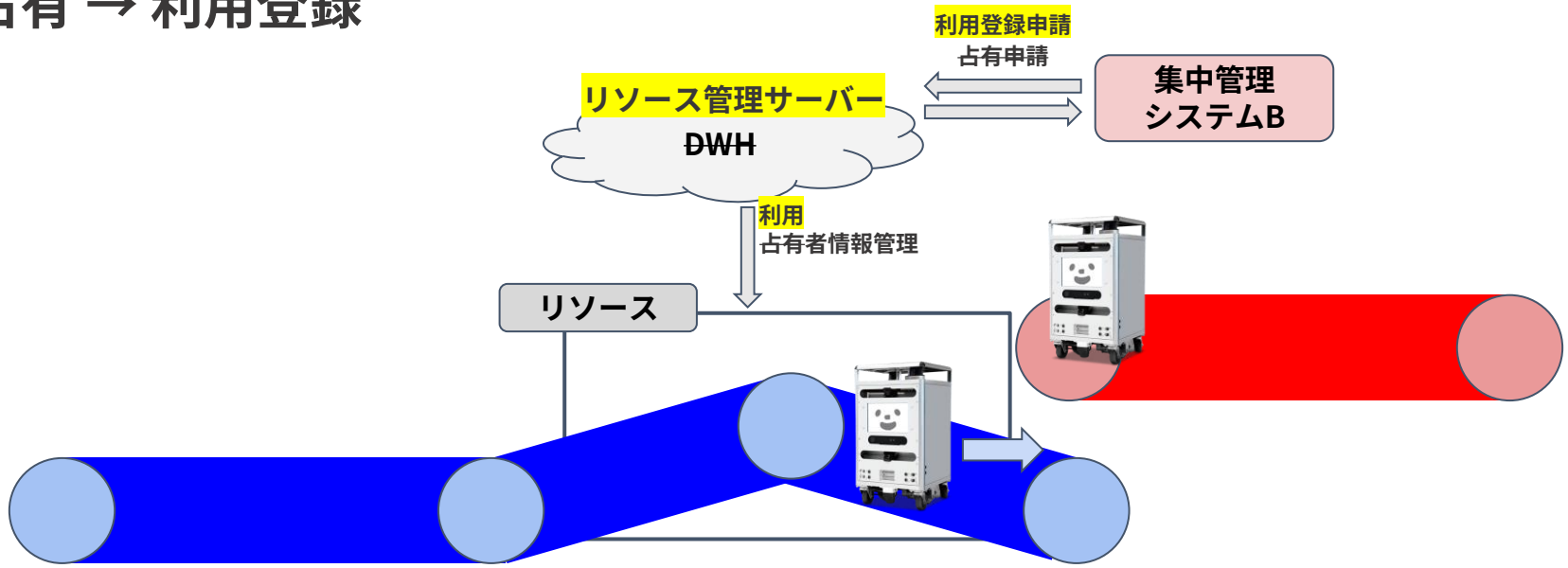
RFA規格(ロボット群管理インタフェイス定義 RFA B 0004 : 2024)
https://robot-friendly.org/publication/rfa%e8%a6%8f%e6%a0%bc%e3%83%ad%e3%83%9c%e3%83%83%e3%83%88%e7%be%a4%e7%ae%a1%e7%90%86%e3%82%a4%e3%83%b3%e3%82%bf%e3%83%95%e3%82%a7%e3%82%a4%e3%82%b9%e5%ae%9a%e7%be%a9-rfa-b-0004-2024/

リソース管理サーバー

SoftBank

先述の実証と実施したい流れは同様と捉えられる

- DWH → リソース管理サーバー
- 占有 → 利用登録

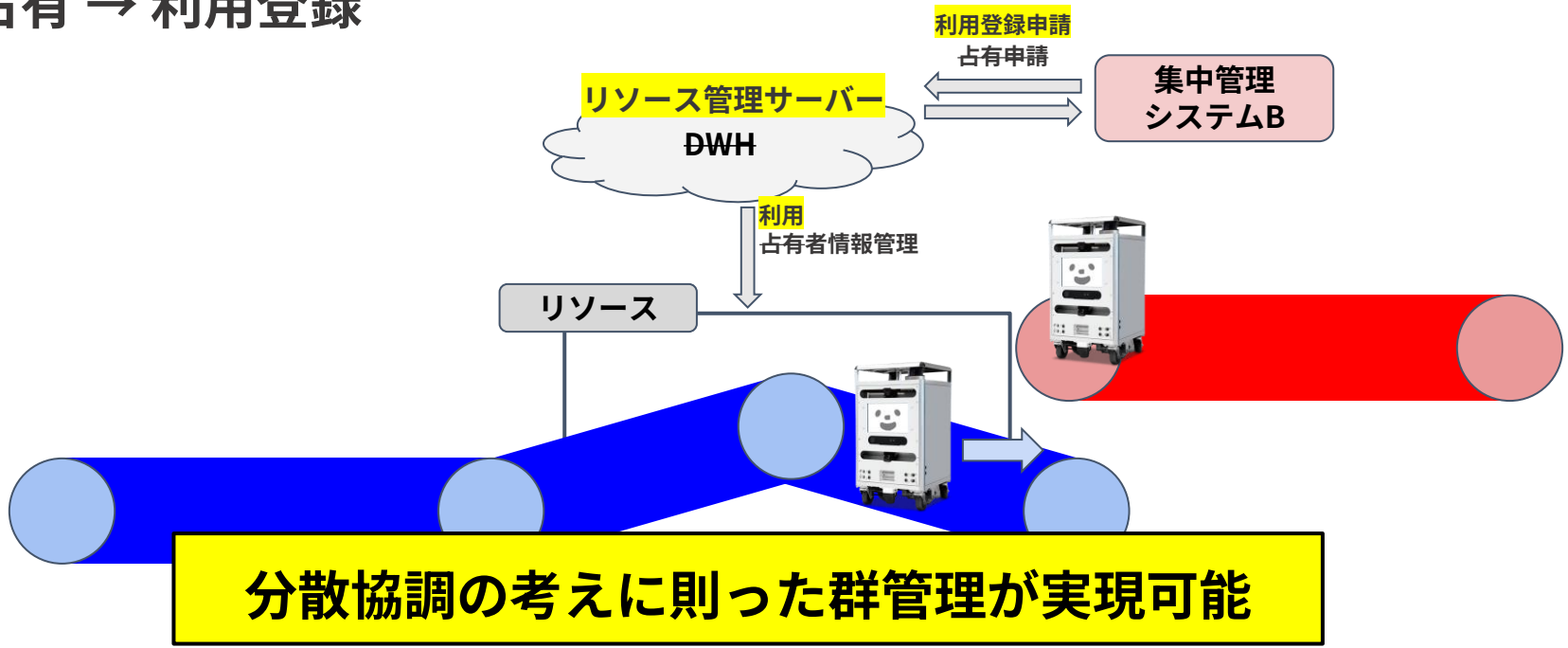


リソース管理サーバー

SoftBank

先述の実証と実施したい流れは同様と捉えられる

- DWH → リソース管理サーバー
- 占有 → 利用登録



Open-RMF × 分散協調

Open-RMFとは

Gazebo環境作成

地図と経路の作成

経路調停とアダプタを
通したロボット操作

ウェブアプリ操作

- ロボット群管理を目的として開発されている
- 提供機能の例
  - Gazebo環境と地図作成・経路設定のためのインタフェース
  - 配下の他のロボットとの経路干渉を考慮した経路計画とそれに基づいた走行指示
  - ロボットフリート・ドア・エレベーター各種を連携させるためのアダプタ機能とそのテンプレート
  - タスク指示や可視化用ウェブアプリ

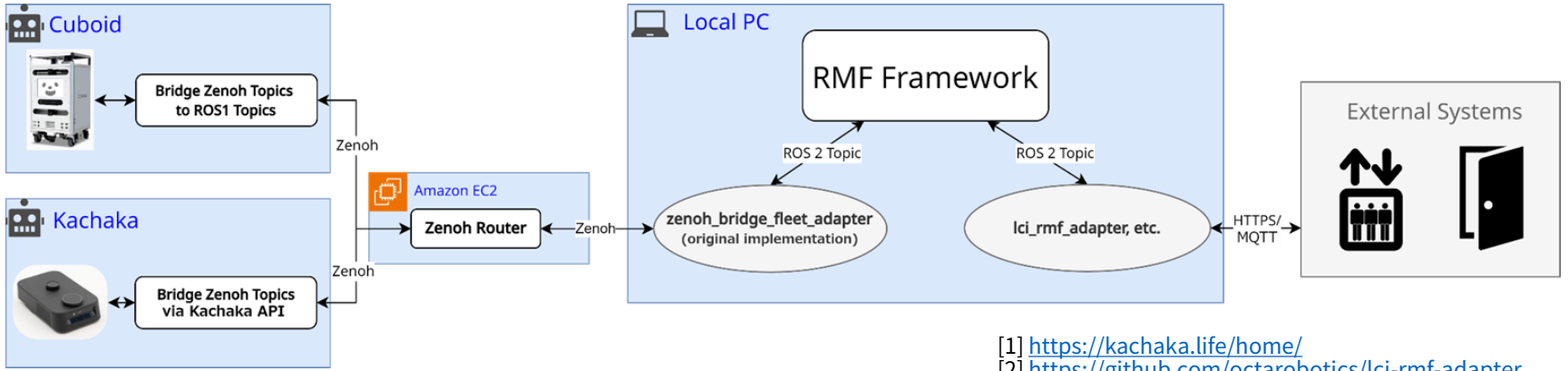
集中管理型の群管理が実現可能なOSS

と言える

https://www.open-rmf.org/
https://github.com/open-rmf/rmf_visualization

社内検証の基本構成

- Open-RMFを用いたCS室管理下ロボットの集中管理を視野に入れて実験を開始
- 以下のロボットを管理
  - Cuboid（ROS 1 / 2 対応内製ロボット、複数台保有）
  - カチャカ[1]（PFR社製、公式APIを利用して操作、3台保有）
- インフラ連携
  - 普段利用しているエレベーター連携システムLCIのRMF用アダプタ[2]を通して連携
- 実験時の基本構成は下図の通り
  - ローカルPC上でRMFフレームワークを実行
  - Fleet Adapterも同ローカルPC上で実行
  - ロボットはLTE経由でAWSインスタンス上で実行されたZenohRouter[3]にアクセス
  - ZenohRouterがトピックをパイプする役目を担う（ロボット↔RMF）

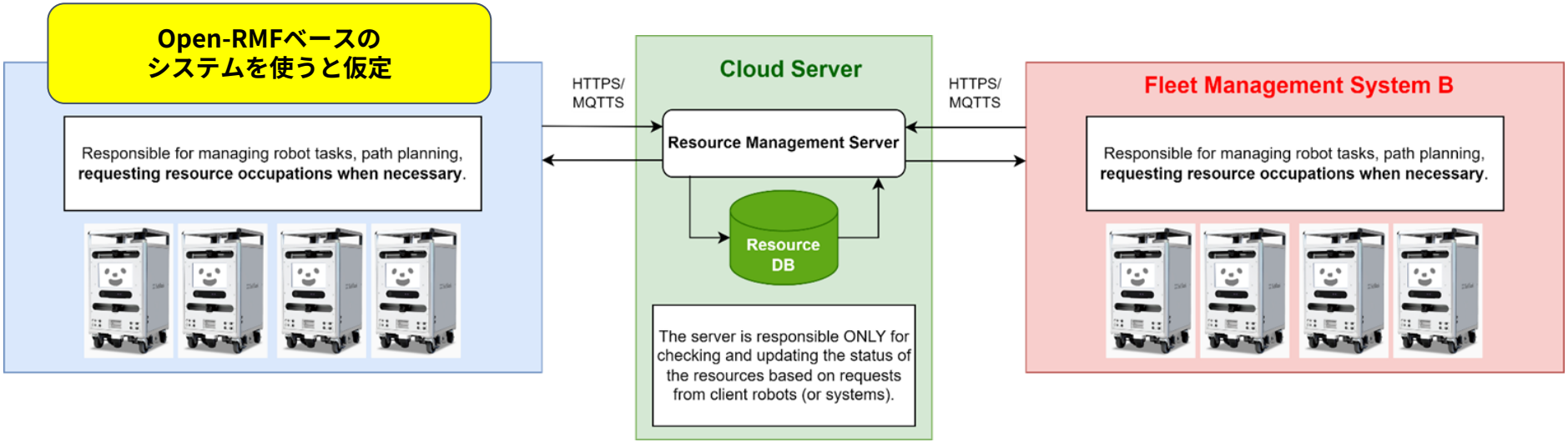


[1] https://kachaka.life/home/
[2] https://github.com/octarobotics/lci-rmf-adapter
[2] https://zenoh.io/docs/getting-started/installation/

集中管理システムとしてOpen-RMFを採用

SoftBank

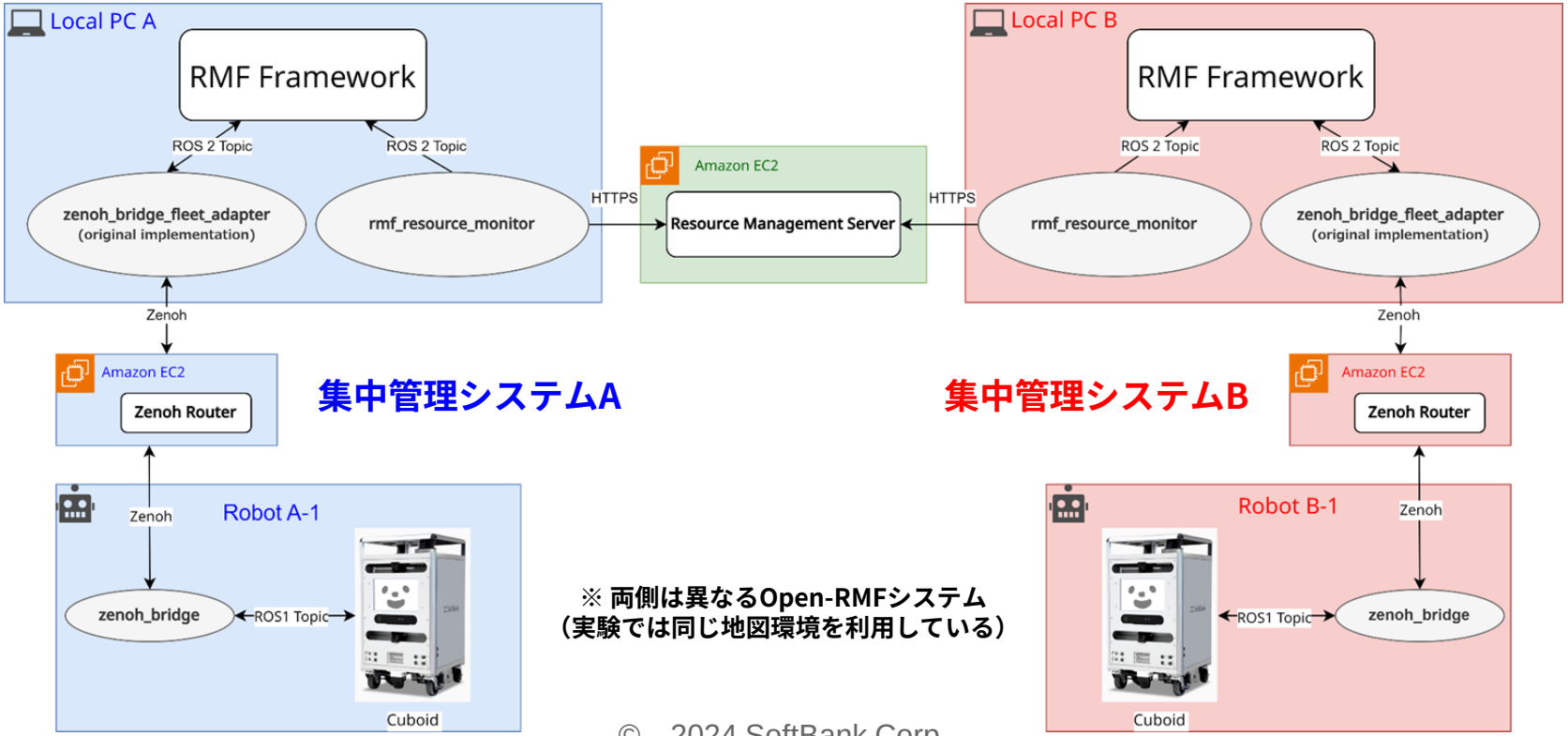
集中管理システムとしてOpen-RMFベースのものを採用した場合を考える
他の集中管理システムとのリソース管理サーバーを介した分散協調を試みる



Open-RMF間の経路調停の検証

SoftBank

社内検証において、以下のようなシステムを構想

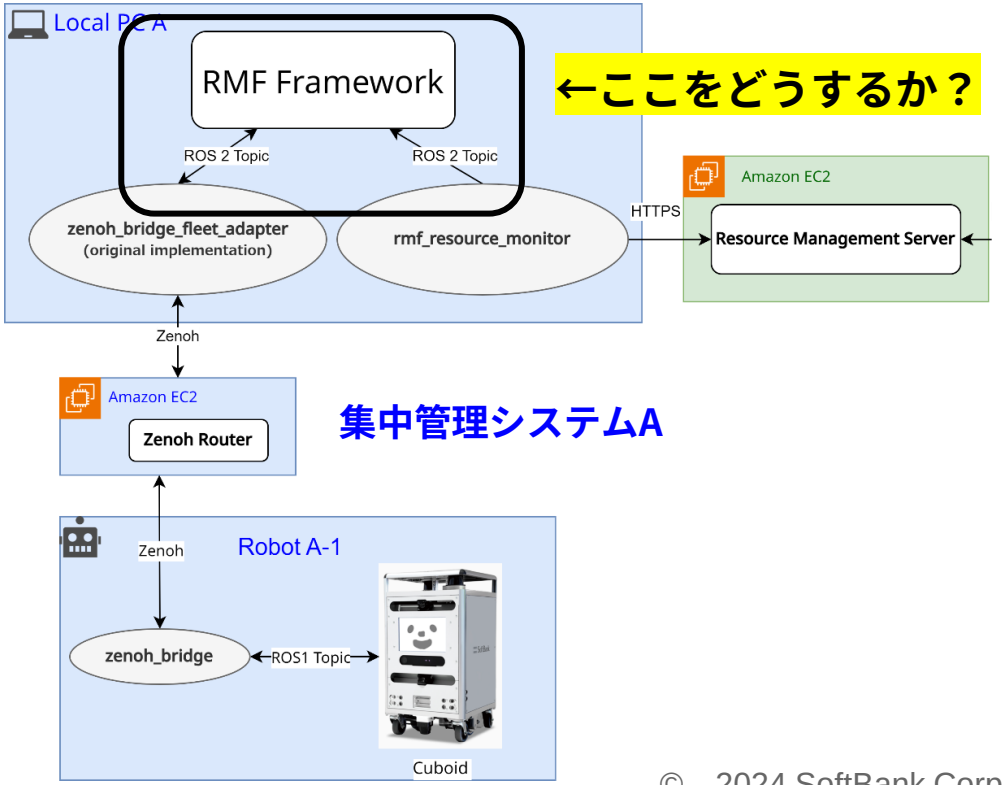


※ 両側は異なるOpen-RMFシステム
（実験では同じ地図環境を利用している）

Open-RMF間の経路調停の検証

SoftBank

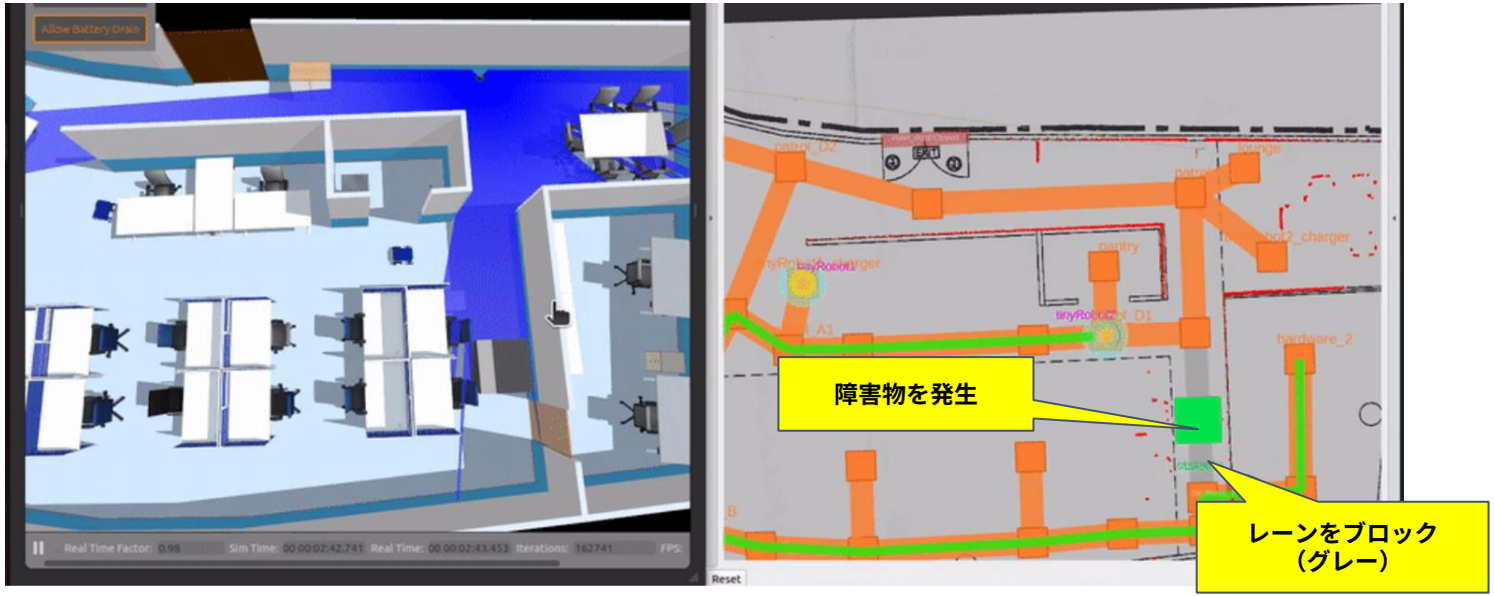
社内検証において、以下のようなシステムを構想



rmf_obstacle の利用

SoftBank

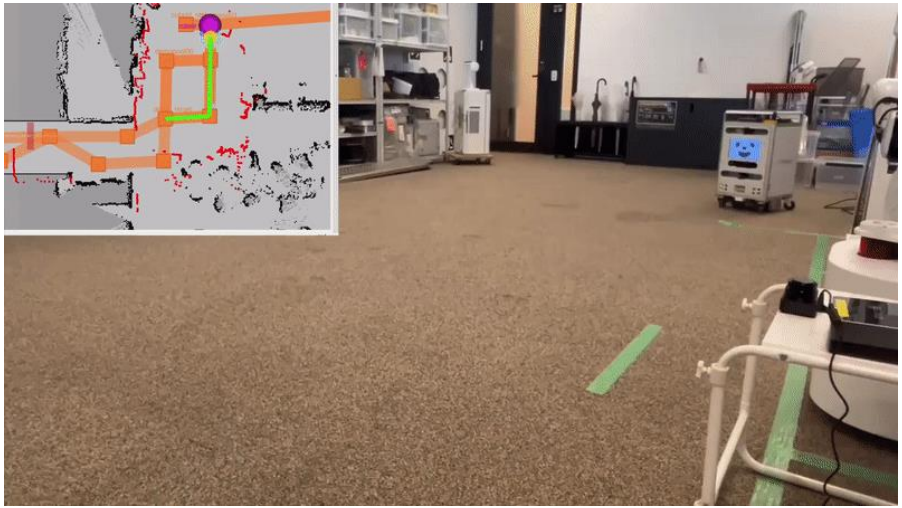
Open-RMF公式が提供している障害物情報を経路計画に反映させるためのリポジトリ
固定センサー（カメラ、LiDAR等）からの情報を連携させるためのパッケージを含む



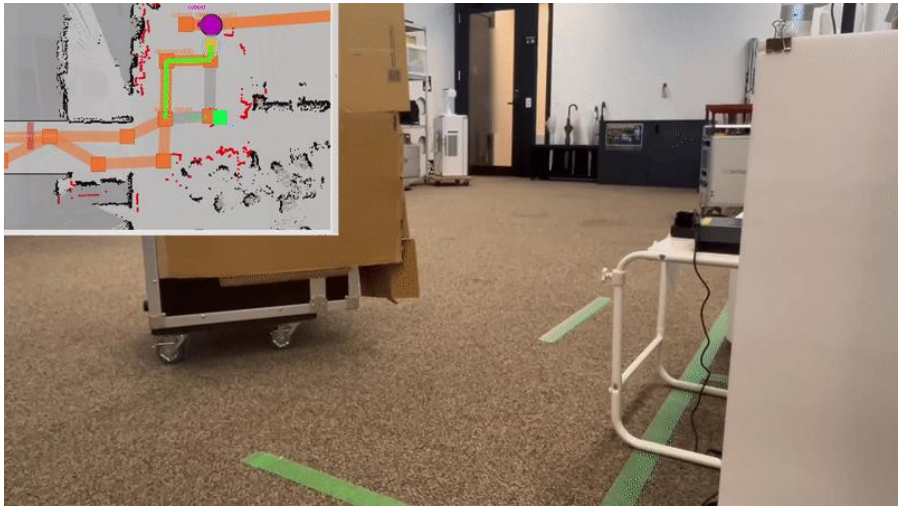
https://github.com/open-rmf/rmf_obstacle

rmf_obstacle の利用

以下は rmf_obstacle_detector_laserscan パッケージを用いて障害物のある経路を回避する例
画面右側に障害物・混雑度検知用インフラセンサーを模した2D LiDARが設置されている



障害物なし



障害物を検知

rmf_obstacle の利用

以下は rmf_obstacle_detector_laserscan パッケージを用いて障害物のある経路を回避する例
画面右側に障害物・混雑度検知用インフラセンサーを模した2D LiDARが設置されている

リソース登録失敗 = 障害物の発生
…としてしまえばそのまま使えるのでは？

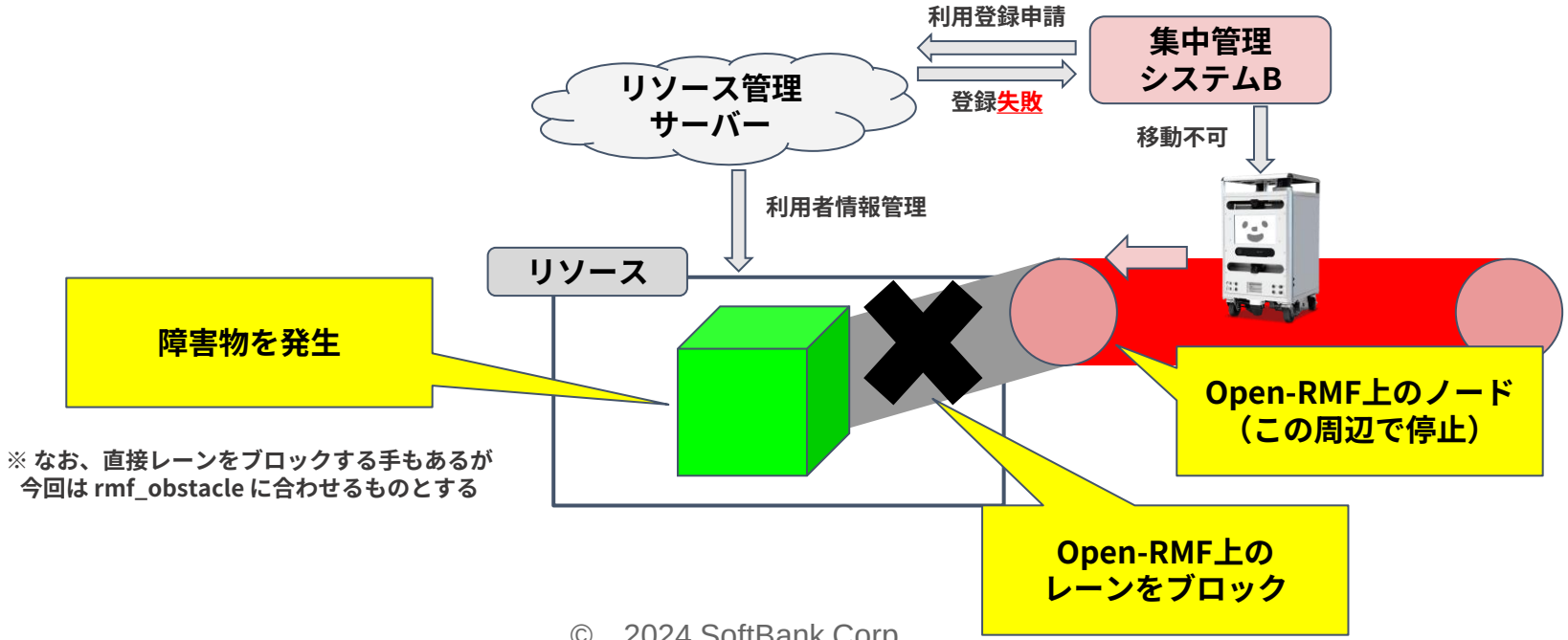
障害物なし

障害物を検知

rmf_obstacle の利用

SoftBank

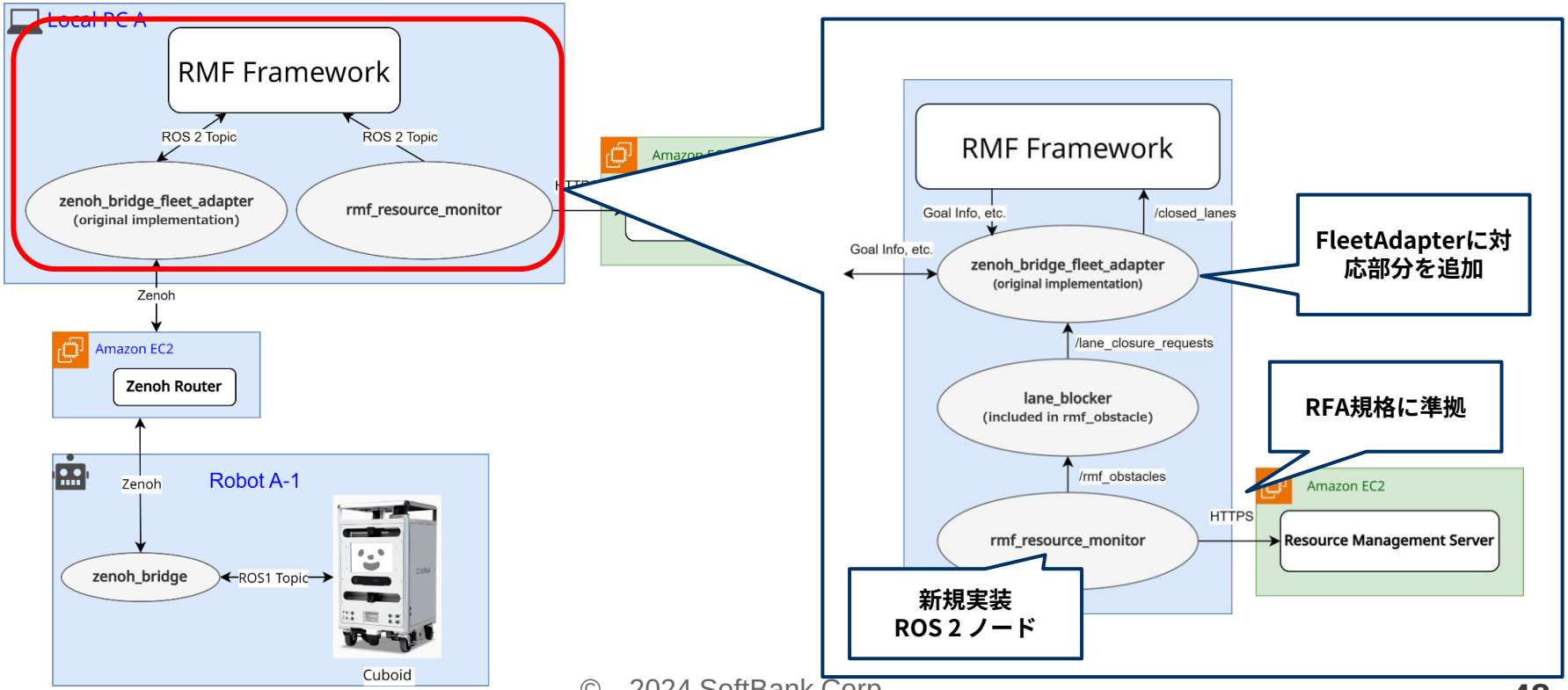
リソースに近づいたタイミングでリソース管理サーバーにアクセス
登録に失敗した場合にのみ障害物を発生させる
これにより、登録に成功するまでロボットを停止させる挙動を実現できる



Open-RMF間の経路調停の検証

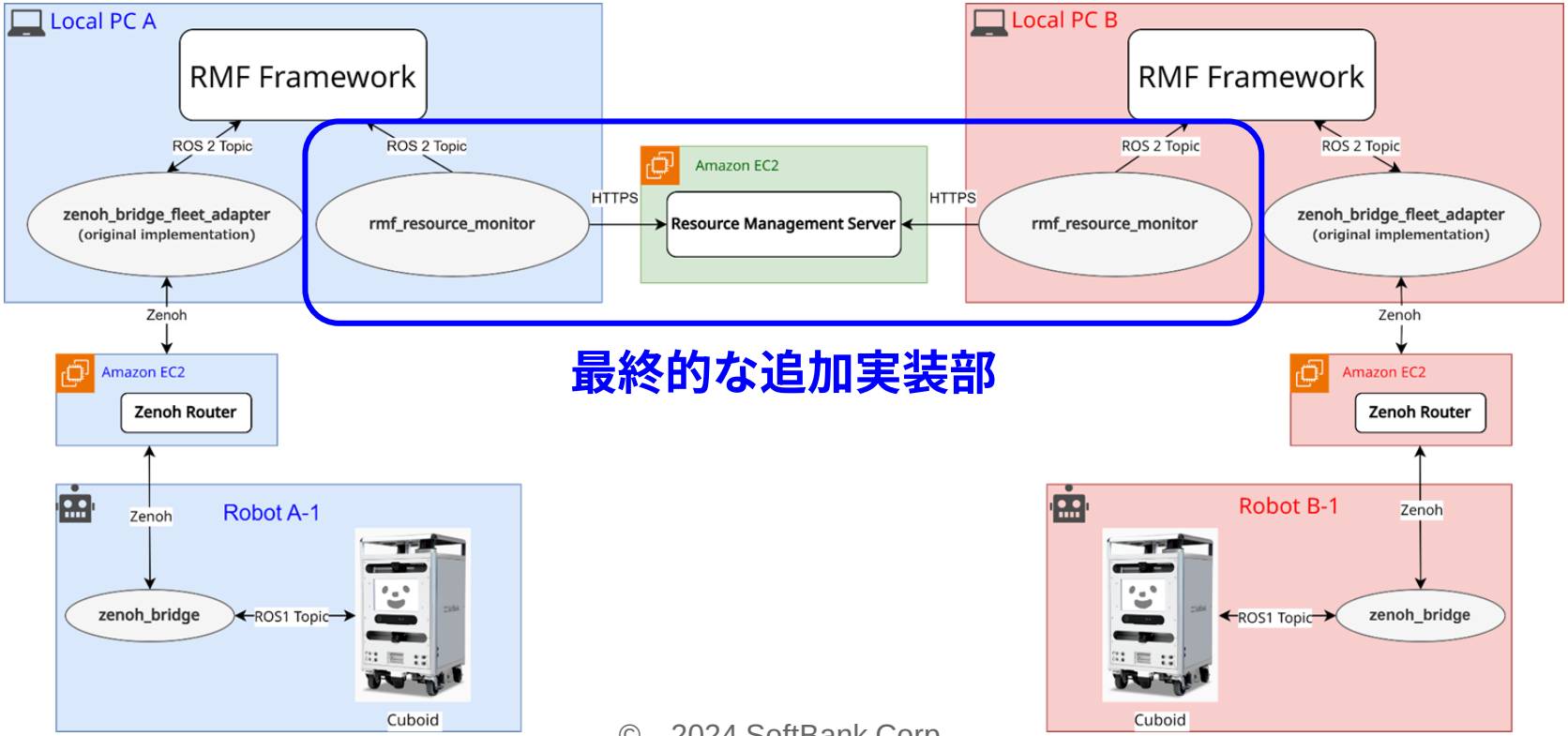
SoftBank

rmf_obstacle の仕組みを流用する形で赤枠部分を実現



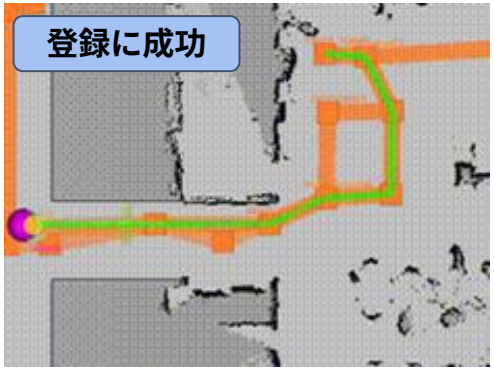
Open-RMF間の経路調停の検証

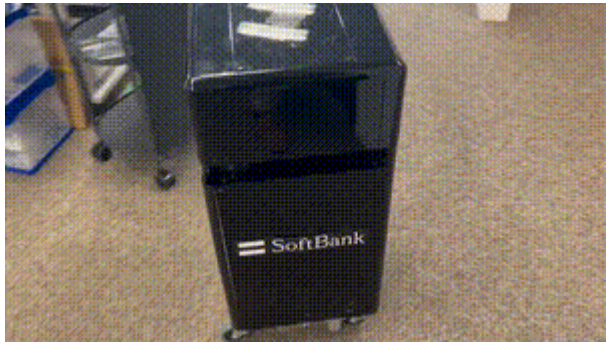
実験にあたり、青枠部分のみを追加で実装すれば十分だった

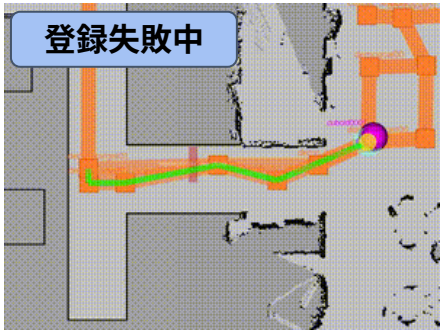


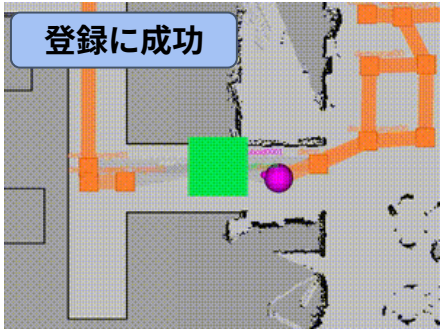
実験動画

隘路・風除室のようなリソースを想定









システムA側が先にリソースを取得
障害物は発生せず素通り

白い機体がシステムA配下
黒い機体がシステムB配下

システムB側はリソース取得に失敗
成功するまで障害物が発生

今回利用したリポジトリの紹介

SoftBank

- rmf_obstacle
  - Open-RMF公式リポジトリ
  - カメラやLiDARなどの事前に設置したセンサ情報から得た障害物情報を経路計画に反映させることを主な目的とする
  - https://github.com/open-rmf/rmf_obstacle
- resource_management_server
  - 内製の実験用リソース管理サーバー（ROS外）
  - RFA規格のインタフェースに従い試験用に一部機能を実装
  - https://github.com/sbgisen/resource_management_server
- rmf_resource_monitor
  - Open-RMFの rmf_obstacle に対応する形で内製したROS 2ノード
  - 事前に用意したリソース情報に基づき、リソース接近時にサーバーへの登録と障害物の発生を実施
  - https://github.com/sbgisen/rmf_resource_monitor

課題とまとめ

リソース管理サーバーの課題

SoftBank

リソース管理サーバーには実証と共通の課題が存在すると考えられる

- 台数が増加するとリソース周囲が混雑してしまう
- 待機位置関連をどうルール化するか（e.g. エリアあたりで台数制限？）
- 何より、各集中管理システムで導入されていなければ意味をなさない

リソース内部での
デッドロックは防げるが…

人の通行の妨げとなる位置で
スタックが発生する恐れあり

まとめ

SoftBank

- 昨今の国プロやRFAにおいては責任分界の観点などからOpen-RMFのような「集中管理」に該当する群管理システム同士をつなぐ「分散協調」型の群管理について議論されており、その規格も発行されている
- ROSの文脈で言えば、Open-RMFベースの集中管理システムにおいても、rmf_obstacleの仕組みを流用することで「リソース管理サーバー」を介して他の集中管理システムとの分散協調制御が実現可能であることが示せた
- 既知の問題点やRFAでの議論がまとまっていない点もあり、これから更なる実証実験、仕様やガイドラインの策定・発行が進められていく見込みである
- 多くのベンダーに使ってもらわなければ価値を生まない標準化であるため、検証機会を増やす意味でもOSSが寄与できる側面があるのではないかと考えている

ブース展示実施中

SoftBank

本取り組みについてご質問のある方もぜひ足をお運びください

CUBOID

ROS-Sler.com

ROS-SI推進課ホームページ
http://www.ros-sier.com/

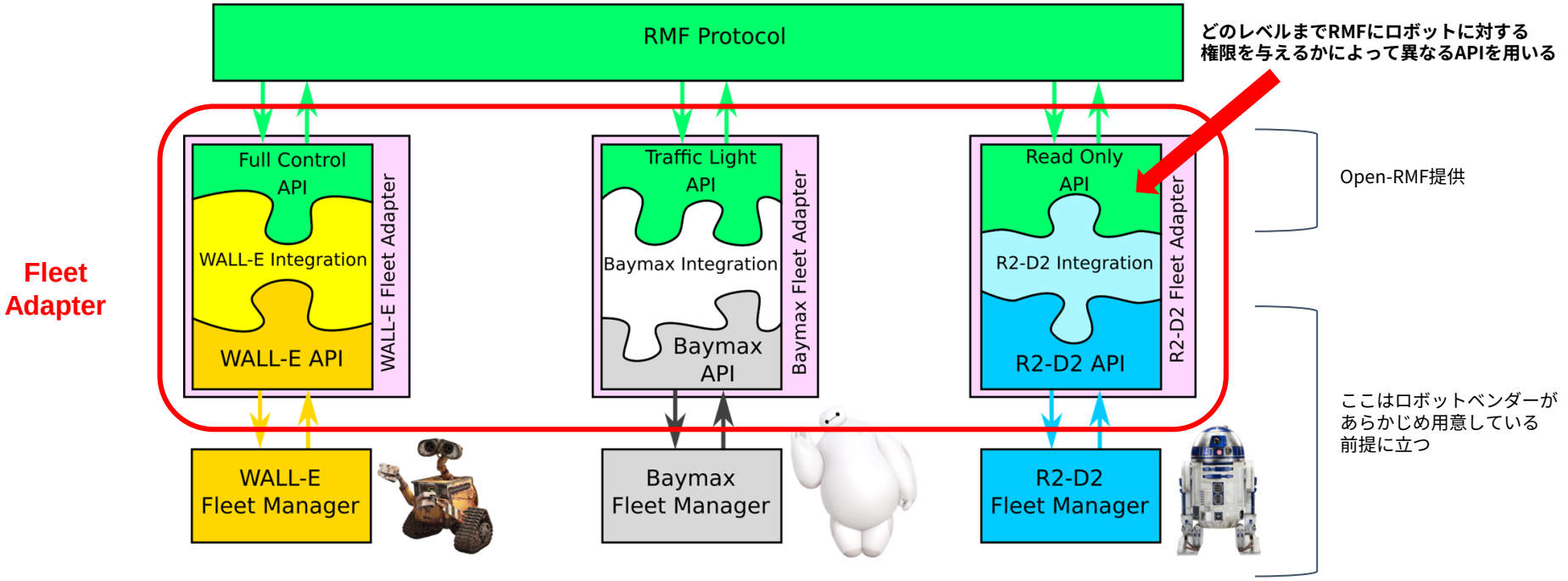
APPENDIX

補足：Open-RMF

Open-RMFのFleetAdapterについて

SoftBank

RMFサーバーと各種管理システムを連携させるためのインタフェース



https://docs.google.com/presentation/d/1Lt79xlM_XkITmURSbI5hkAAgnjSX8dHKBkgvz3x3Uzw/edit#slide=id.g1648edacea5_0_969 （一部改変）

Open-RMFの提供するアダプタについて

SoftBank

各種ロボット、ドア・エレベーターなどのインフラ設備との連携を容易にするために、Open-RMFは「アダプタ」と呼ばれる仕組みを用意しており、それを導入しやすくするための各種テンプレートを用意している

- Fleet：fleet_adapter_template
- Door：door_adapter_template
- Lift：lift_adapter_template

制御に際して送られてくるROSトピックが事前に定められており、テンプレートに従ったアダプタを記述することで新規ロボットや設備連携の導入が可能

現在公開されているアダプタの実装例一覧

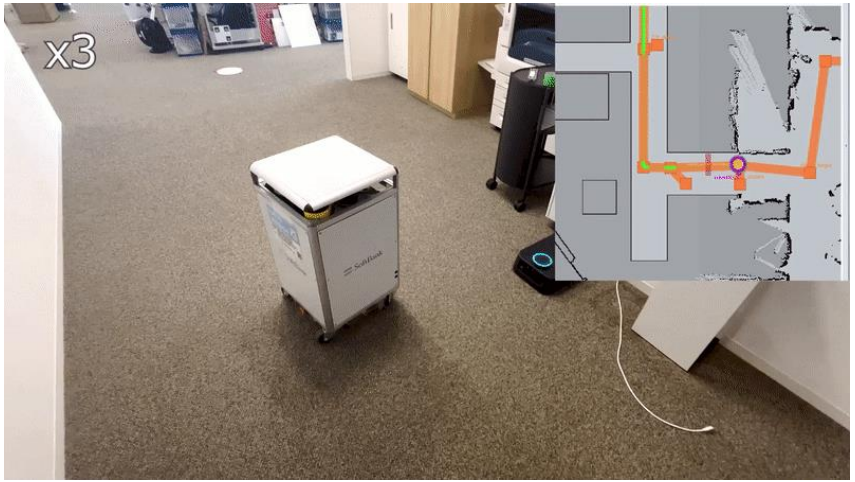
https://github.com/open-rmf/awesome_adapters

補足：インフラ連携の実施

SwitchBotとの連携

SoftBank

- SwitchBot製品の1つである開閉検知センサを手動扉脇に装着
- SwitchBot APIを利用して開閉状態をRMFに通達することのできる（なんちゃって）DoorAdapterを作成
- ドア開けはあくまで手動



https://www.switchbot.jp/pages/switchbot-contact-sensor?srsltid=AfmBOoooUx1in2KXHaYoKEtLYZi3fvgBDoI9UnlddoM05XWYshUP0wpk
https://github.com/sbgisen/door_adapter_switchbot

Octa Robotics LCIとの連携

- Octa Robotics 社製LCIシステム（RFA規格準拠）が当社オフィスの一部エレベーターにて導入されている
- Octa Robotics 社公式実装のRMF用LiftAdapterを利用して連携

https://www.octa8.jp/service/
https://github.com/octarobotics/lci-rmf-adapter

補足：Zenoh

Zenohとは

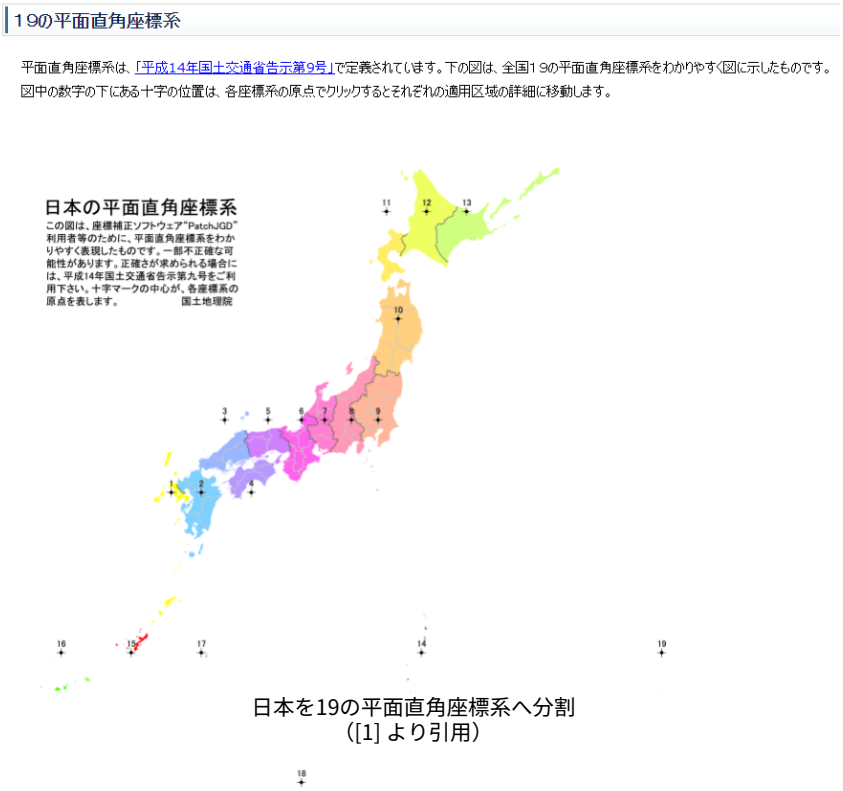
SoftBank

- 主にZettaScale Technology社が開発している通信フレームワーク
- ROS/ROS 2内でも用いられるPub/Sub形式のプロトコルが利用可能
- エッジコンピューティングやIoTでの利活用を念頭に以下を実現
  - 低オーバーヘッド
  - 高エネルギー効率
  - 信頼性と柔軟性
- 当社の実験ではNAT越えのためにも活用
- 昨今特にROS 2での利用と標準採用が推進されている
  - **一貫したデータ管理**
    システム全体で採用することでシームレスなデータフローが実現可能
  - **高効率**
    高速なスループットと低遅延を誇り、リアルタイムアプリケーションに最適
  - **スケーラブルな通信**
    分散型設計により、大規模なネットワークでも効率的にスケーリング可能

補足：19座標系を用いた緯度経度の変換

ローカル座標への変換に19座標系を活用

SoftBank



日本を19の平面直角座標系へ分割
（[1] より引用）

- 日本での位置の基準となる測地系の1つとしての平面直角座標系が定められている
- これは全国を19の座標系に区分したものであり、「19座標系」とも呼ばれる[1][2]
- 例えば東京近辺などはEPSGコード6677の測地系としてPythonライブラリpyproj[3] などから扱うことができ、実証実験ではこれを用いてローカル座標からグローバル座標（＝緯度経度）への変換を実施した
- 当然ながら、計算にはローカル地図上のある基準点における緯度経度情報並びに経線（緯線）からの傾き（収束角）情報が必要となる

[1] https://www.gsi.go.jp/sokuchikijun/jpc.html
[2] https://nkgias.com/ja-jp/NetHelp/index.html#!Documents/%E5%B9%B3%E9%9D%A2%E7%9B%B4%E8%A7%92%E5%BA%A7%E6%A8%99%E7%B3%BB%EF%BC%91%EF%BC%99%E5%BA%A7%E6%A8%99%E7%B3%BB.htm
[3] https://pyproj4.github.io/pyproj/stable/

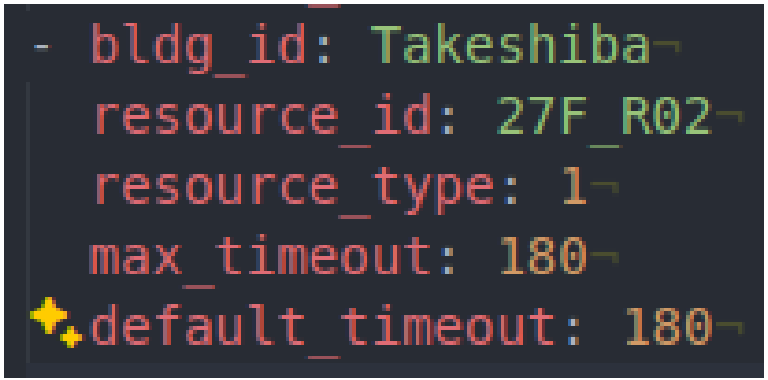
補足：リソースの定義方法

リソースの定義方法について

SoftBank

現在RFA規格内ではインタフェース内でリソース管理サーバー側の利用する情報については定義がなされている

以下は各リソースを管理するに当たって事前に取り決める情報の例である



仕様に基づき、リソース毎に各種情報を設定

リソースの定義方法について

SoftBank

一方で群管理システム側が利用する情報については各個定義となる
今回、Open-RMFで利用するに当たっては下に示すような内容を用意した

```yaml
# Define a list of resources that the robot needs to register when passing.
- resource_id: "R01"  # Should match the resource ID used by the resource management server
  floor_id: "1F"  # Should match the level name used by Open-RMF
  center_x: 0.0
  center_y: 0.0
  # Default values will be used for the following values when unspecified
  size_x: 2.0  # The bounding box size of the published obstacles
  size_y: 2.0
  size_z: 2.0
  registration_distance: 3.4  # If this is defined, the release_distance must also be defined
  release_distance: 4.0  # Must be larger than the registration_distance
```

rmf_resource_monitor パッケージでは必要な情報を独自に定義している
Open-RMFベースの場合に限らず、リソースの位置情報については各システムごとに設定する必要がある

End of File