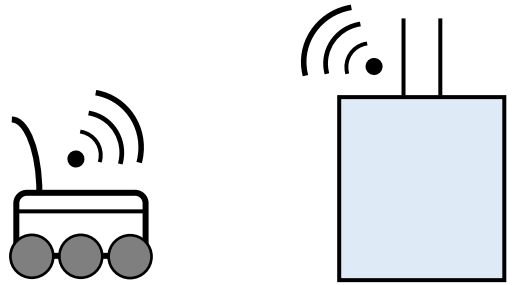
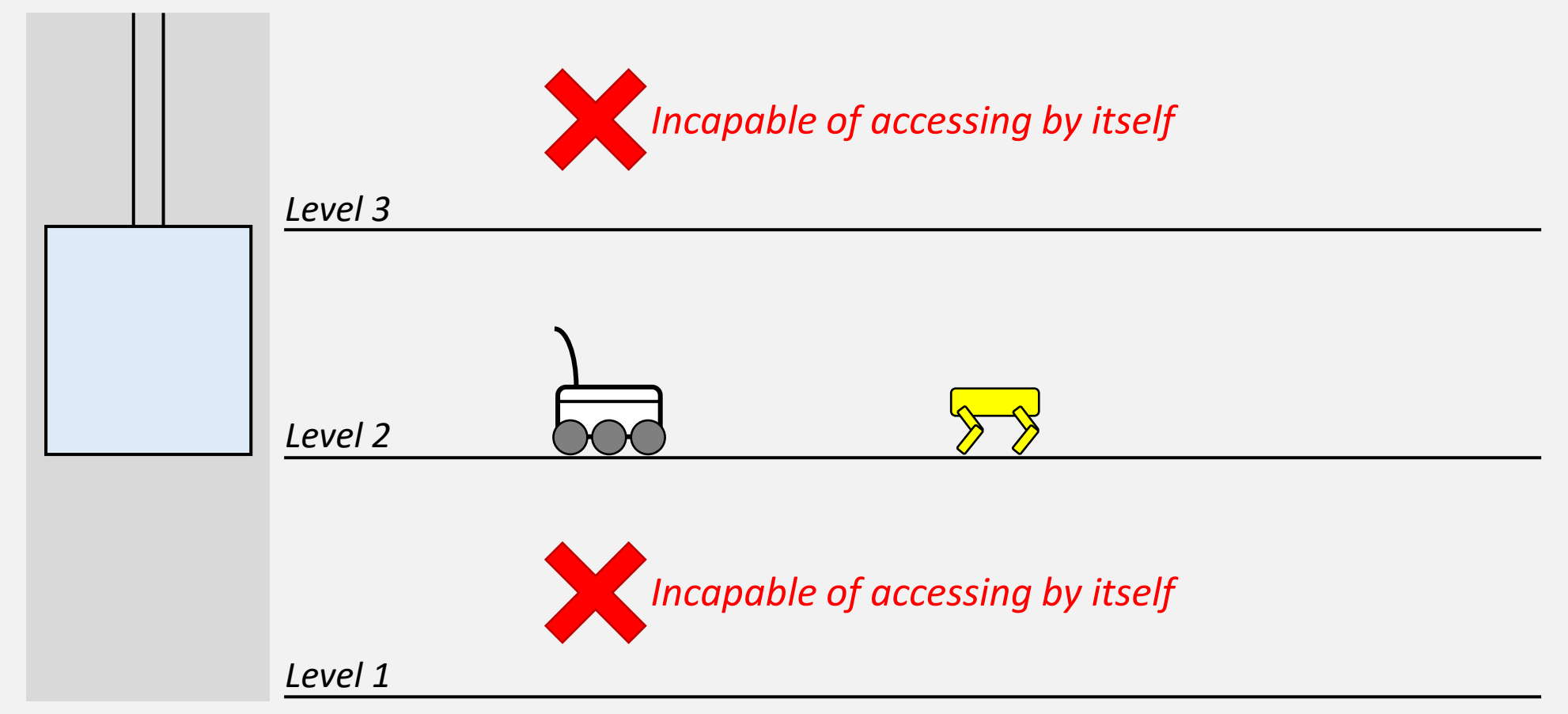
Developing an add-on kit to allow robots to operate legacy elevators



Roy Chenyu Luo

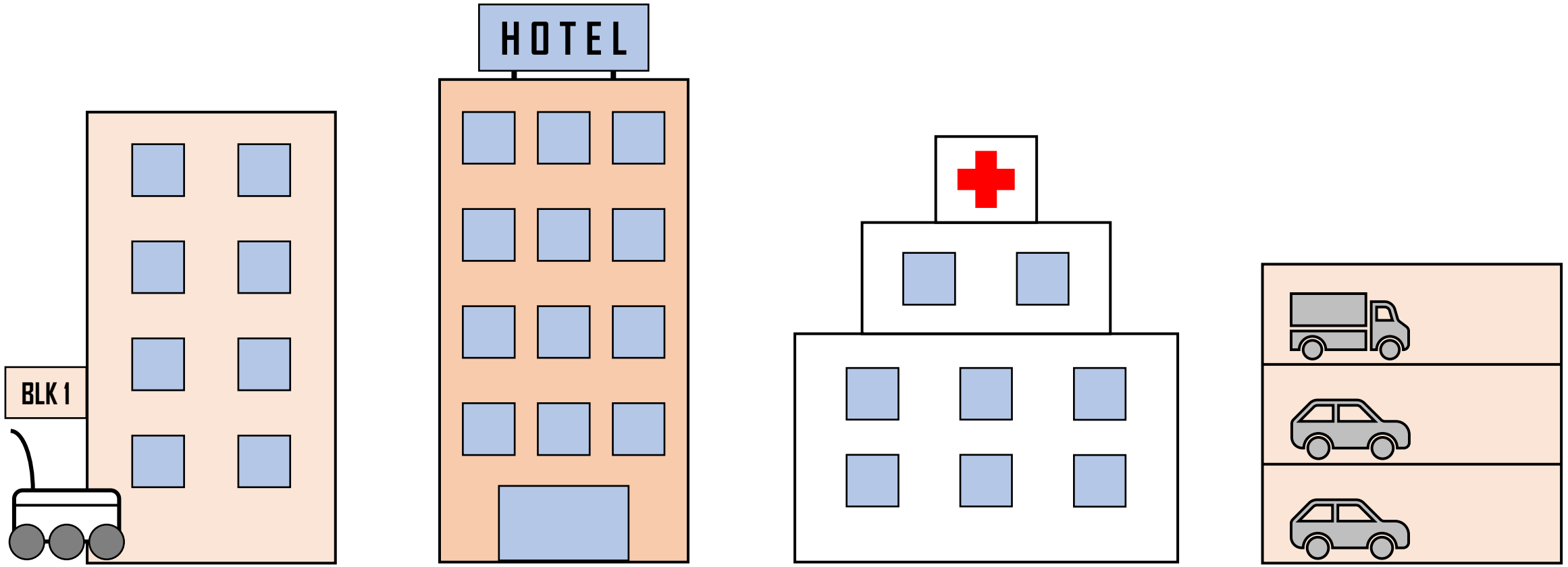
Mentored by: Hee Yong Siong

Rationale



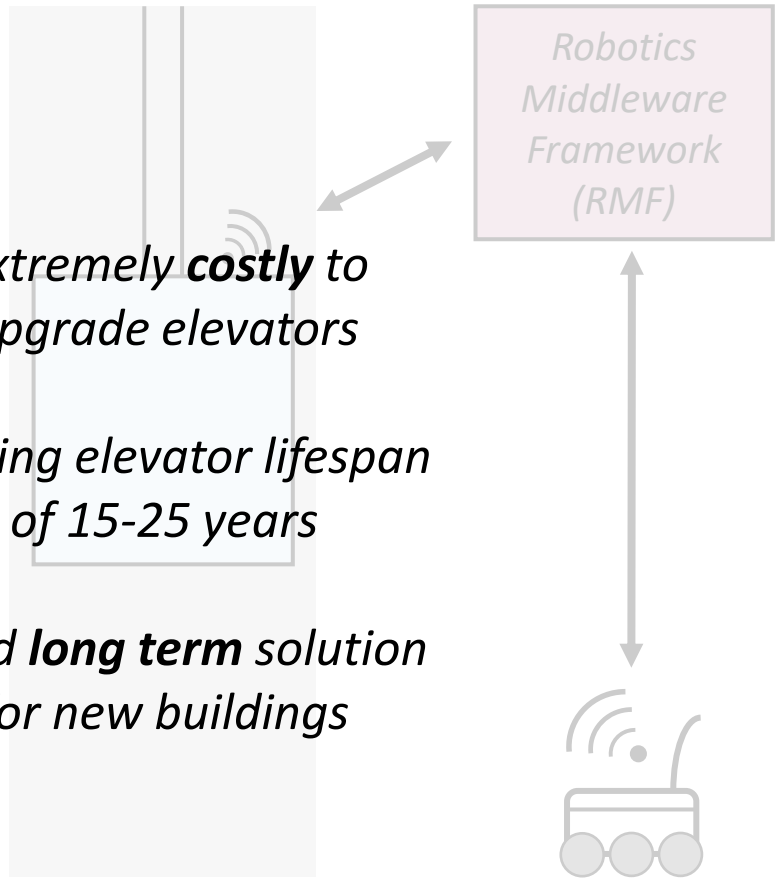
*Many robots in the market are **incapable of accessing** multiple floors by itself and are restricted to operating on a **single floor**.*

Rationale



*In a city with many **high-rise buildings** and elevators, there are many potential applications for a solution.*

Solution: RMF Compliant Elevators



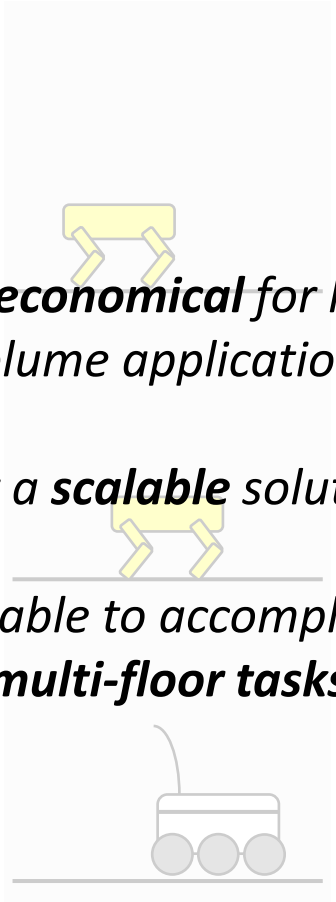
*Extremely **costly** to upgrade elevators*

Existing elevator lifespan of 15-25 years

*Good **long term** solution for new buildings*

Upgrading to smart elevators

Alternative Solutions



***Uneconomical** for low volume applications*

*Not a **scalable** solution*

*Unable to accomplish **multi-floor tasks***

Multiple robots to serve multiple floors

***Costly** to upgrade all robots*

*Not a **scalable** solution*

***Impractical** for some types of robots*

Upgrading (all) robots to operate elevators

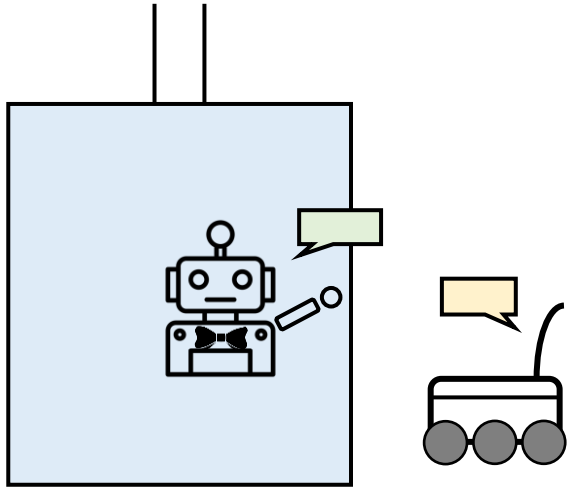
***Cheaper** alternative to upgrading lift*

*Easily **scalable***

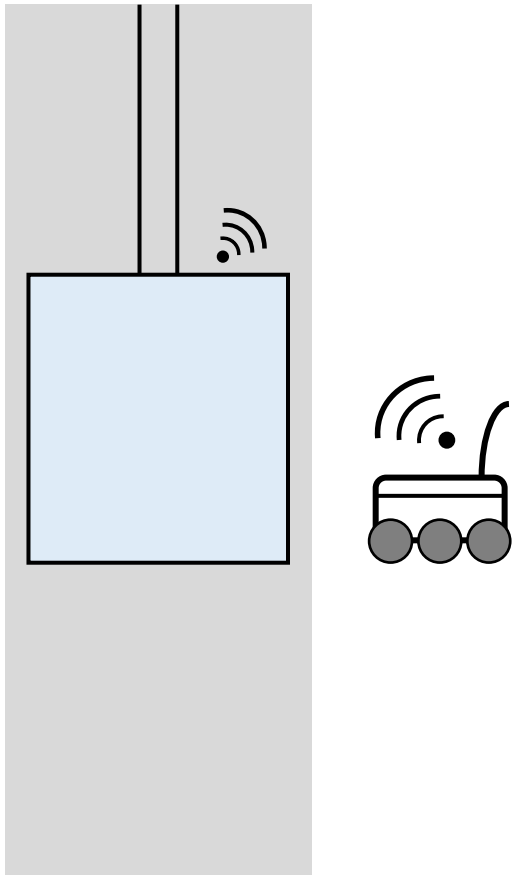
*Good **short-mid** term solution*

Create a custom add-on kit for existing elevators

Idea in a Glance



*Add-on Kit acts like a **personal butler** operating the elevator*

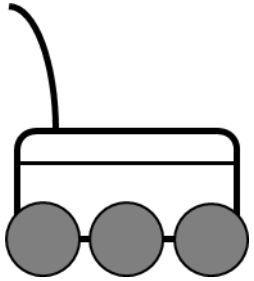


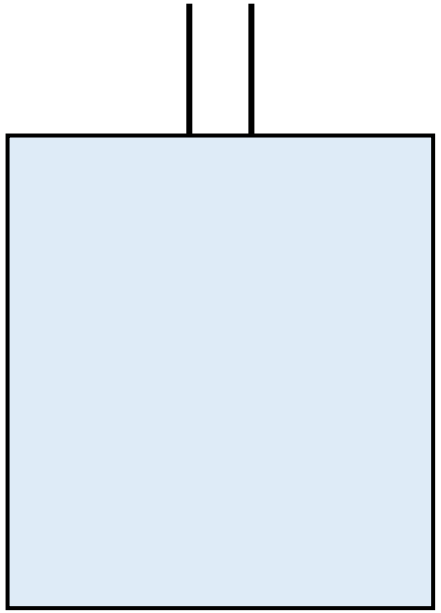
*A key enabler is **connectivity** inside and outside elevator*

Workflow Overview

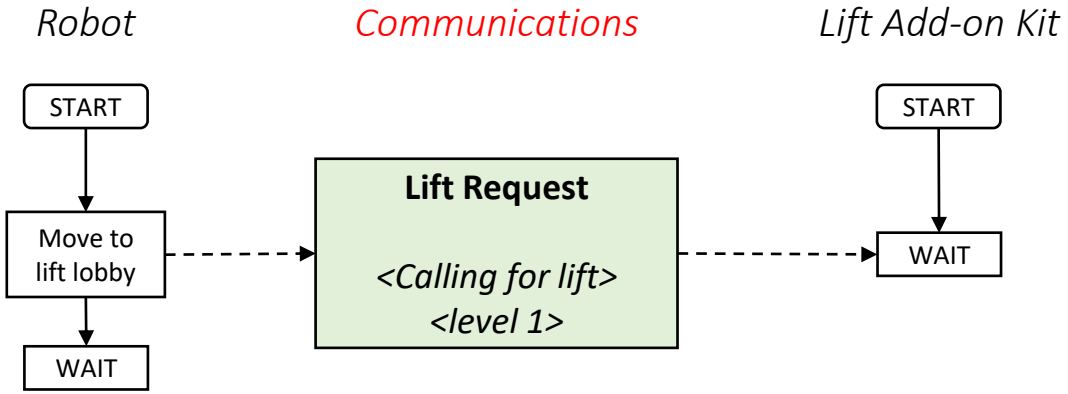
Robot | *Communications* | *Lift Add-on Kit*

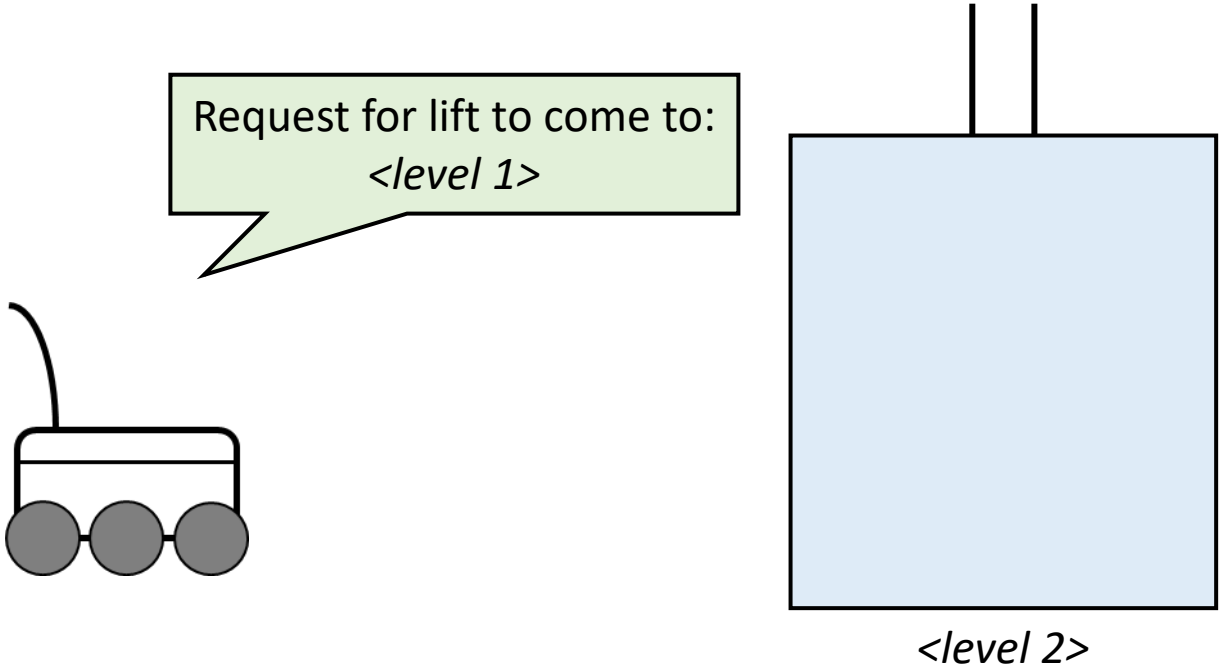
START | START



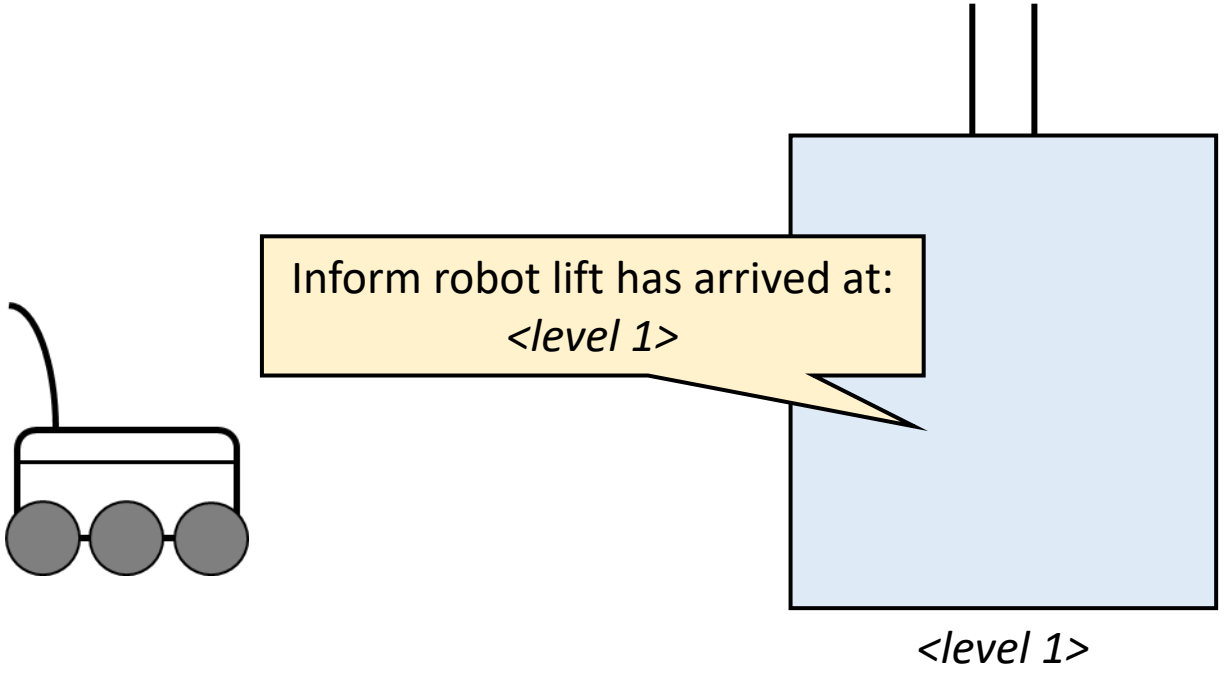


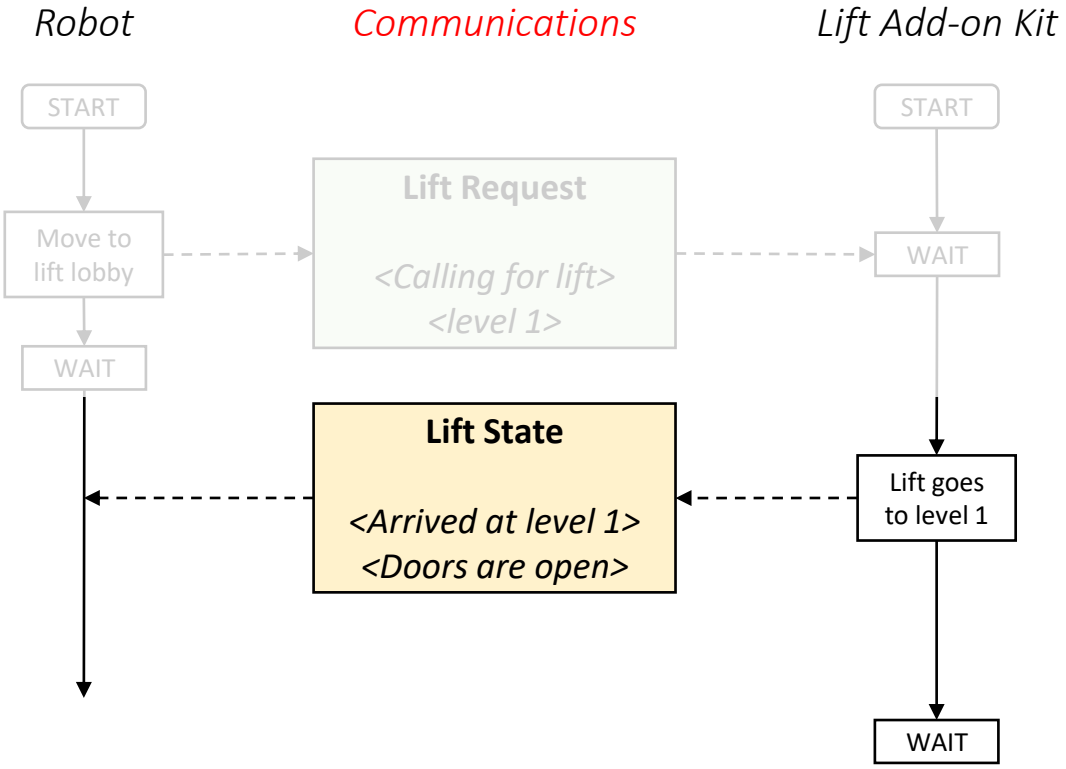
Workflow Overview



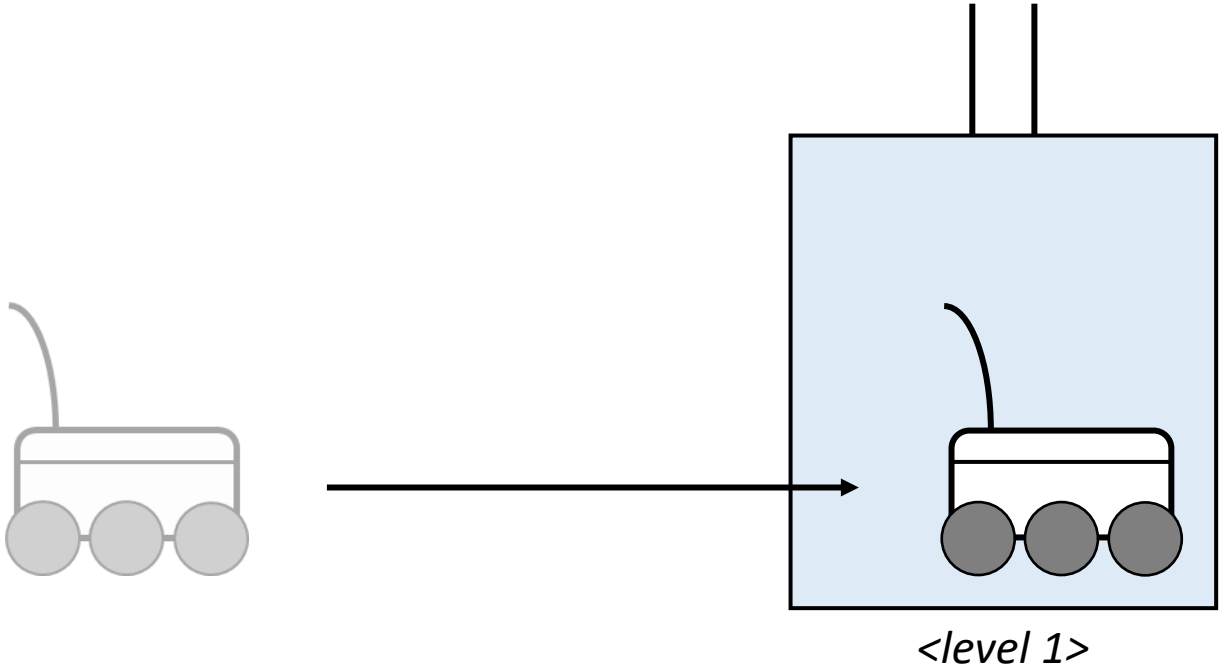


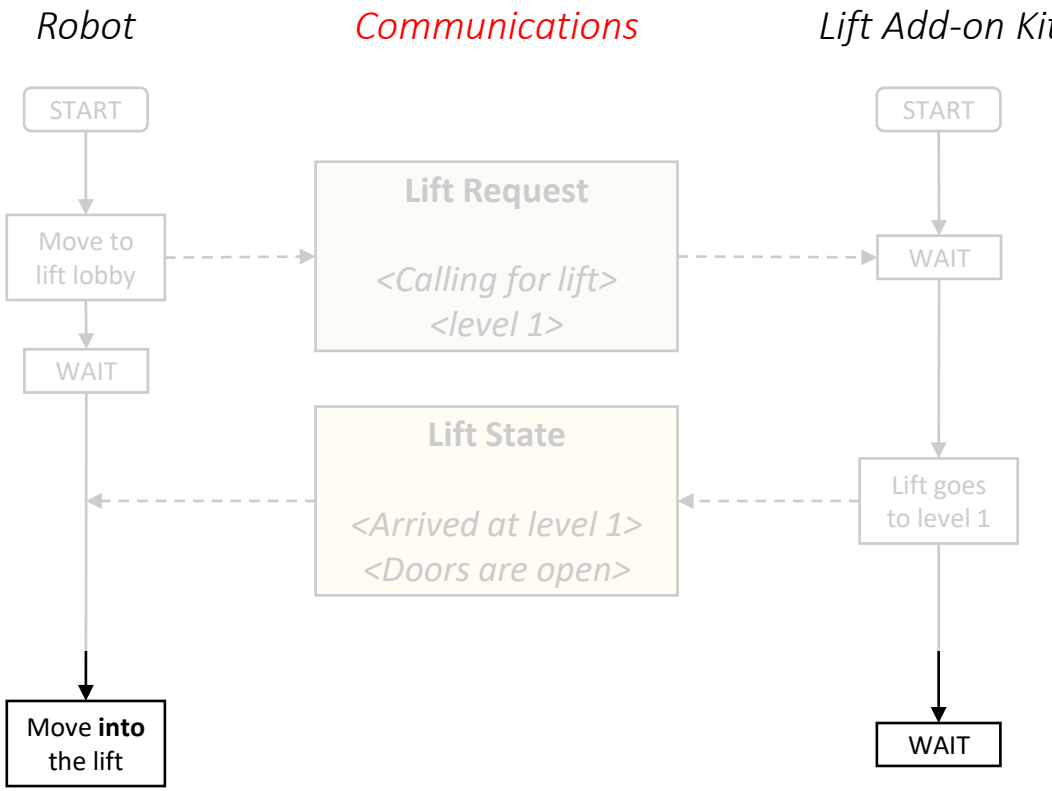
Workflow Overview



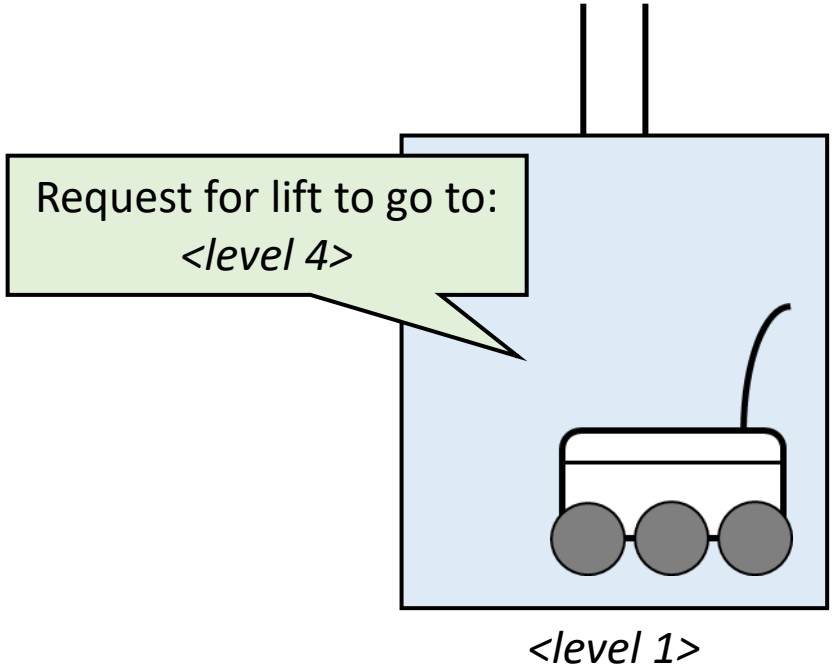


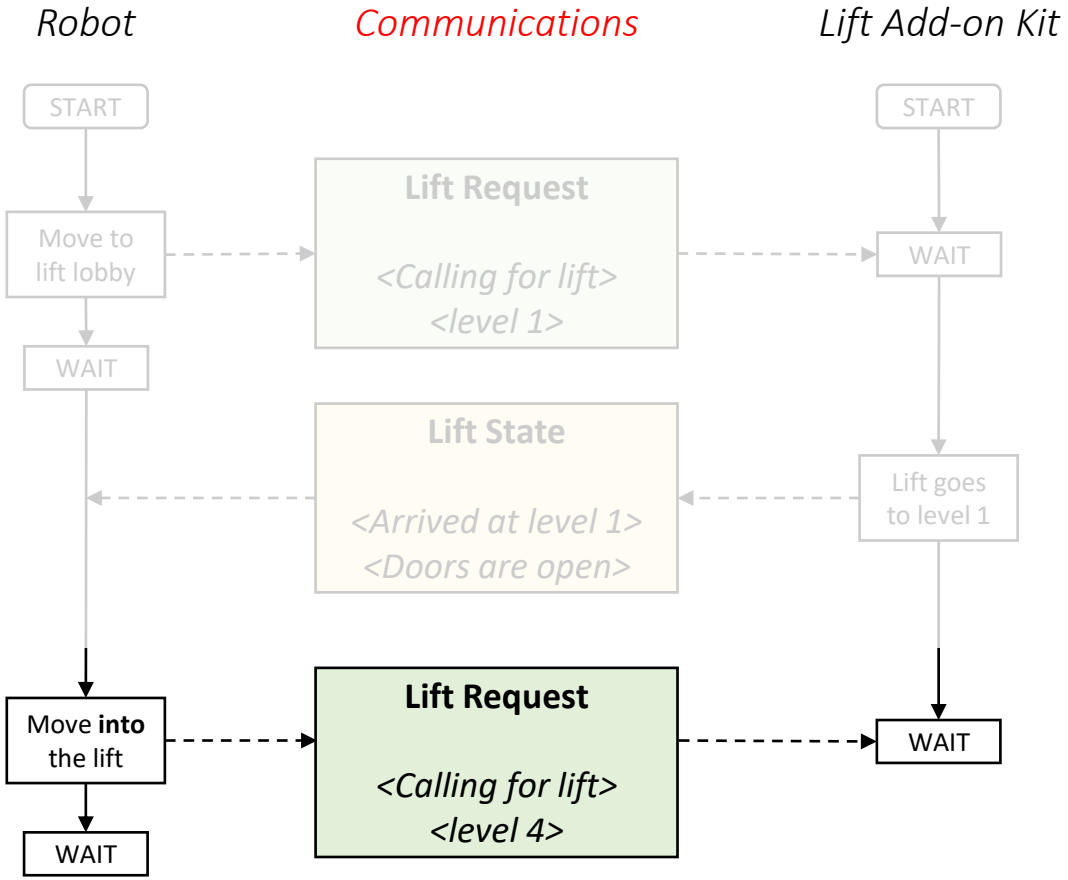
Workflow Overview



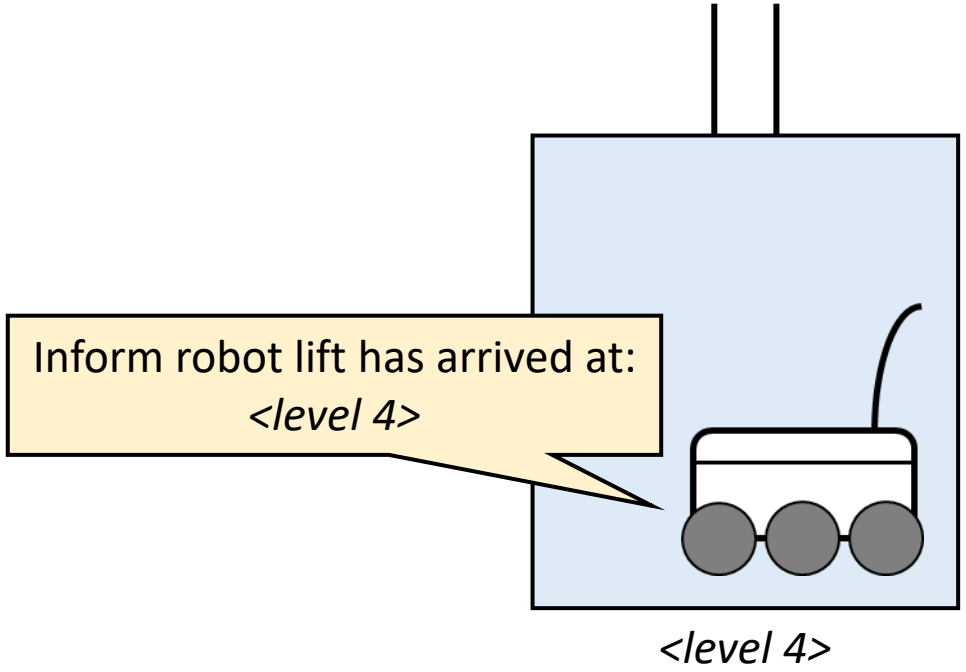


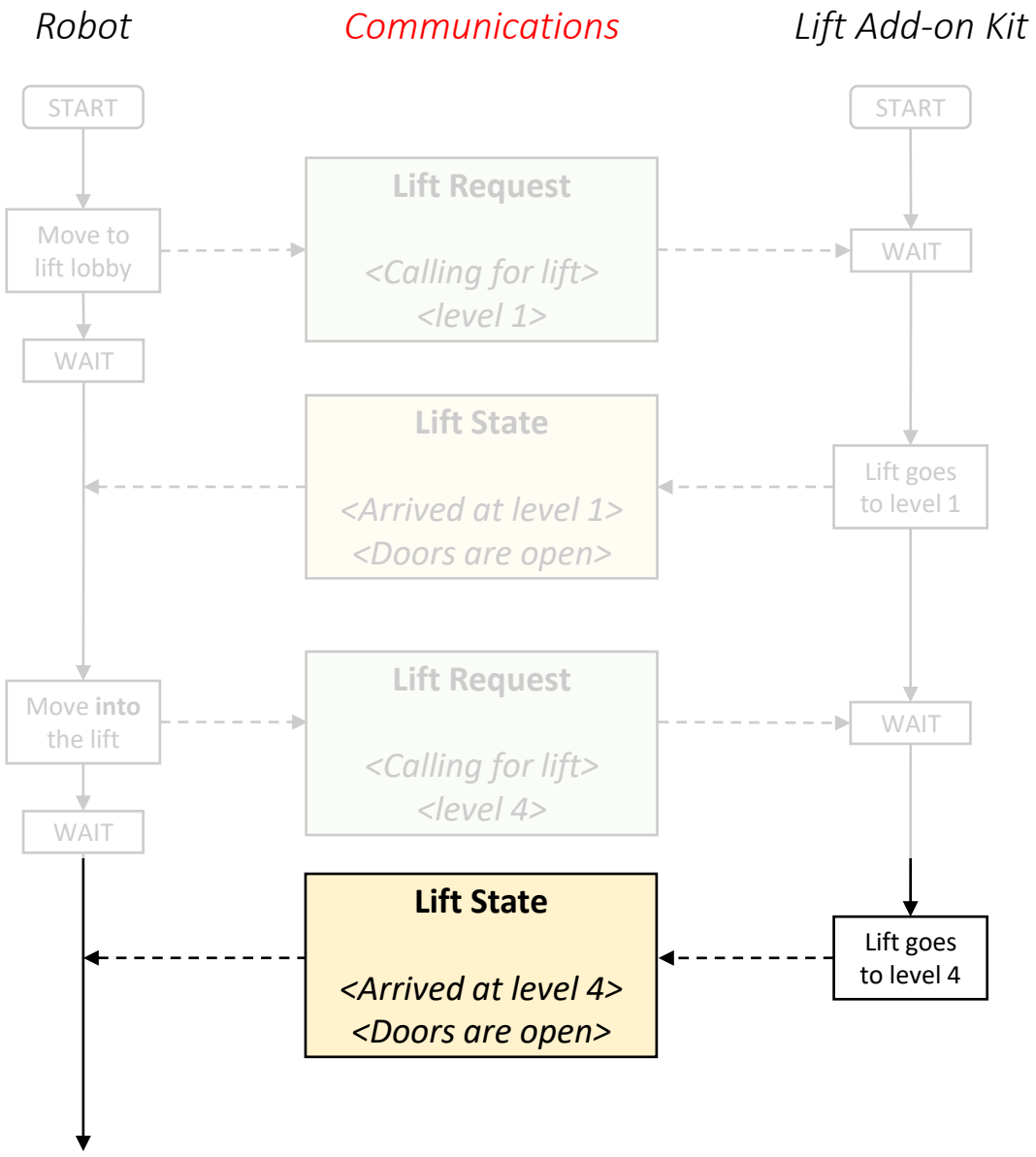
Workflow Overview



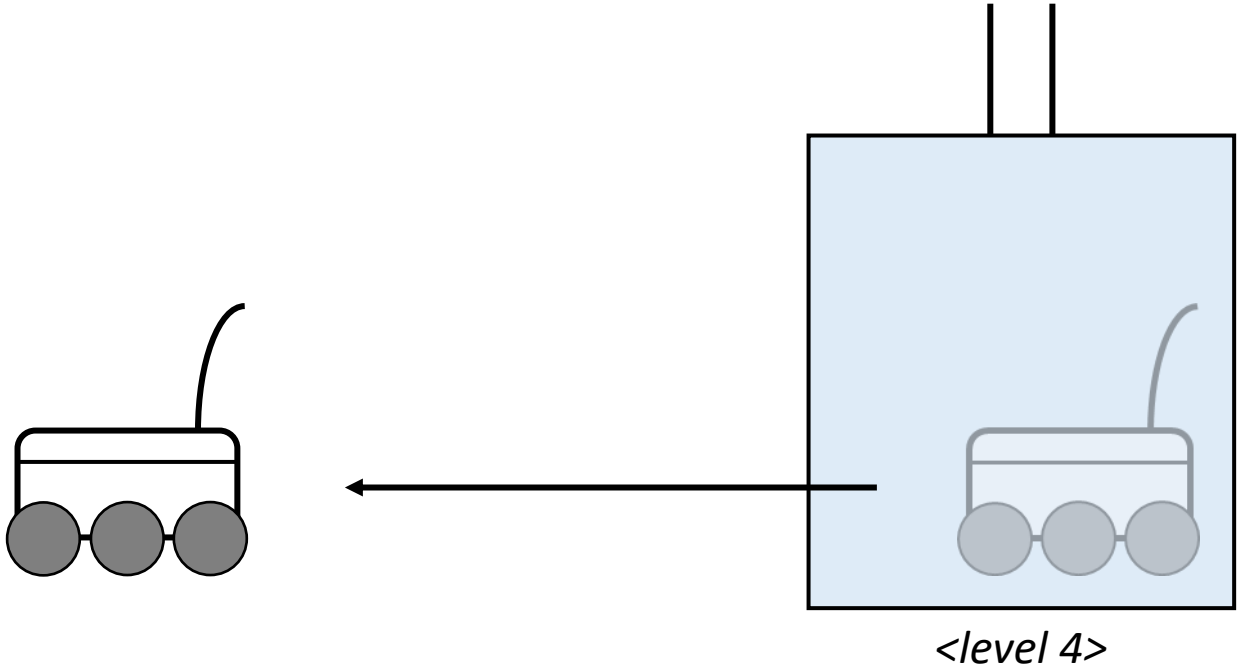


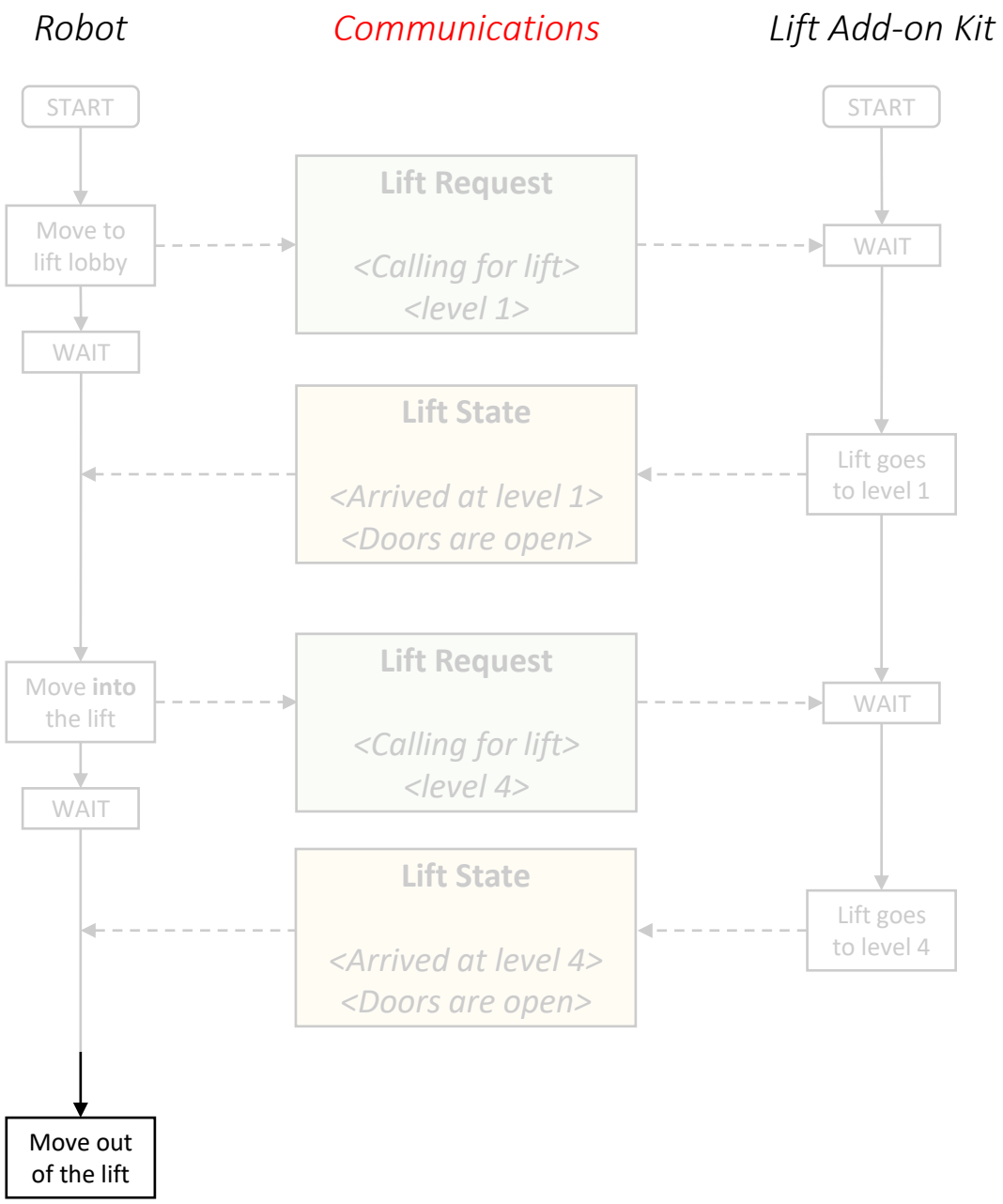
Workflow Overview



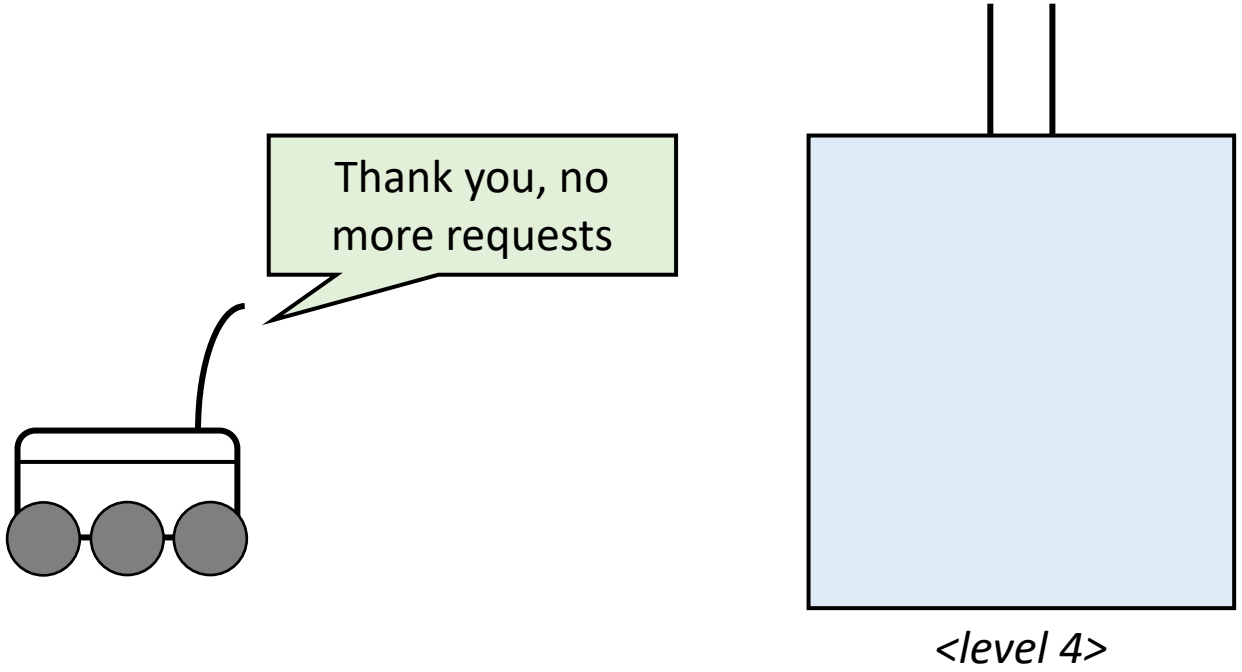


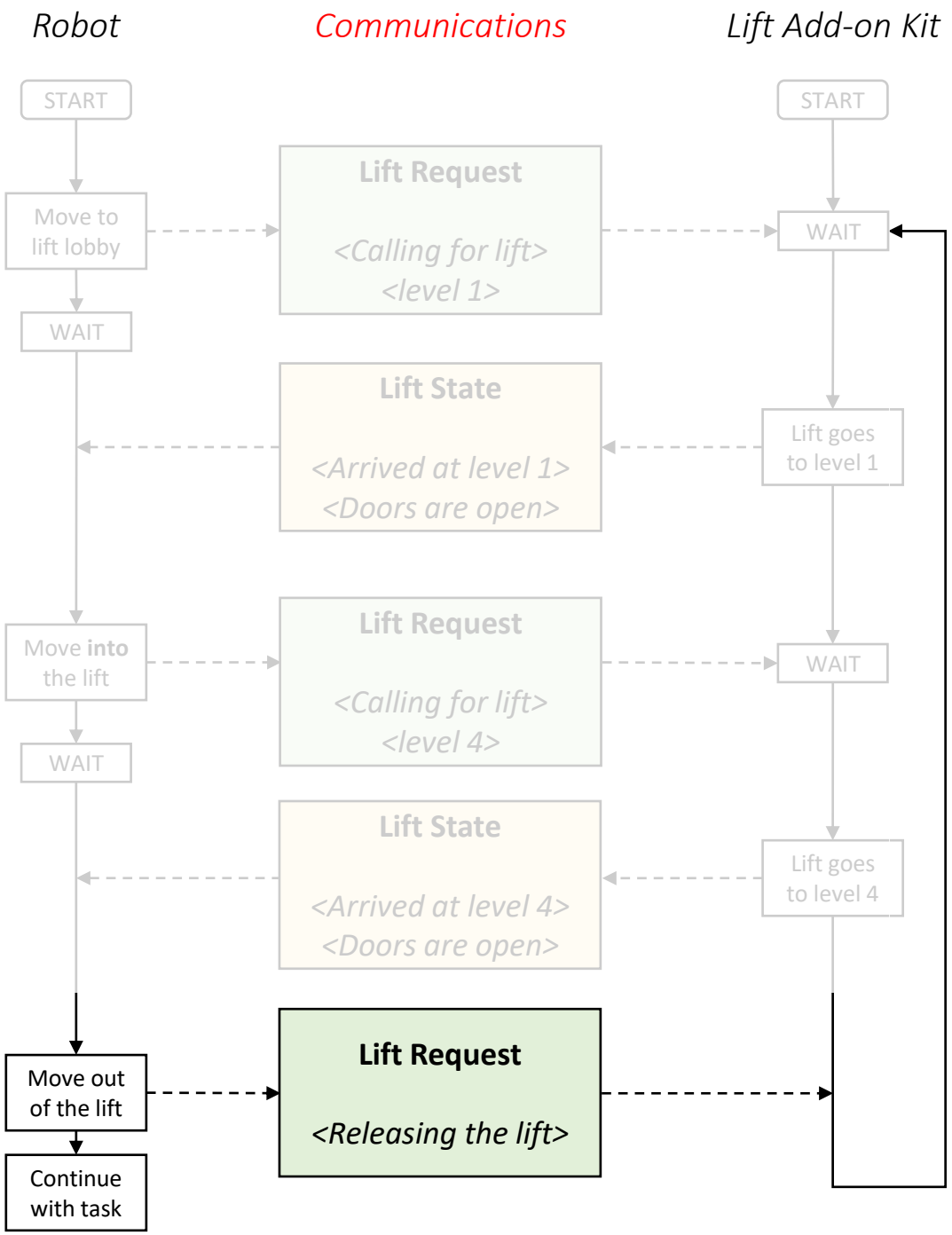
Workflow Overview



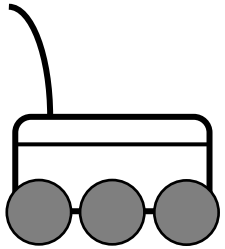


Workflow Overview





Requirements

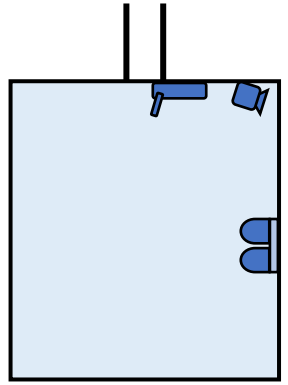


Robot needs to <u>know</u>:

- *What is it's current floor*
- *What is it's destination floor*

Robot needs to be <u>able to</u>:

- *Communicate 'Lift Request' to elevator*
- *Listen to 'Lift State' from elevator*
- *Navigate in & out of elevator*



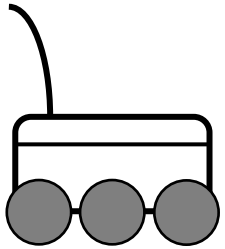
Elevator add-on kit needs to <u>know</u>:

- *Current floor elevator is on*
- *Door is open or closed*

Elevator add-on kit needs to be <u>able to</u>:

- *Communicate 'Lift State' to robot*
- *Listen to 'Lift Request' from robot*
- *Press button for requested floor*
- *Hold elevator door open*

Requirements



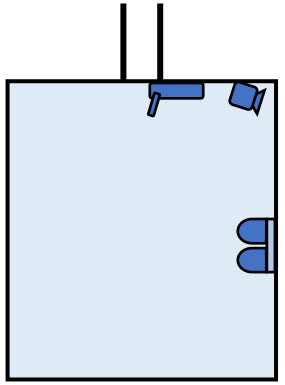
Robot needs to <u>know</u>:

- *What is it's current floor*
- *What is it's destination floor*

Robot needs to be <u>able to</u>:

Simulated for this project

- *Communicate 'Lift Request' to elevator*
- *Listen to 'Lift State' from elevator*
- *Navigate in & out of elevator*



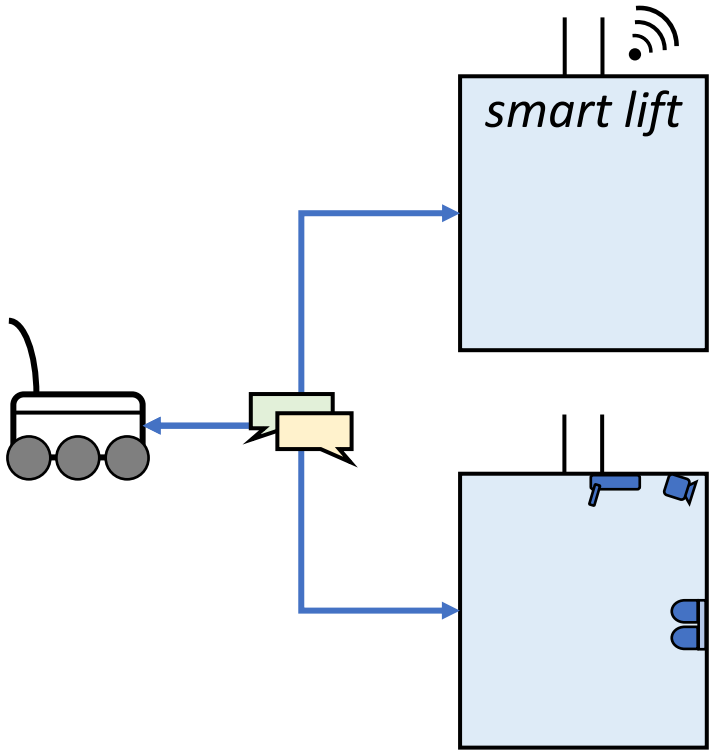
Elevator add-on kit needs to <u>know</u>:

- *Current floor elevator is on*
- *Door is open or closed*

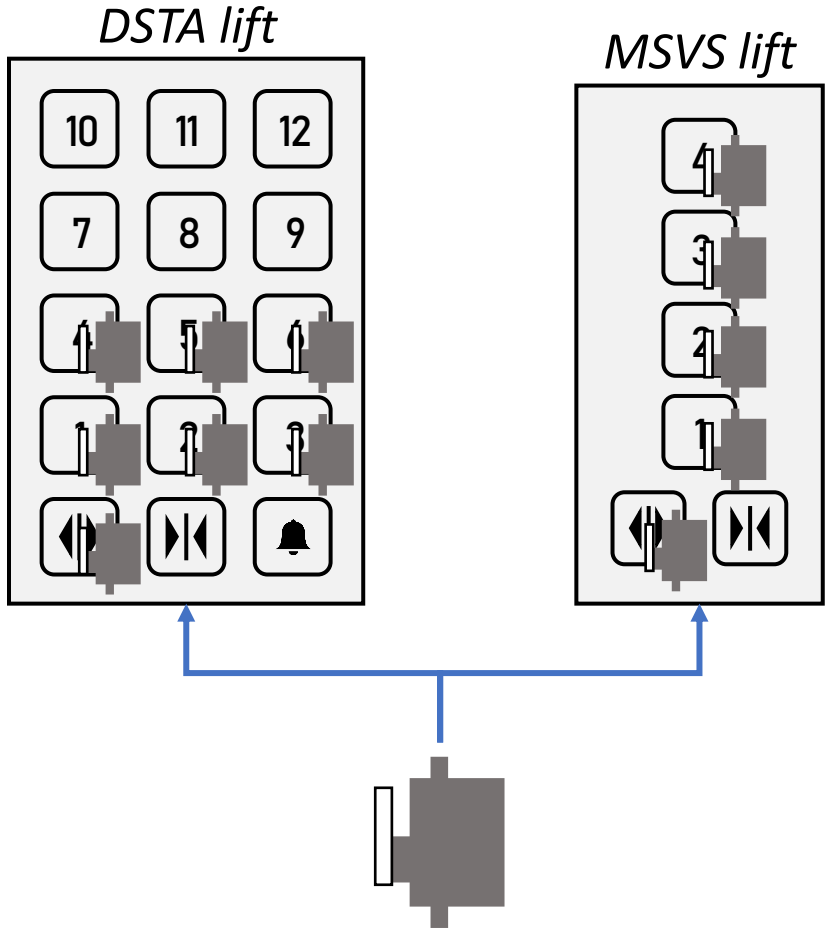
Elevator add-on kit needs to be <u>able to</u>:

- *Communicate 'Lift State' to robot*
- *Listen to 'Lift Request' from robot*
- *Press button for requested floor*
- *Hold elevator door open*

Considerations



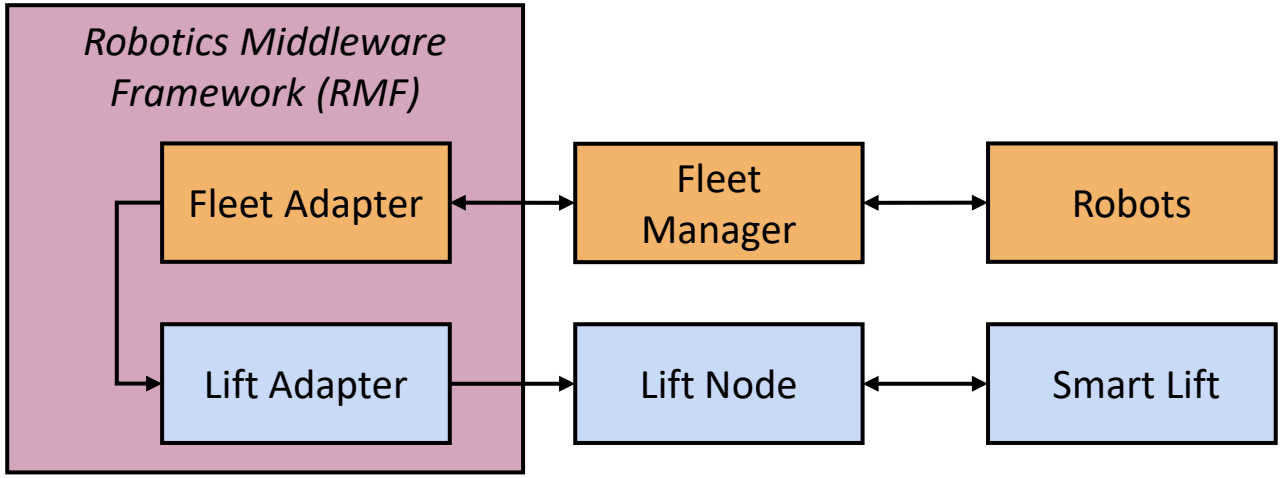
Software:
Mimic a smart lift's operation from robot's POV



Hardware:
Modular, for scalability and ease of customization

Considerations (Software)

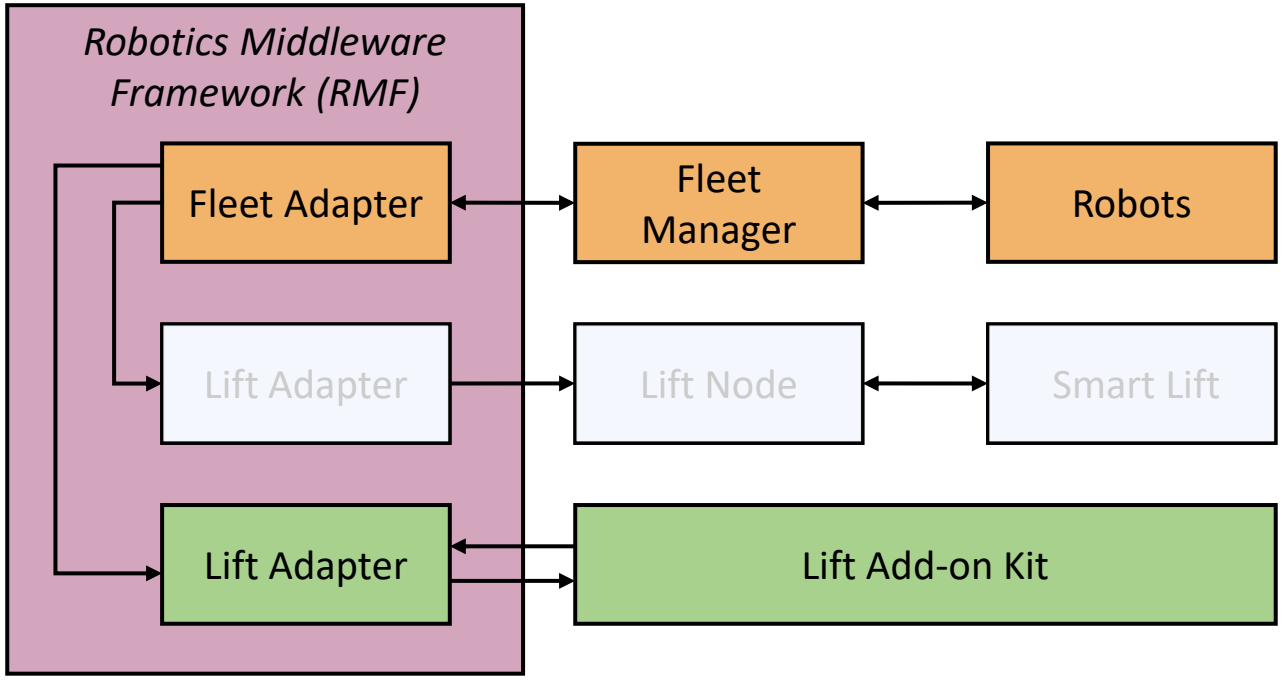
Mimic a smart lift's operation from robot's POV



Current standardized middleware framework for robot-lift operations using RMF

Considerations (Software)

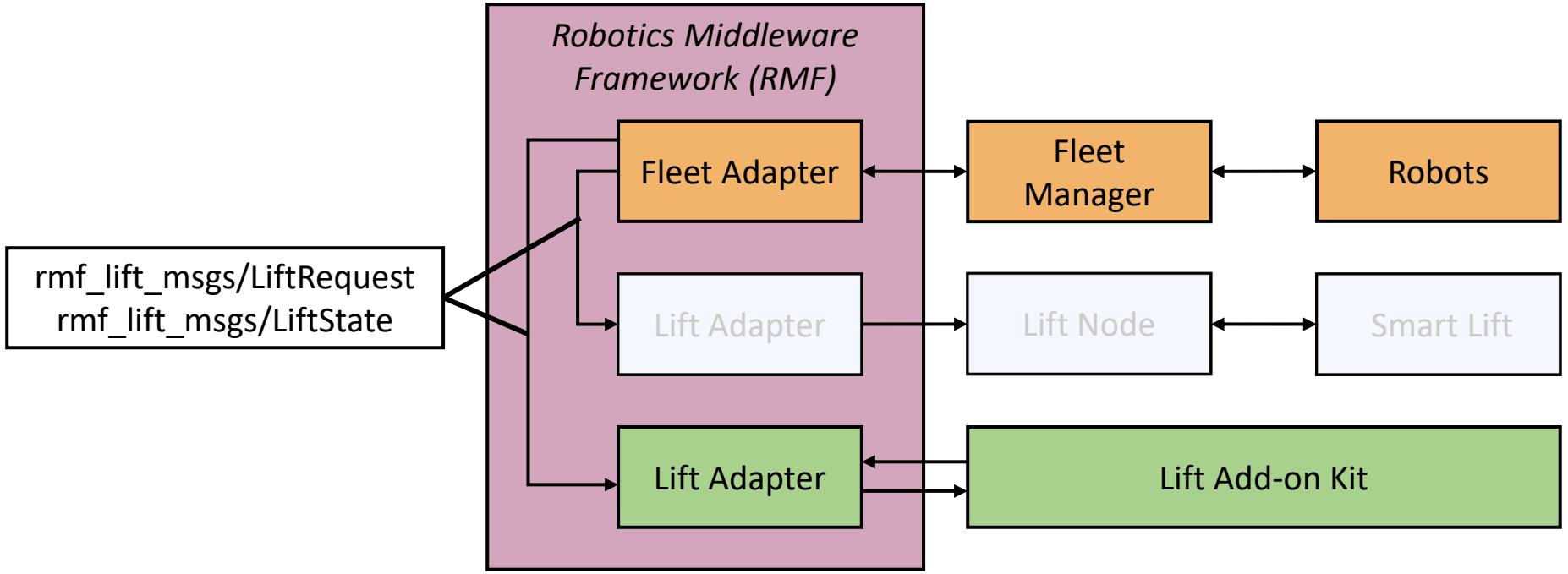
Mimic a smart lift's operation from robot's POV



Our custom add-on kit aims to seamlessly integrate into the standardized RMF framework, mimicking a smart lift

Considerations (Software)

Mimic a smart lift's operation from robot's POV

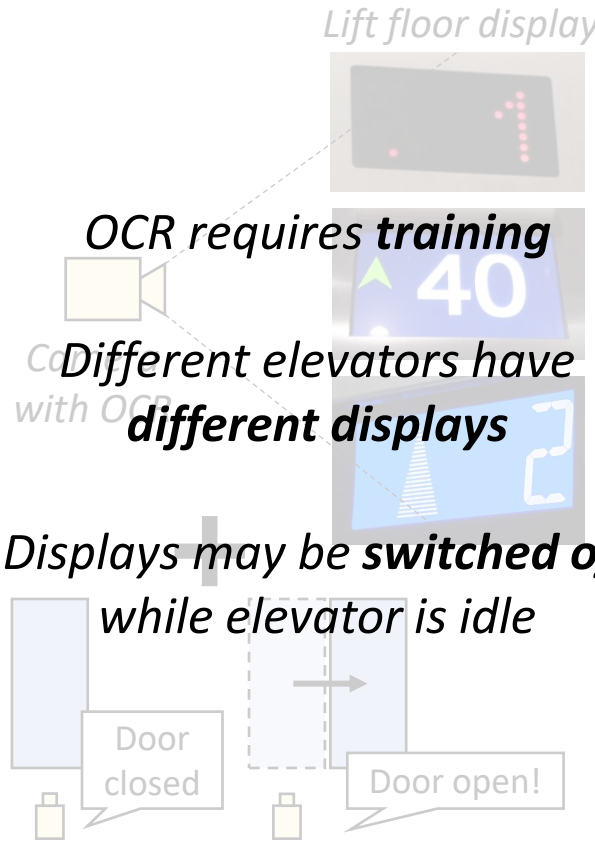


*Our custom add-on kit will communicate via **standard RMF messages***

Considerations (Hardware)

Modular, for scalability and ease of customization

What floor is the elevator on? Is the elevator door open or closed?

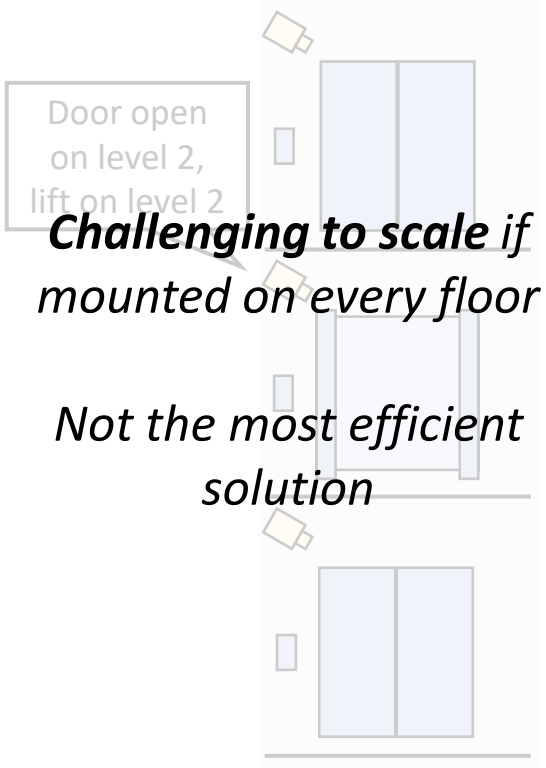


*OCR requires **training***

*Different elevators have **different displays***

*Displays may be **switched off** while elevator is idle*

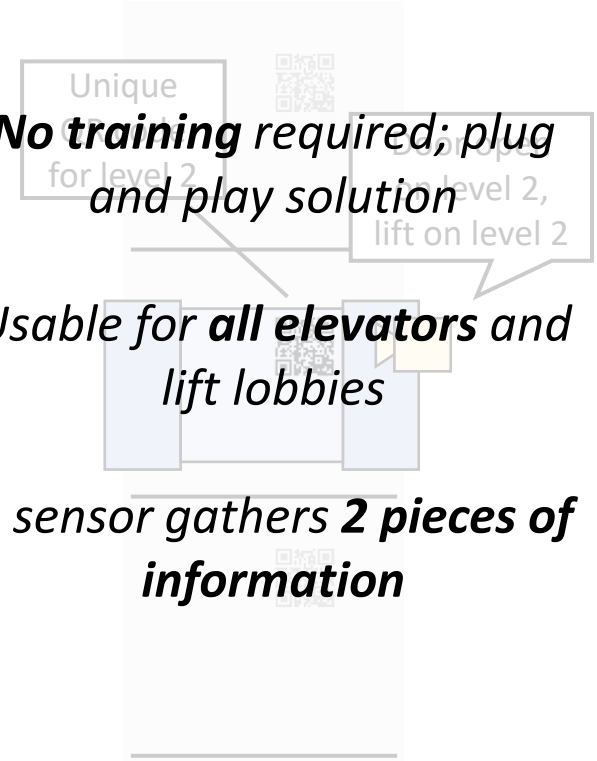
Camera + proximity sensor <u>inside</u> a single elevator



***Challenging to scale** if mounted on every floor*

Not the most efficient solution

Proximity sensor <u>outside</u> every lift door



***No training** required; plug and play solution*

*Usable for **all elevators** and lift lobbies*

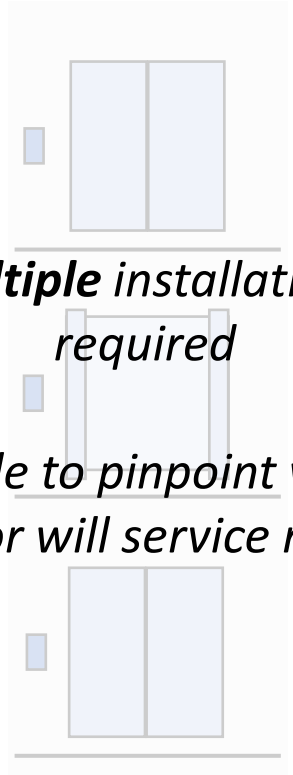
*1 sensor gathers **2 pieces of information***

Unique QR code on each floor with camera <u>inside</u> elevator

Considerations (Hardware)

Modular, for scalability and ease of customization

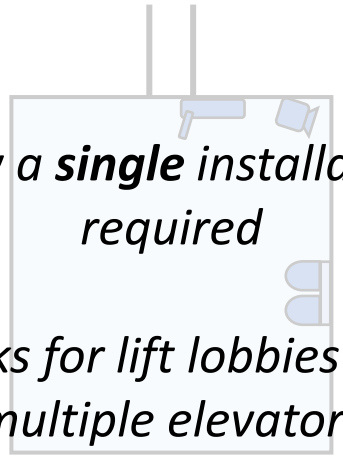
Should the elevator be controlled inside or outside?



***Multiple** installations required*

Unable to pinpoint which elevator will service request

*Actuators on up-down buttons **outside** elevator at every floor*



*Only a **single** installation required*

Works for lift lobbies with multiple elevators

*All actuators **inside** a single elevator*

Considerations (Hardware)

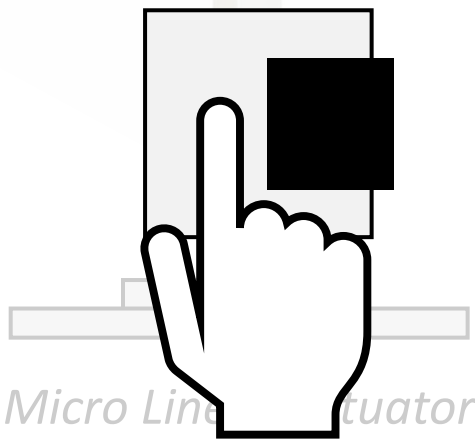
Modular, for scalability and ease of customization

How can the elevator buttons be pressed and held (without obstructing human use)?

***High** cost ~$100+*

***High** vertical profile*

*Likely to **obstruct** human use of buttons with its height*



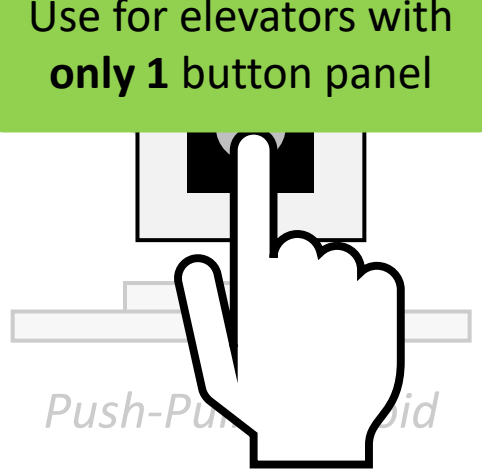
Micro Linear Actuator

***Low** cost ~$20*

***High** vertical profile*

*Allows human to **manually push down** on plunger*

Use for elevators with **only 1** button panel



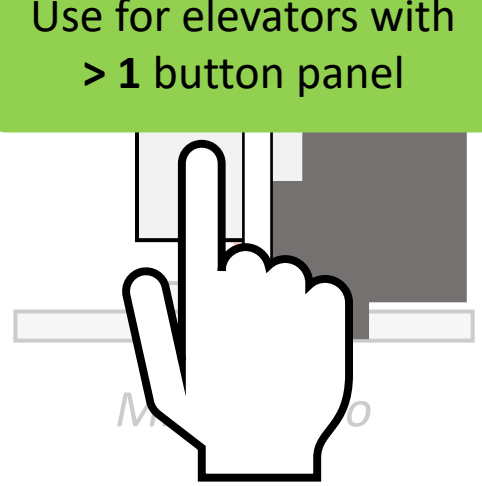
Push-Pull Solenoid

***Low** cost ~$10*

***Low** vertical profile*

*Likely to **obstruct** human use of buttons with its width*

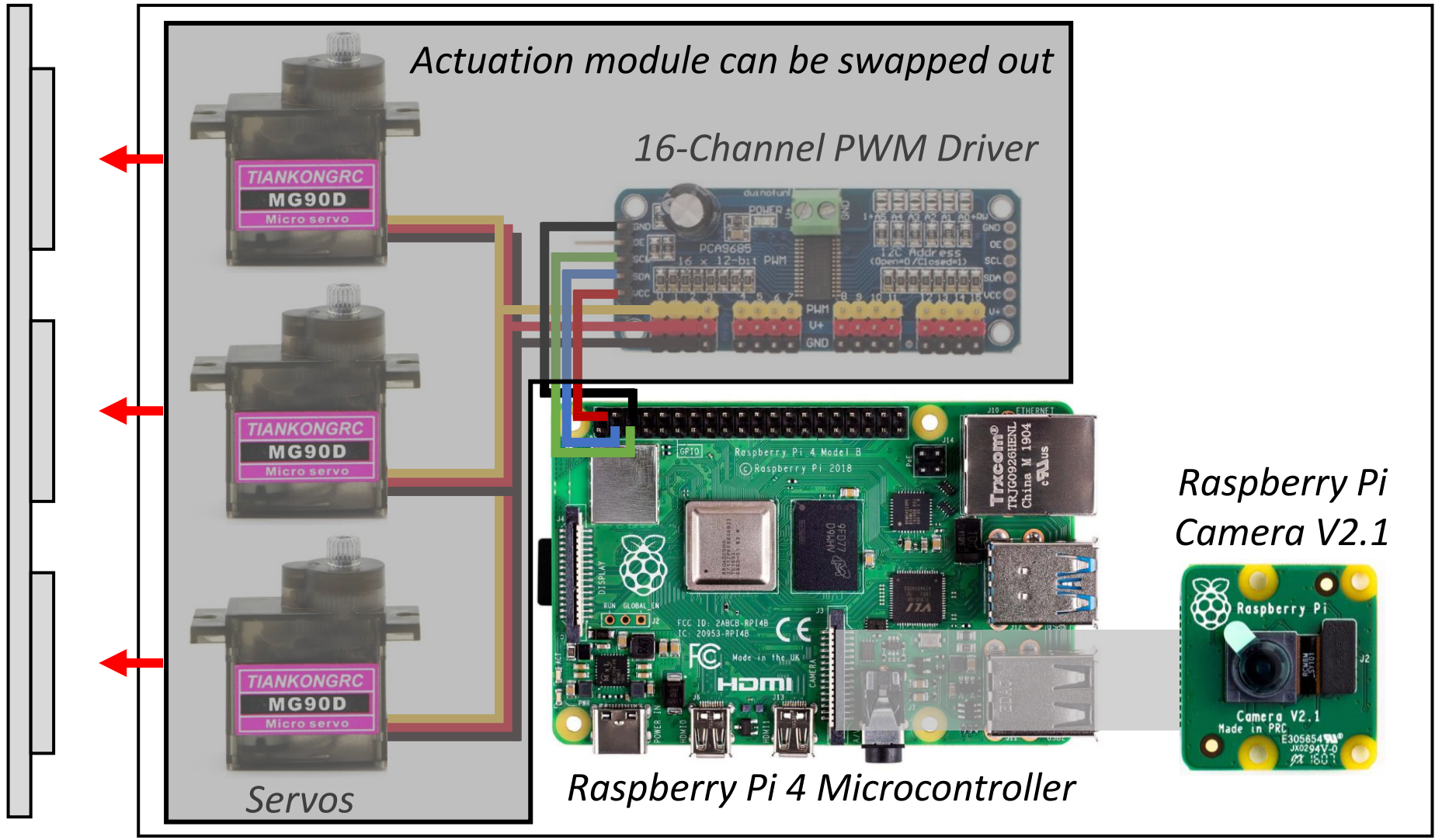
Use for elevators with **> 1** button panel



Micro Servo

Our Solution

Add-on Kit inside elevator



4

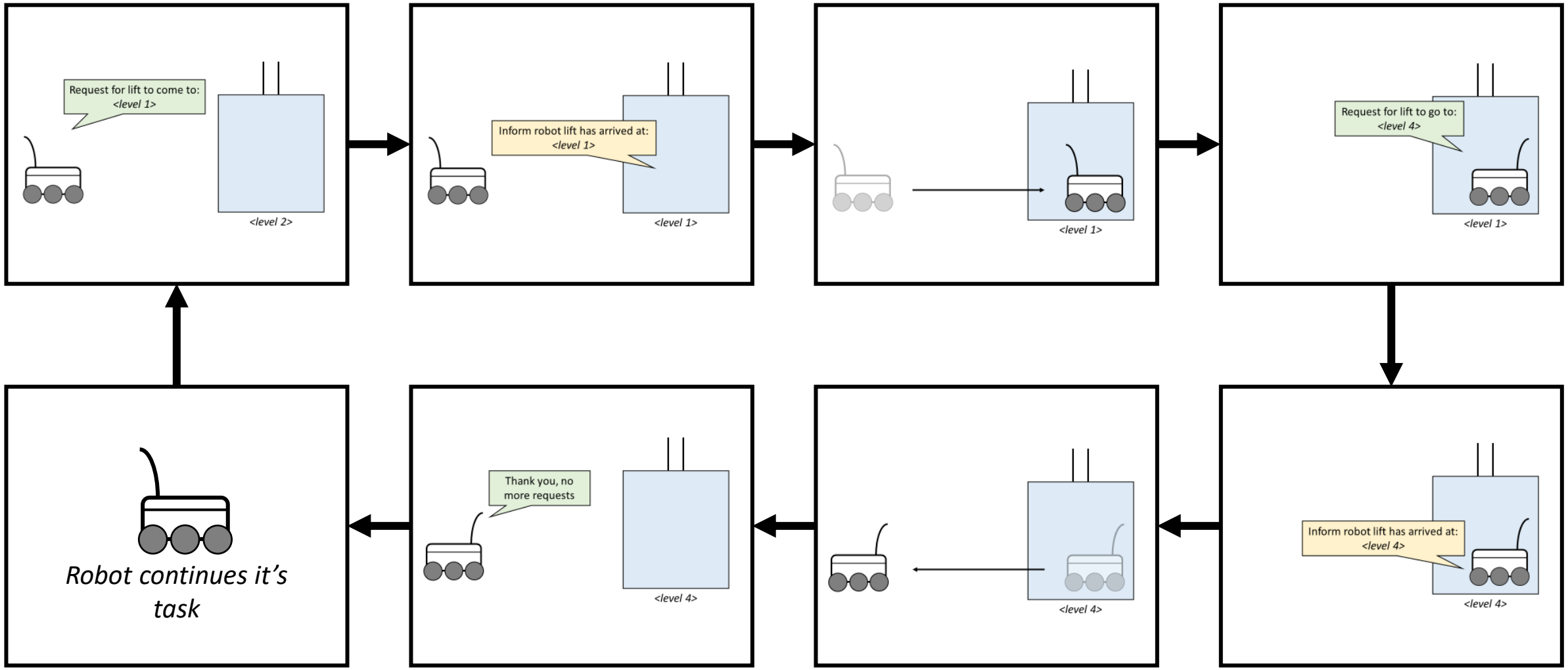
3

2

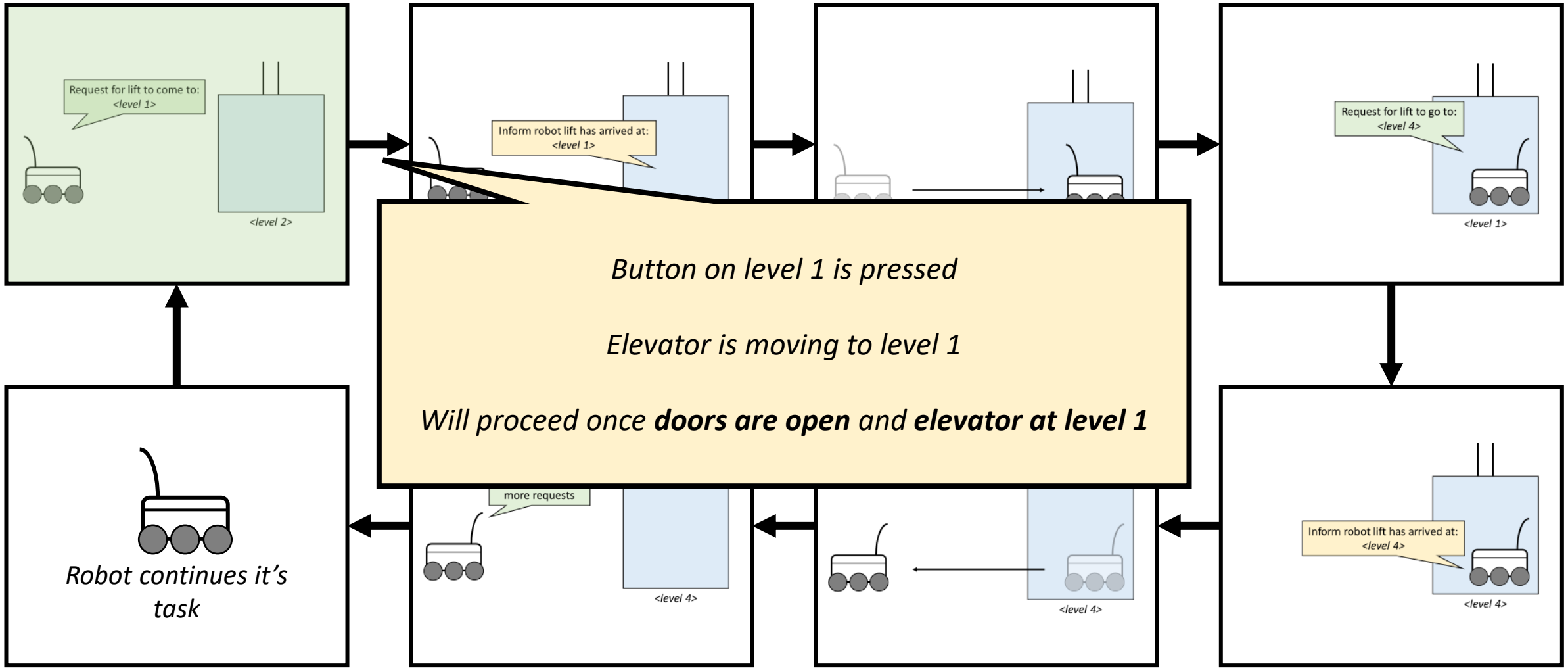
1

Custom QR codes

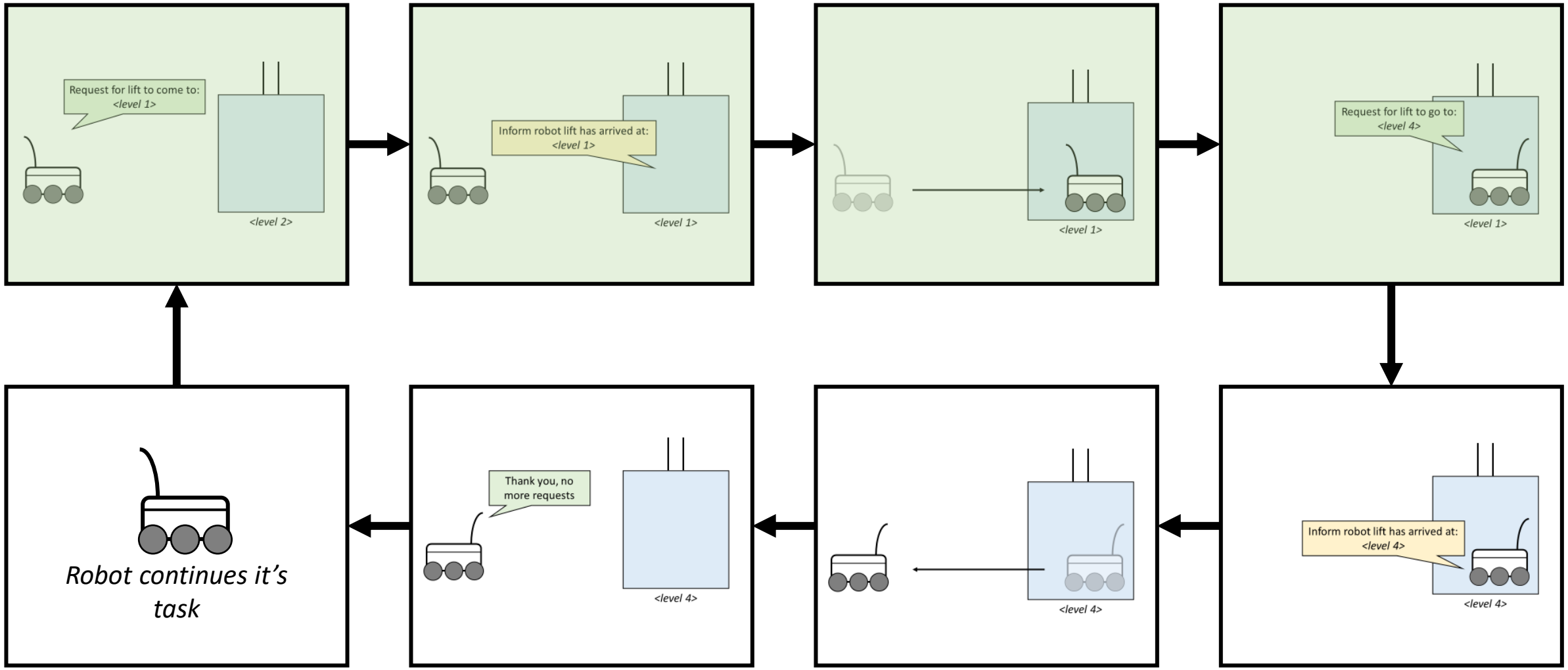
Demonstration



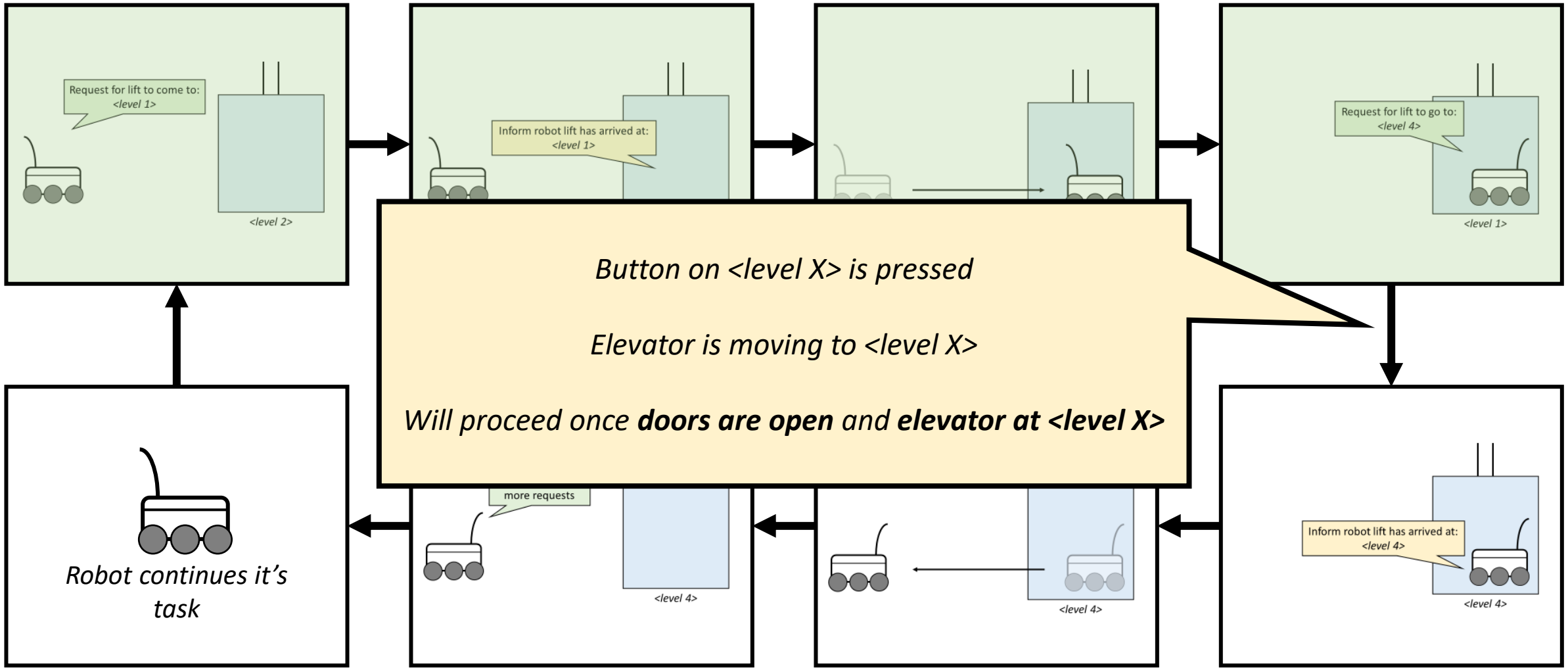
Demonstration



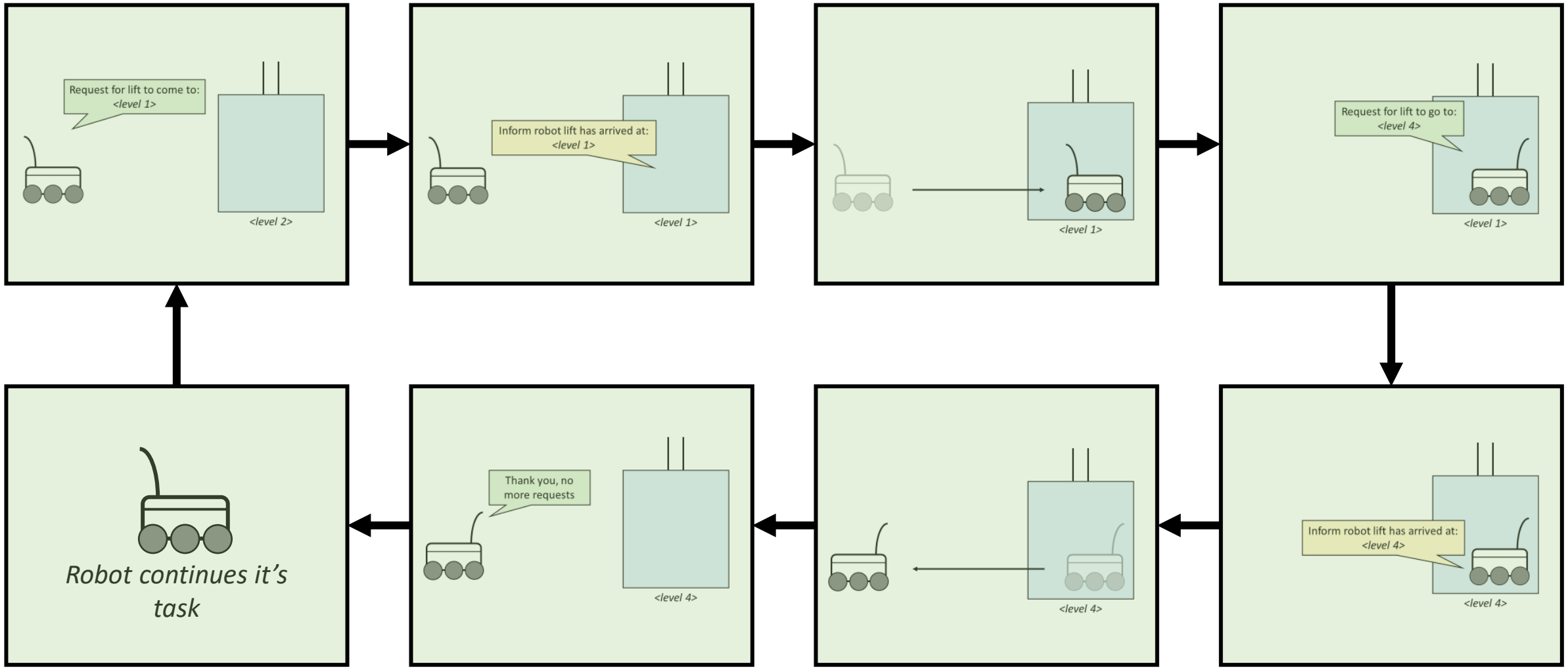
Demonstration



Demonstration



Demonstration



Camera detects QR code when door opens, and knows that elevator is at level 12

Camera detects QR code when door opens, and knows that elevator is at level 1

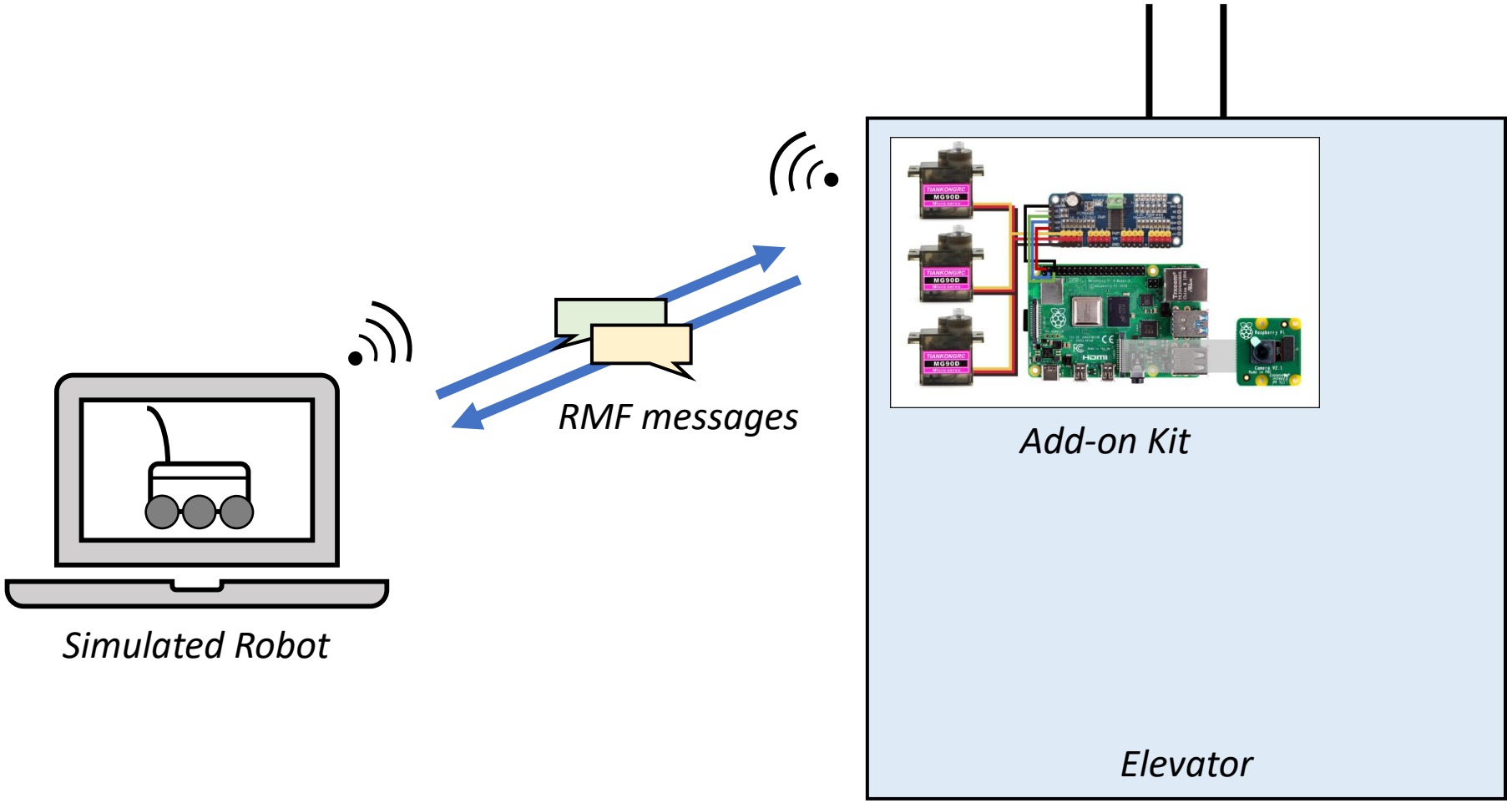
Robot requests to go to level 1

Door open button is held until robot is in the elevator

Robot at level 12 requests for lift

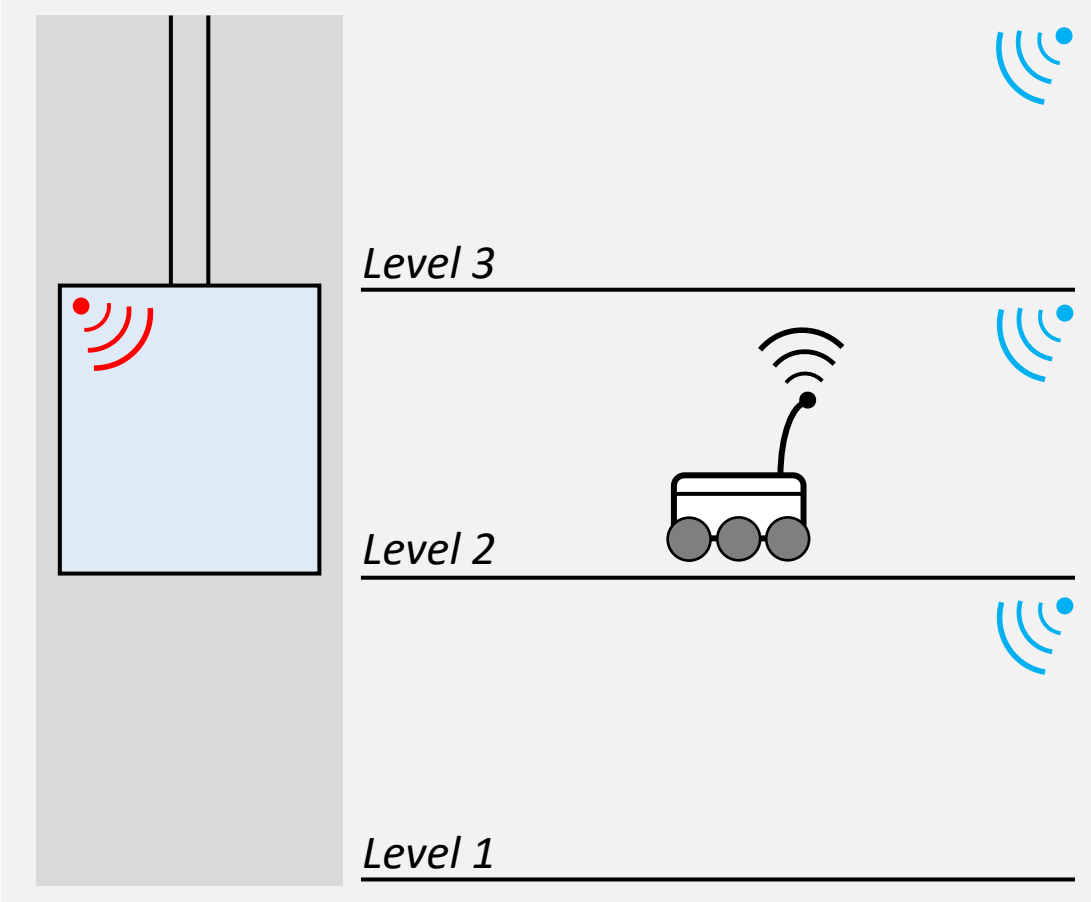
Human enters at level 10 and presses button for level 2

Summary



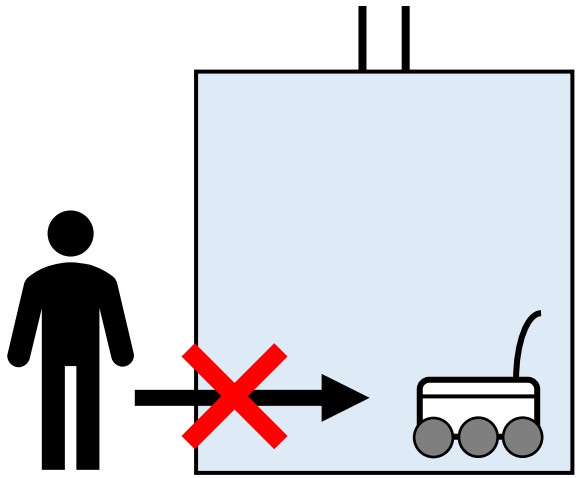
A prototype add-on kit was developed and demonstrated to work with a simulated robot over RMF

Next Steps

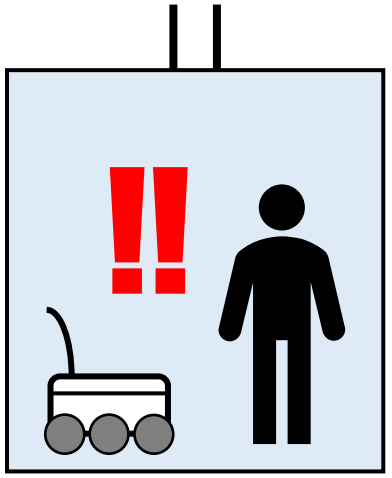


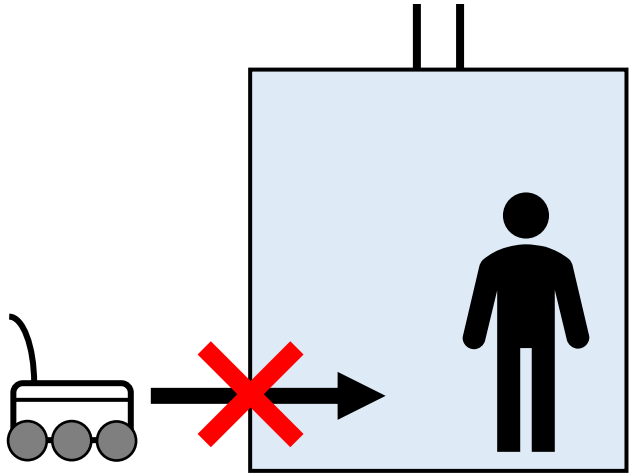
*1. Tackle **imperfect network connectivity** between within the elevator and outside the elevator*

Next Steps



Educate humans not to enter elevator if already occupied by robot

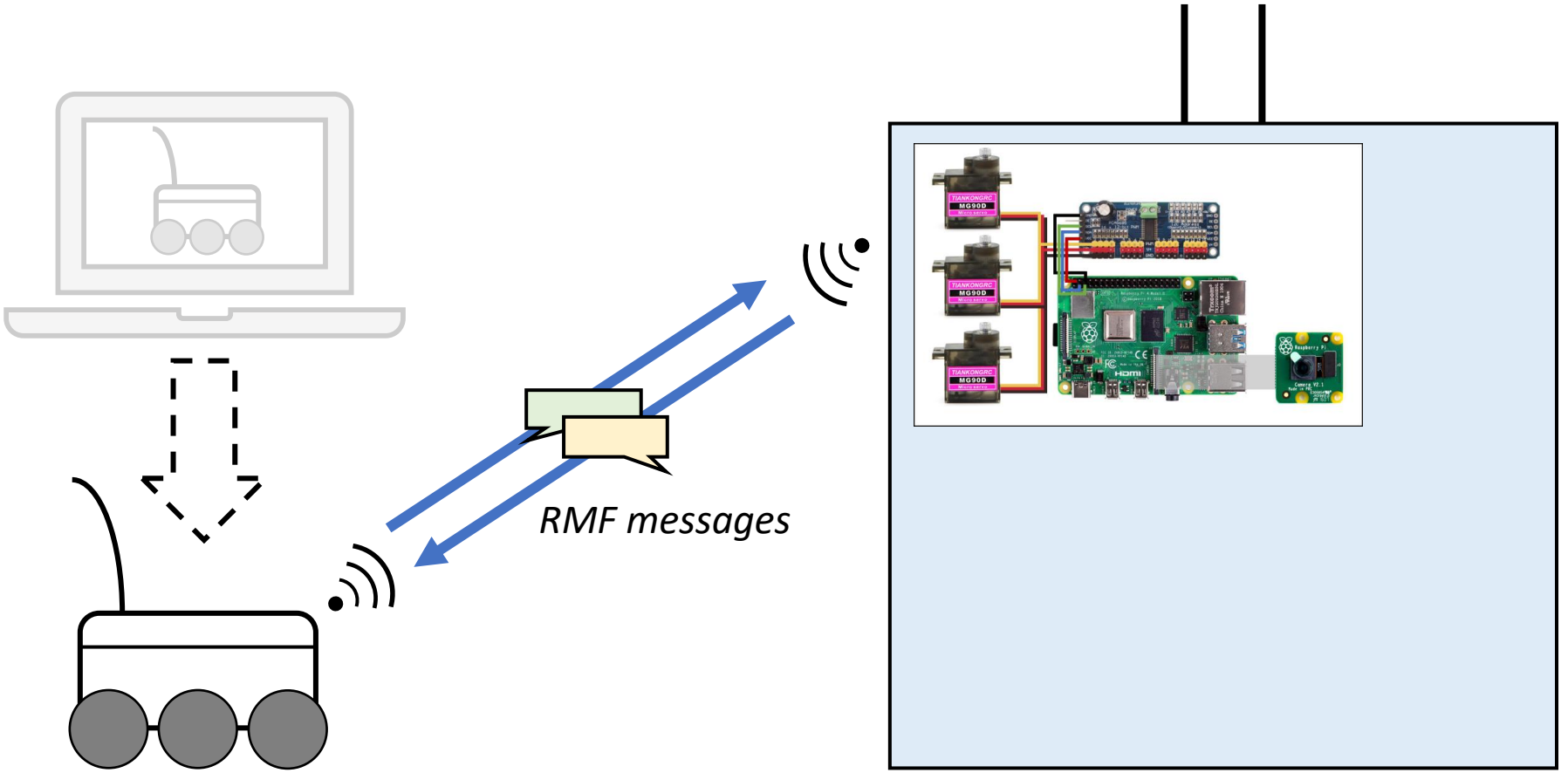




Additional sensors to detect human presence in elevator and instruct robot not to enter

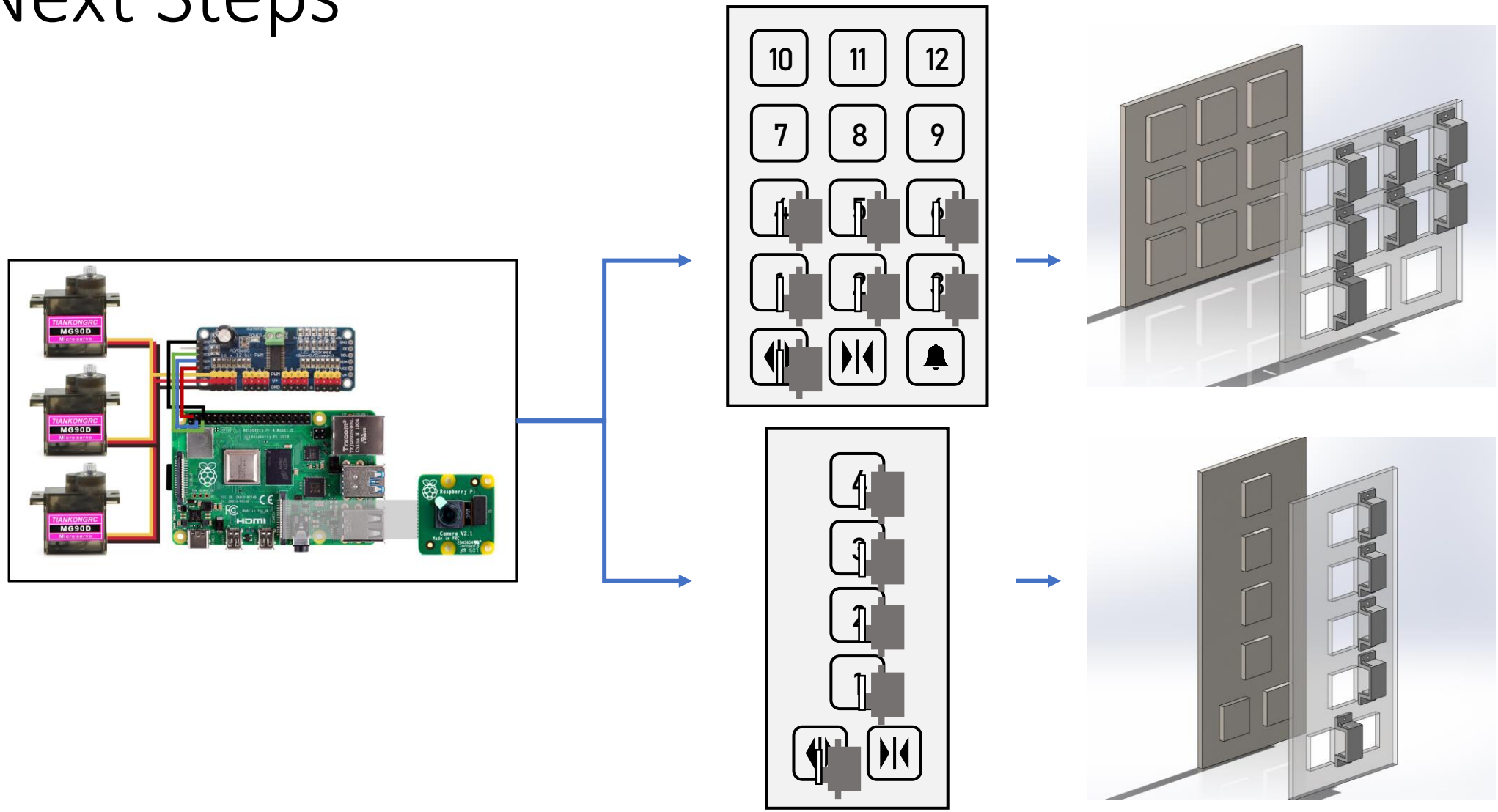
*2. Define **human-robot coexistence** rules and algorithms*

Next Steps



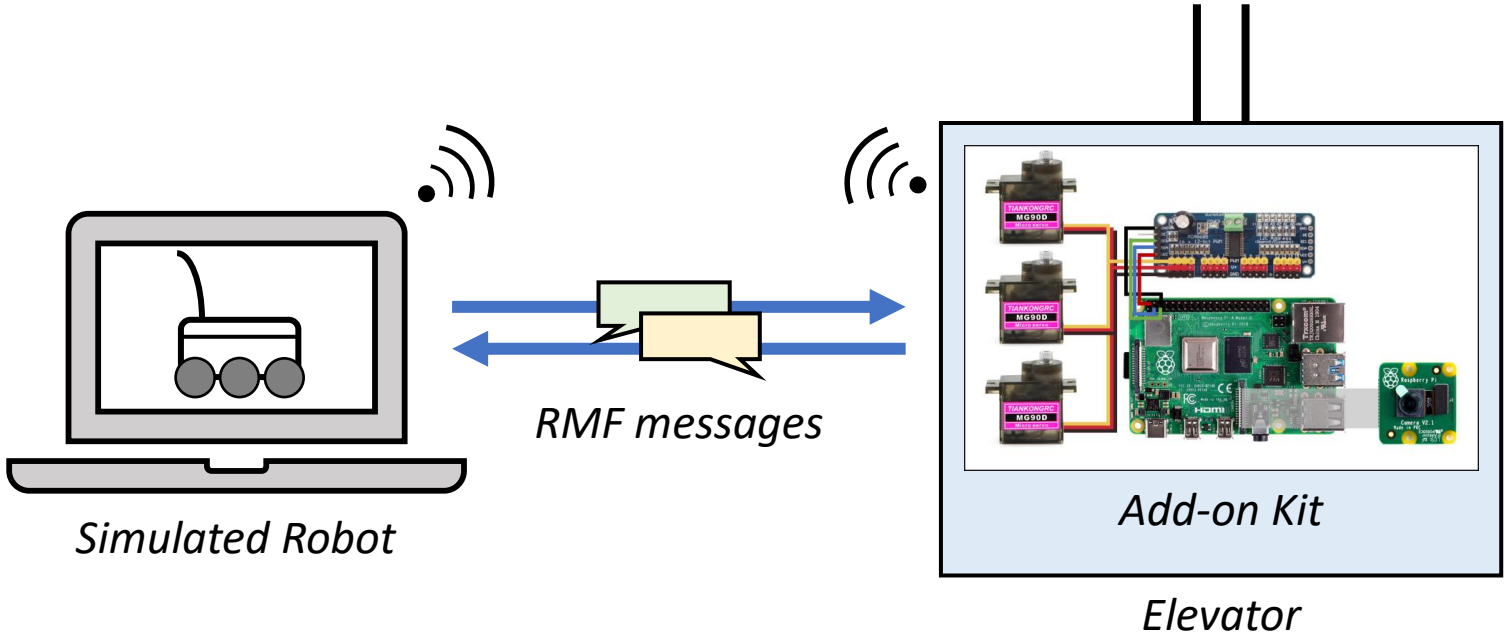
*3. Integrate and test with **real robots** and elevators*

Next Steps

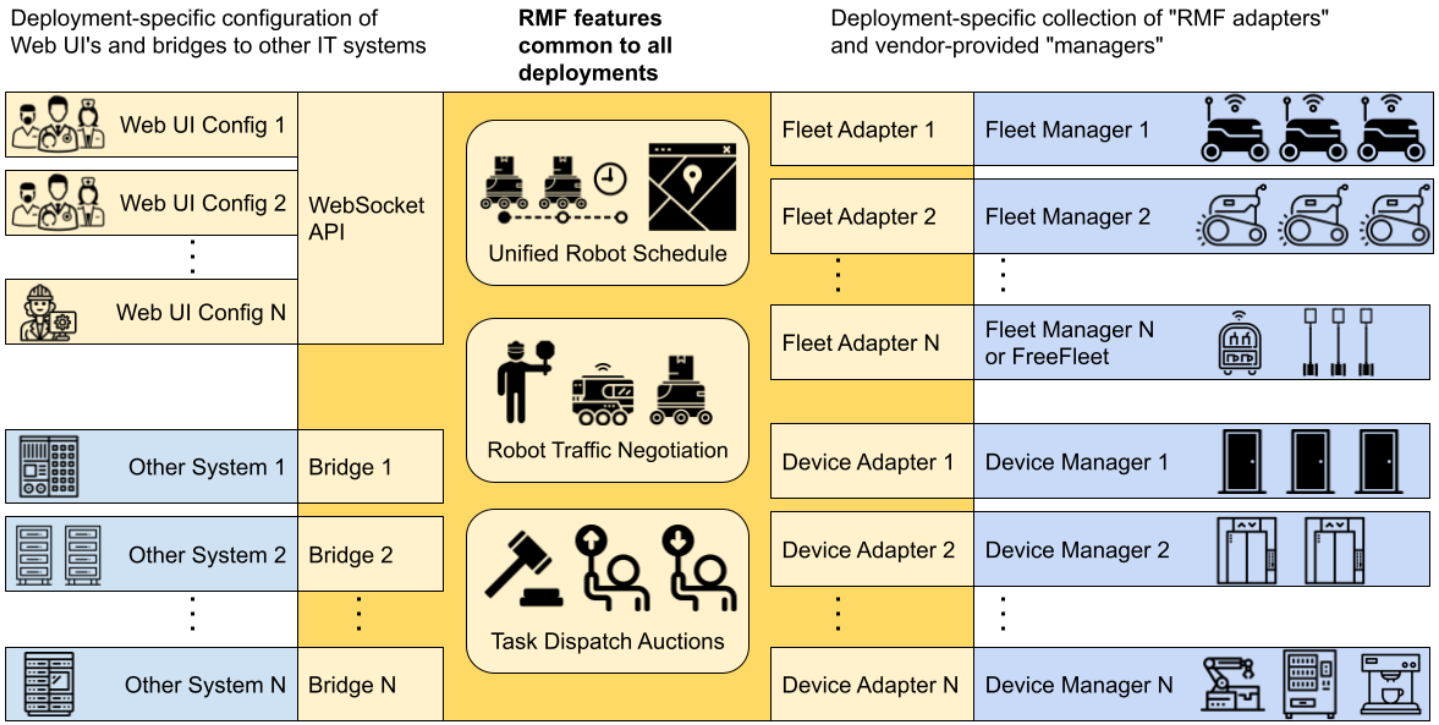


*4. Commercialize prototype to **market-ready product***

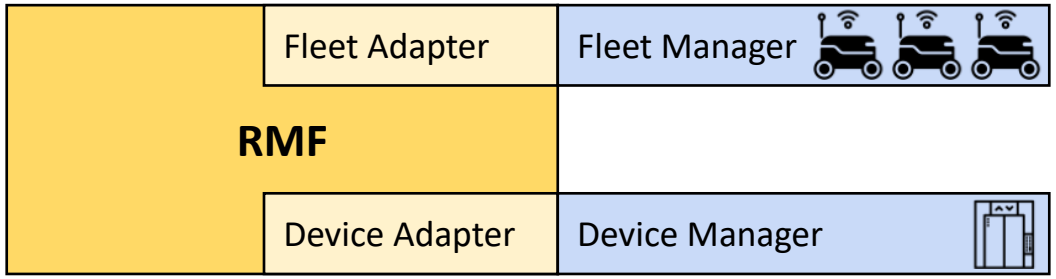
Thank You



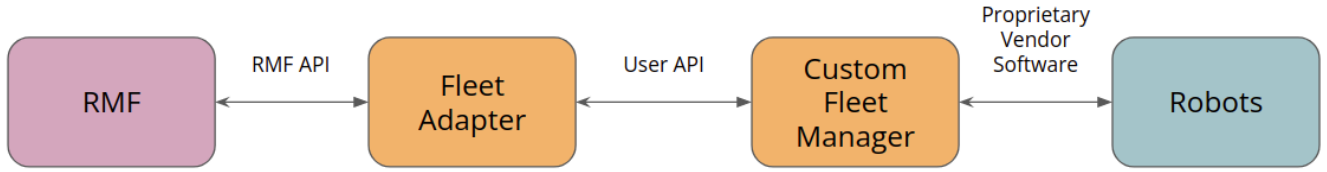
Summary of Robotics Middleware Framework (RMF)



RMF contextualized to MSVS / DSTA elevator operation

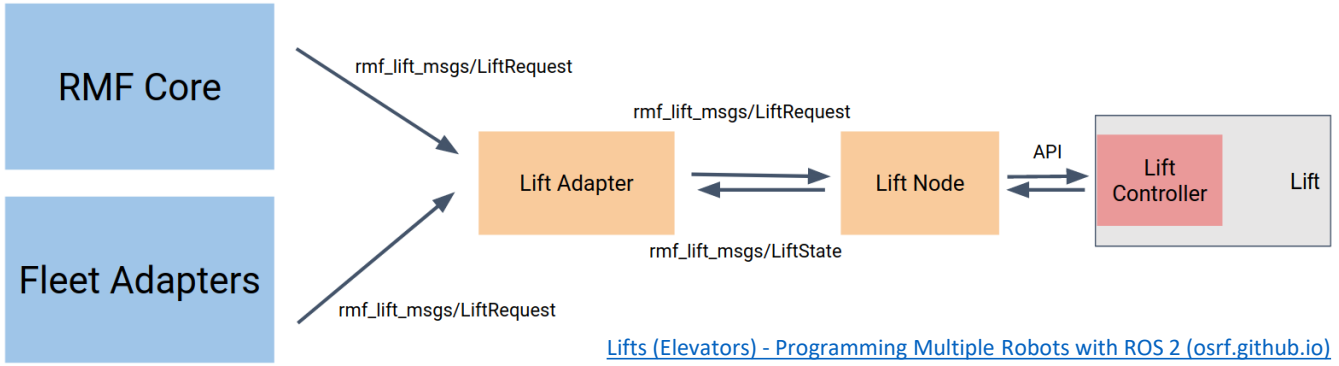


Fleet Adapter template



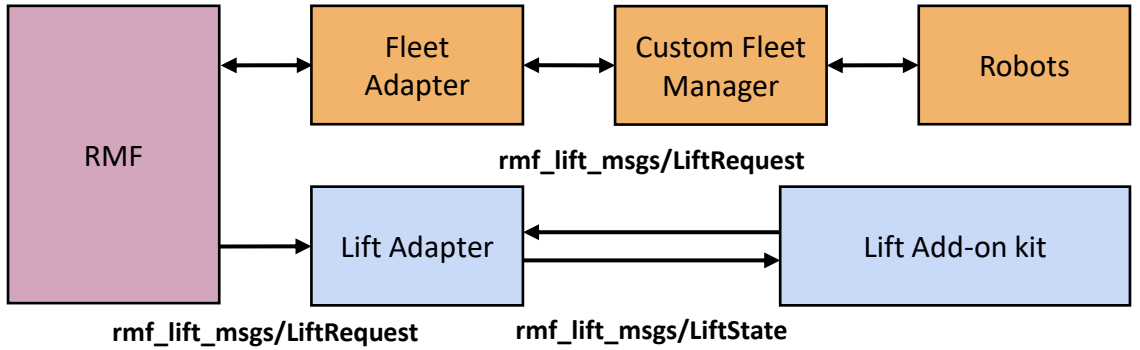
Mobile Robot Fleets - Programming Multiple Robots with ROS 2 (osrf.github.io)

Elevator integration with RMF

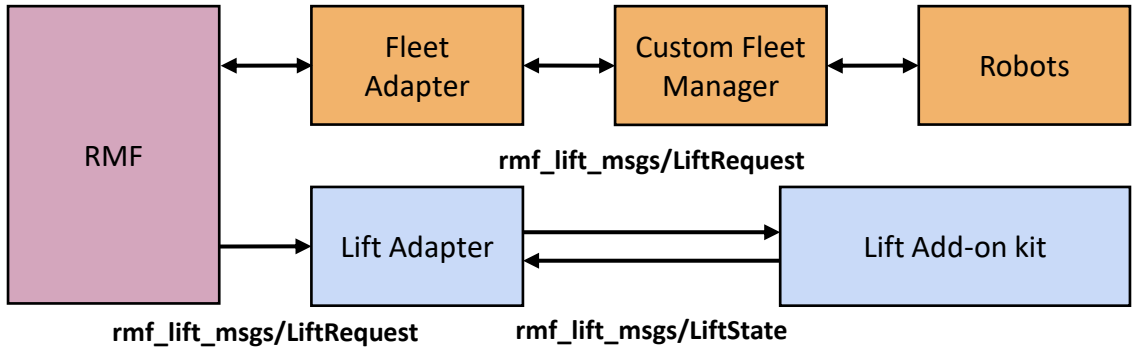


Lifts (Elevators) - Programming Multiple Robots with ROS 2 (osrf.github.io)

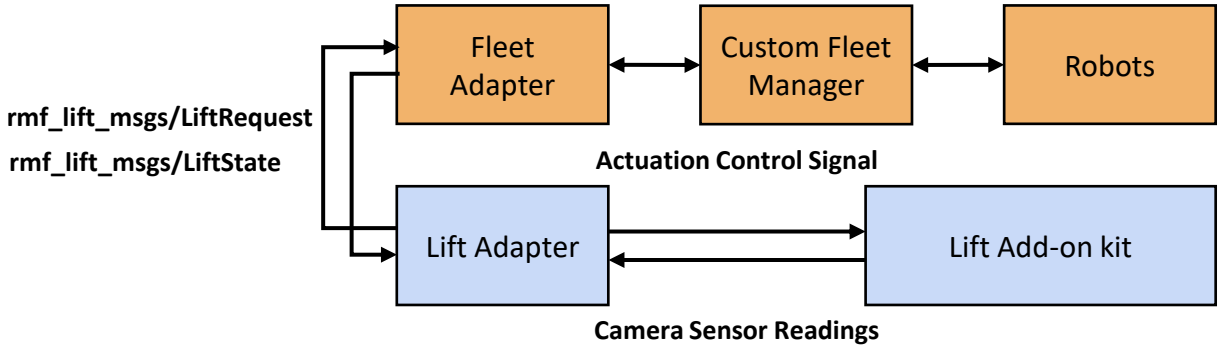
RMF contextualized to MSVS / DSTA elevator operation



RMF contextualized to MSVS / DSTA elevator operation



Current Implementation



Lift Request

```
1   string lift_name
2   builtin_interfaces/Time request_time
3
4   # session_id should be unique at least between different requesters.
5   # For example, session_id could be the requester's node name.
6   string session_id
7
8   # AGV mode means that the doors are always open when the lift is stopped
9   # Human mode means that LiftDoorRequest messages must be used to open/close
10  # the doors explicitly, since they may "time out" and close automatically.
11  uint8 request_type
12  uint8 REQUEST_END_SESSION=0
13  uint8 REQUEST_AGV_MODE=1
14  uint8 REQUEST_HUMAN_MODE=2
15
16  # The destination_floor must be one of the values returned in a LiftState.
17  string destination_floor
18
19  # Explicit door requests are necessary in "human" mode to open/close doors.
20  # Door requests are not necessary in "AGV" mode, when the doors are always
21  # held open when the lift cabin is stopped.
22  uint8 door_state
23  uint8 DOOR_CLOSED=0
24  uint8 DOOR_OPEN=2
```

Message Types	ROS2 Topic	Description
`rmf_lift_msgs/LiftState`	`/lift_states`	State of the lift published by the lift node
`rmf_lift_msgs/LiftRequest`	`/lift_requests`	Direct requests subscribed by the lift node and published by the lift adapter
`rmf_lift_msgs/LiftRequest`	`/adapter_lift_requests`	Requests to be sent to the lift adapter/supervisor to request safe operation of lifts

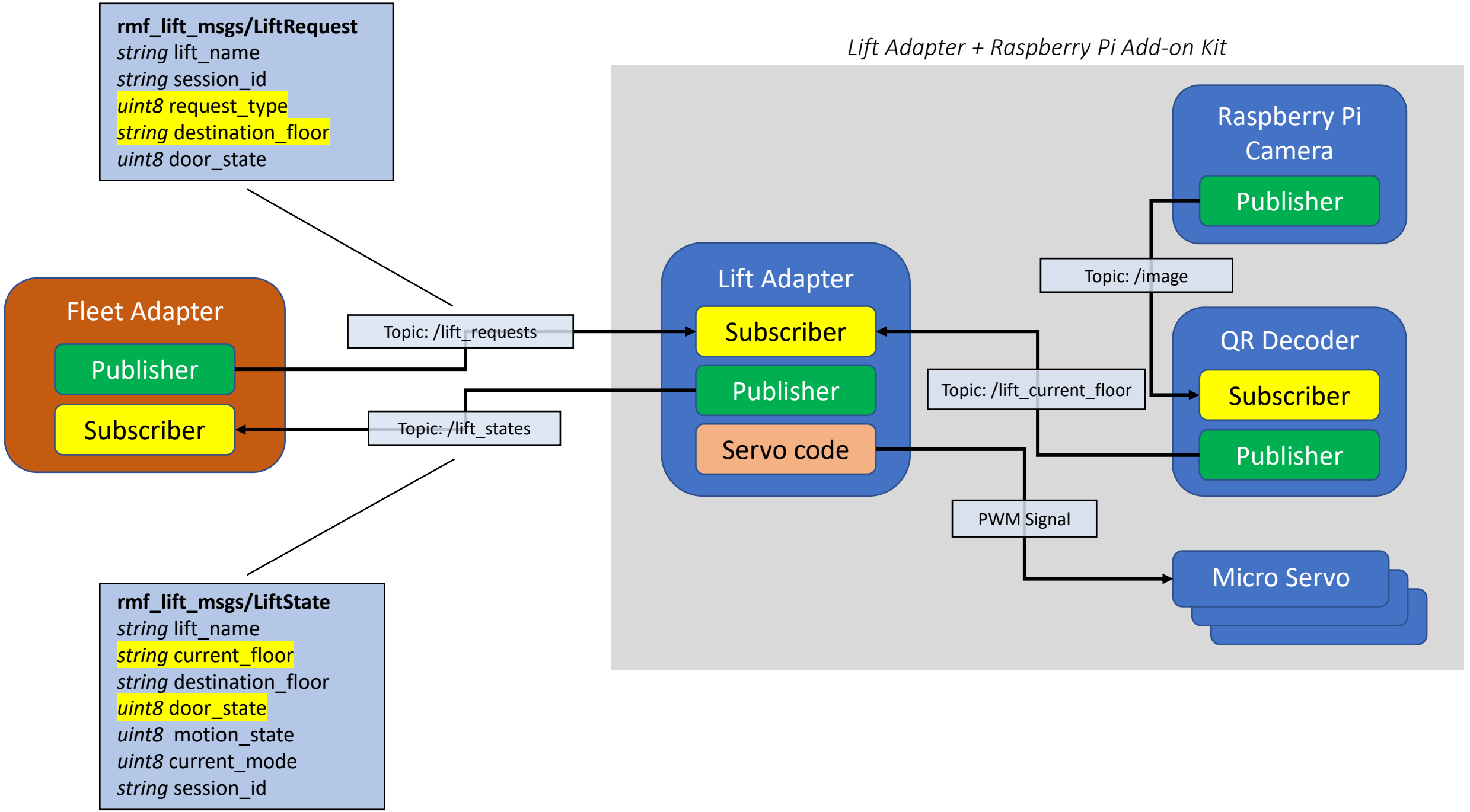
rmf_internal_msgs/rmf_lift_msgs/msg/LiftRequest.msg at main · open-rmf/rmf_internal_msgs · GitHub

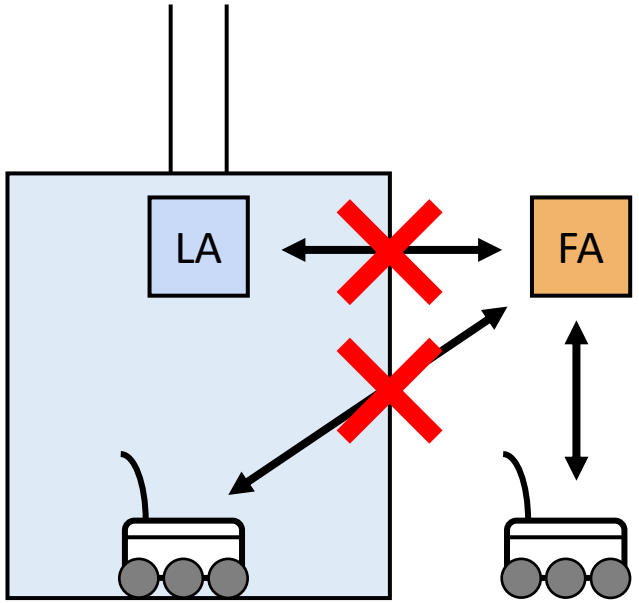
Lift State

```
# lift_time records when the information in this message was generated
builtin_interfaces/Time lift_time

string lift_name

string[] available_floors
string current_floor
string destination_floor

uint8 door_state
uint8 DOOR_CLOSED=0
uint8 DOOR_MOVING=1
uint8 DOOR_OPEN=2

uint8 motion_state
uint8 MOTION_STOPPED=0
uint8 MOTION_UP=1
uint8 MOTION_DOWN=2
uint8 MOTION_UNKNOWN=3

# We can only set human or agv mode, but we can read other modes: fire, etc.
uint8[] available_modes
uint8 current_mode
uint8 MODE_UNKNOWN=0
uint8 MODE_HUMAN=1
uint8 MODE_AGV=2
uint8 MODE_FIRE=3
uint8 MODE_OFFLINE=4
uint8 MODE_EMERGENCY=5
# we can add more "read-only" modes as we come across more of them.

# this field records the session_id that has been granted control of the lift
# until it sends a request with a request_type of REQUEST_END_SESSION
string session_id
```

Message Types	ROS2 Topic	Description
`rmf_lift_msgs/LiftState`	`/lift_states`	State of the lift published by the lift node
`rmf_lift_msgs/LiftRequest`	`/lift_requests`	Direct requests subscribed by the lift node and published by the lift adapter
`rmf_lift_msgs/LiftRequest`	`/adapter_lift_requests`	Requests to be sent to the lift adapter/supervisor to request safe operation of lifts

rmf_internal_msgs/rmf_lift_msgs/msg/LiftState.msg at main · open-rmf/rmf_internal_msgs · GitHub


rmf_lift_msgs/LiftRequest
string lift_name
string session_id
uint8 request_type
string destination_floor
uint8 door_state
Lift Adapter + Raspberry Pi Add-on Kit
Raspberry Pi Camera
Publisher
Topic: /image
Fleet Adapter
Publisher
Subscriber
Topic: /lift_requests
Lift Adapter
Subscriber
Publisher
Servo code
Topic: /lift_current_floor
QR Decoder
Subscriber
Publisher
Topic: /lift_states
PWM Signal
Micro Servo
rmf_lift_msgs/LiftState
string lift_name
string current_floor
string destination_floor
uint8 door_state
uint8 motion_state
uint8 current_mode
string session_id

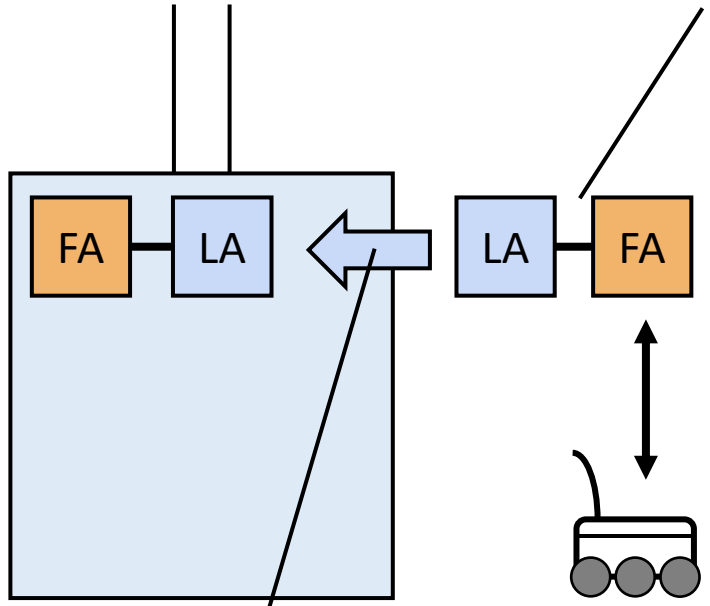


Ideal communication with perfect connectivity

*Most elevators act as a **faraday cage***

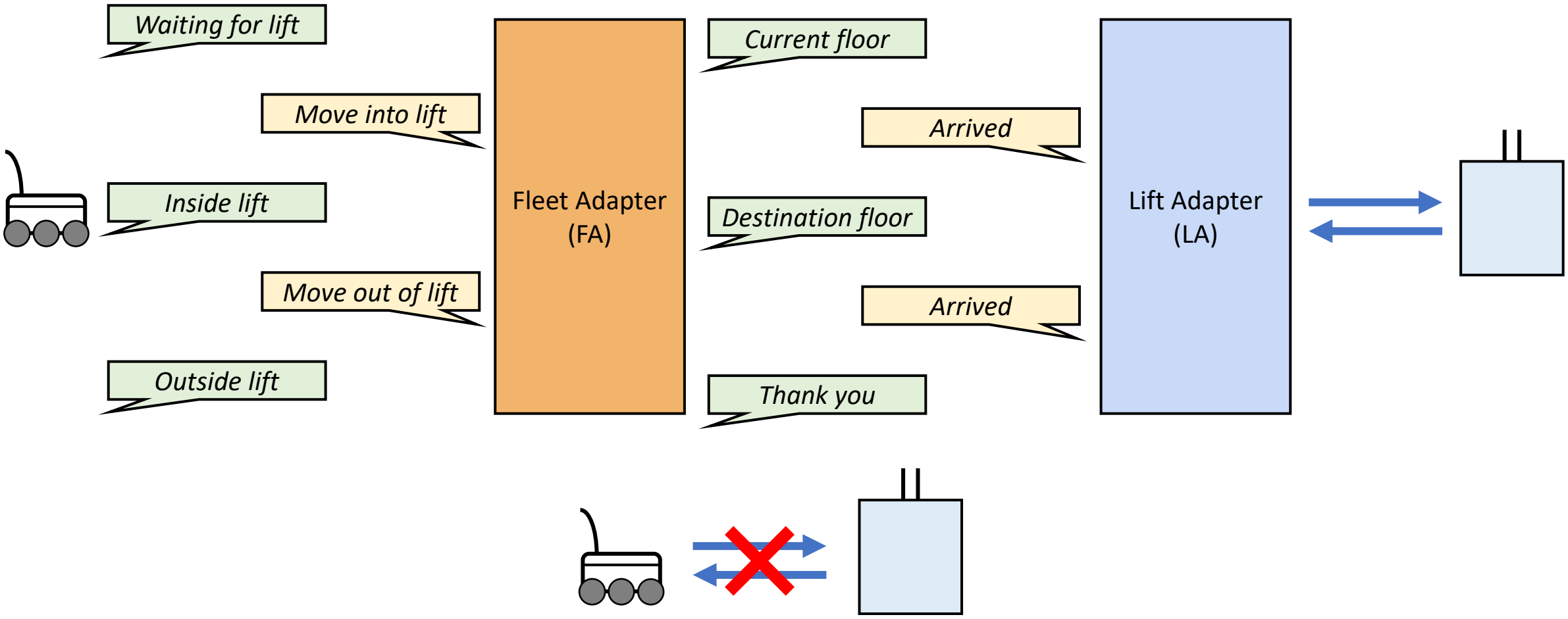
Every floor will require a new add-on kit:

- ***Proximity sensor*** *to check if door is open*
- ***Screen*** *to display a dynamic QR code*

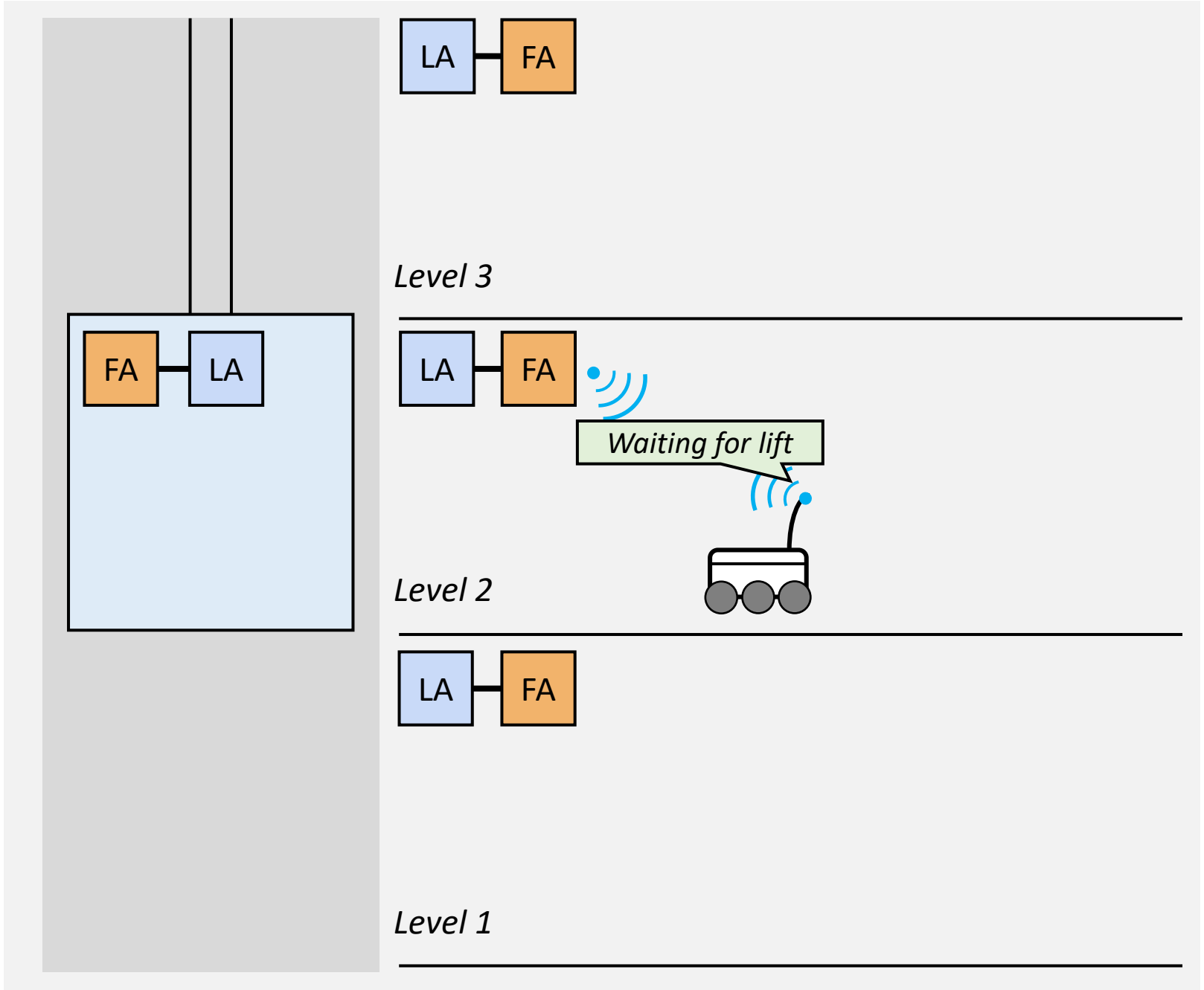


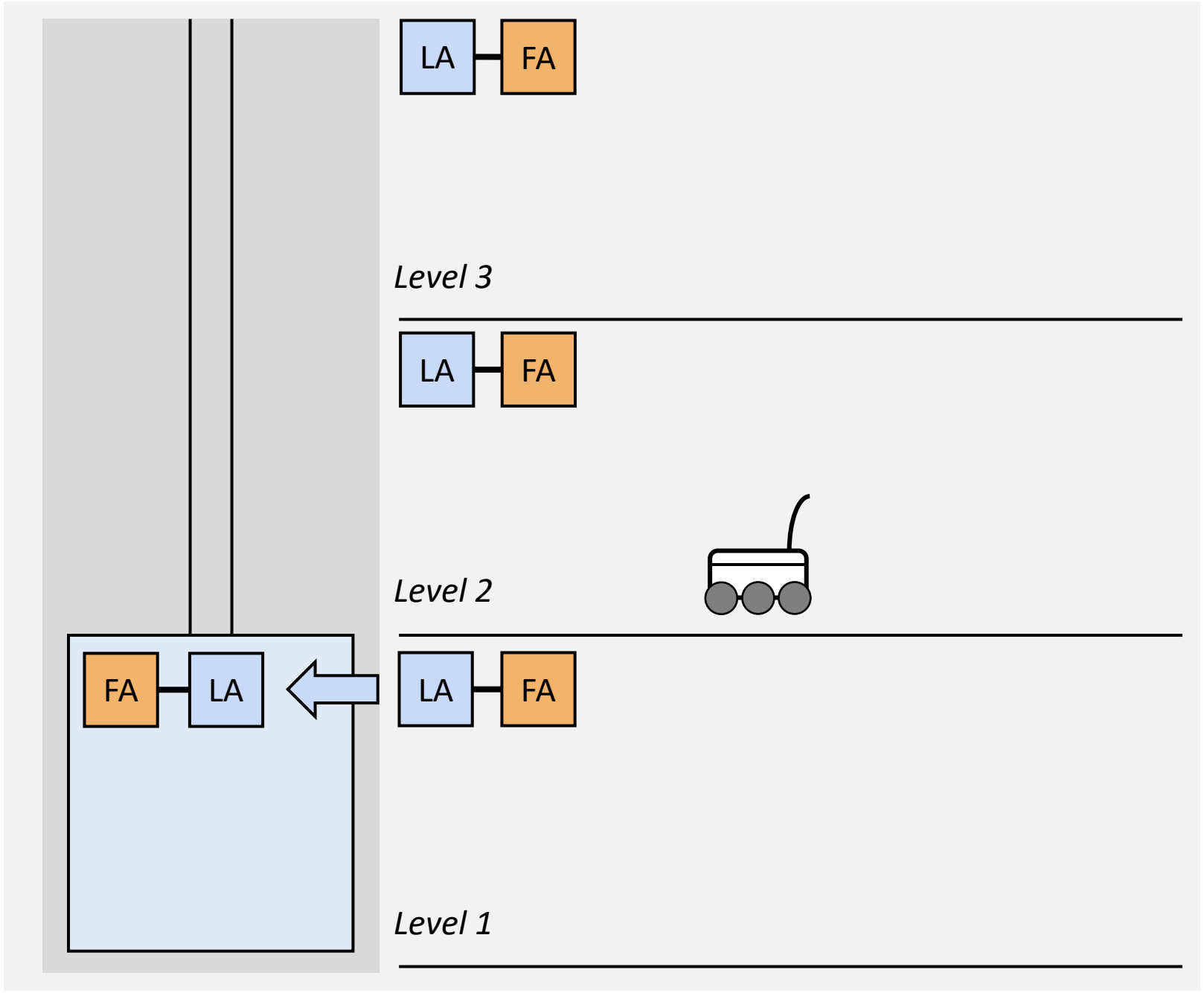
Passing along information through visual means when lift door is open

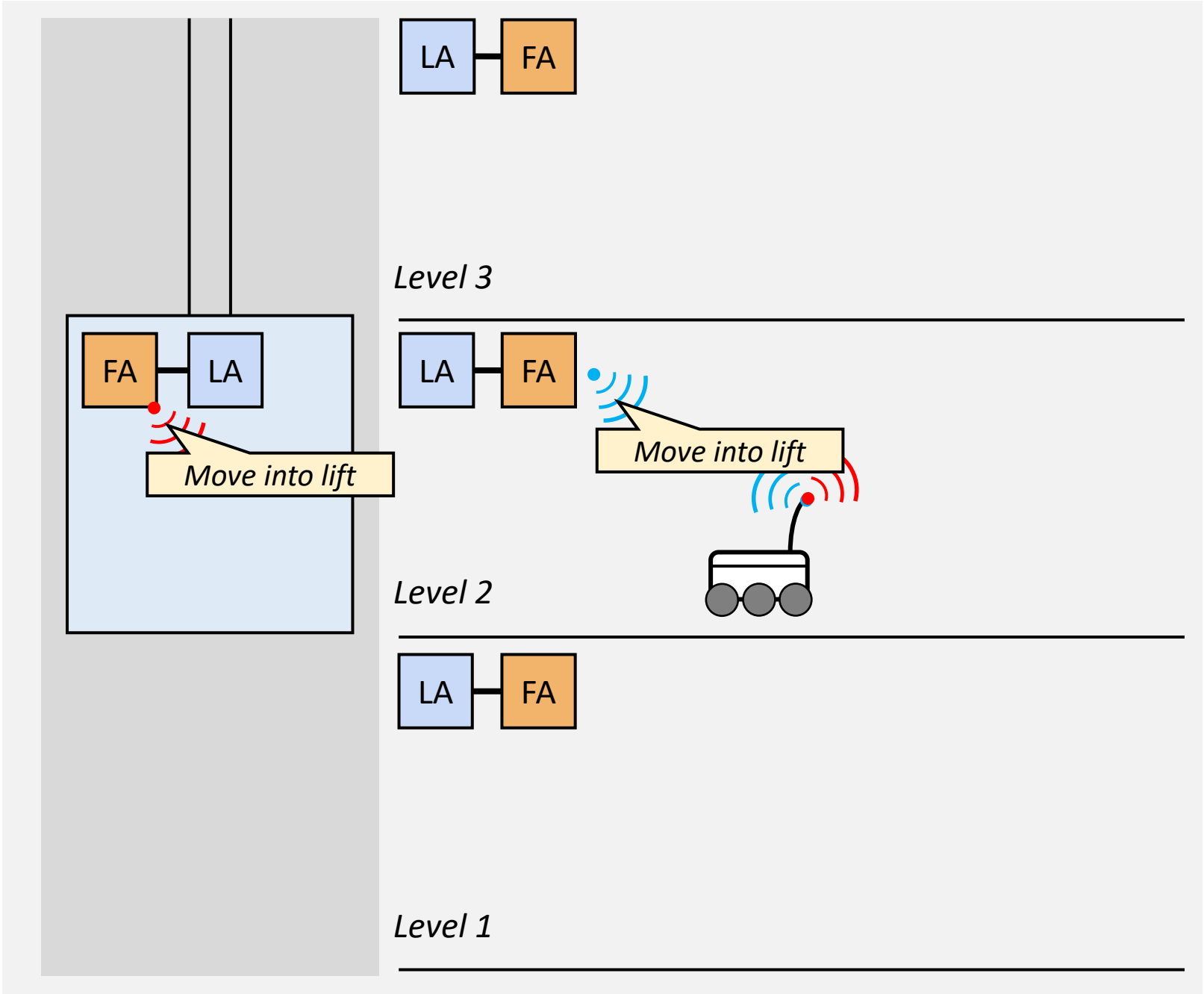
When robot is only connected to FA outside, LA inside is still informed

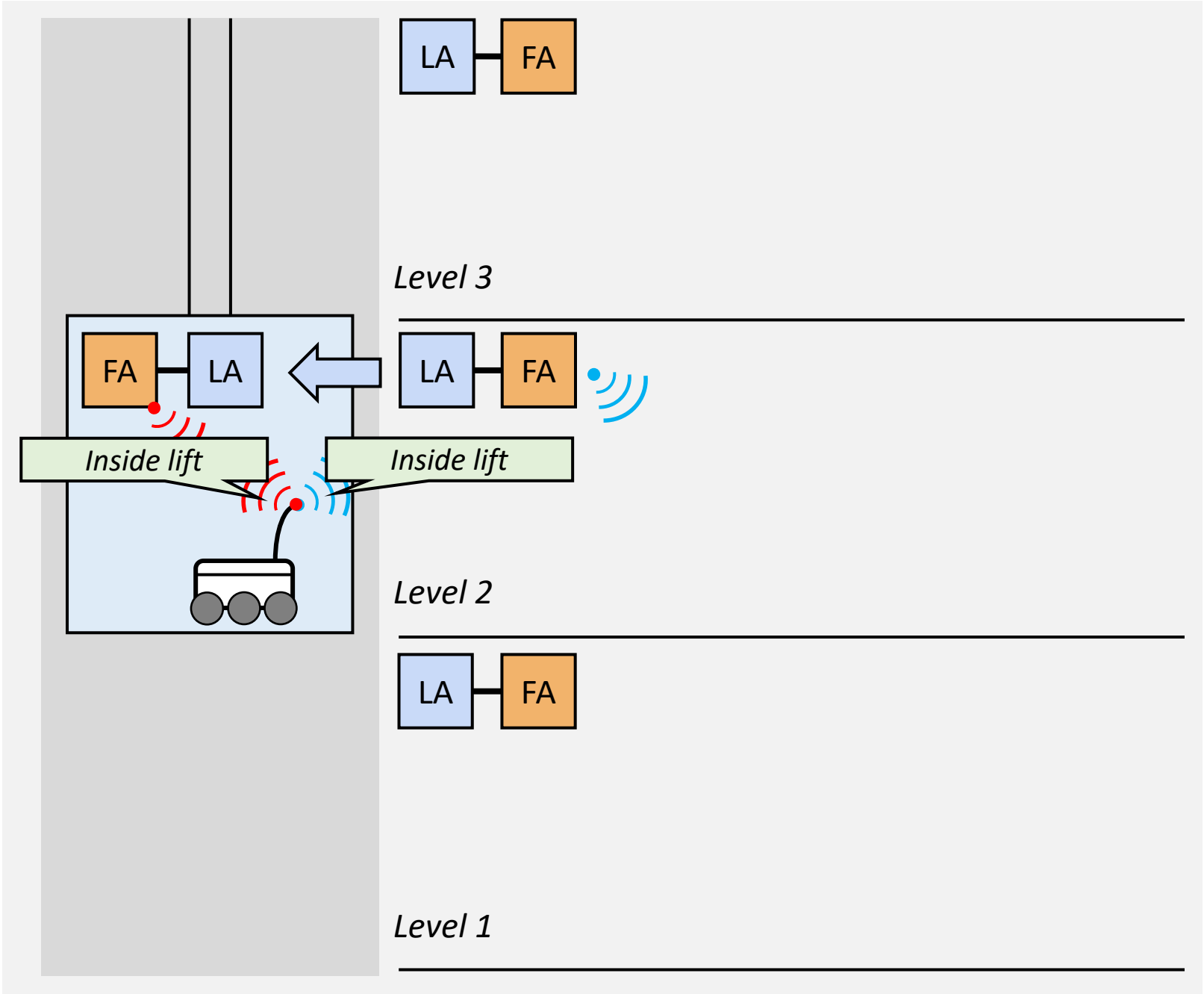


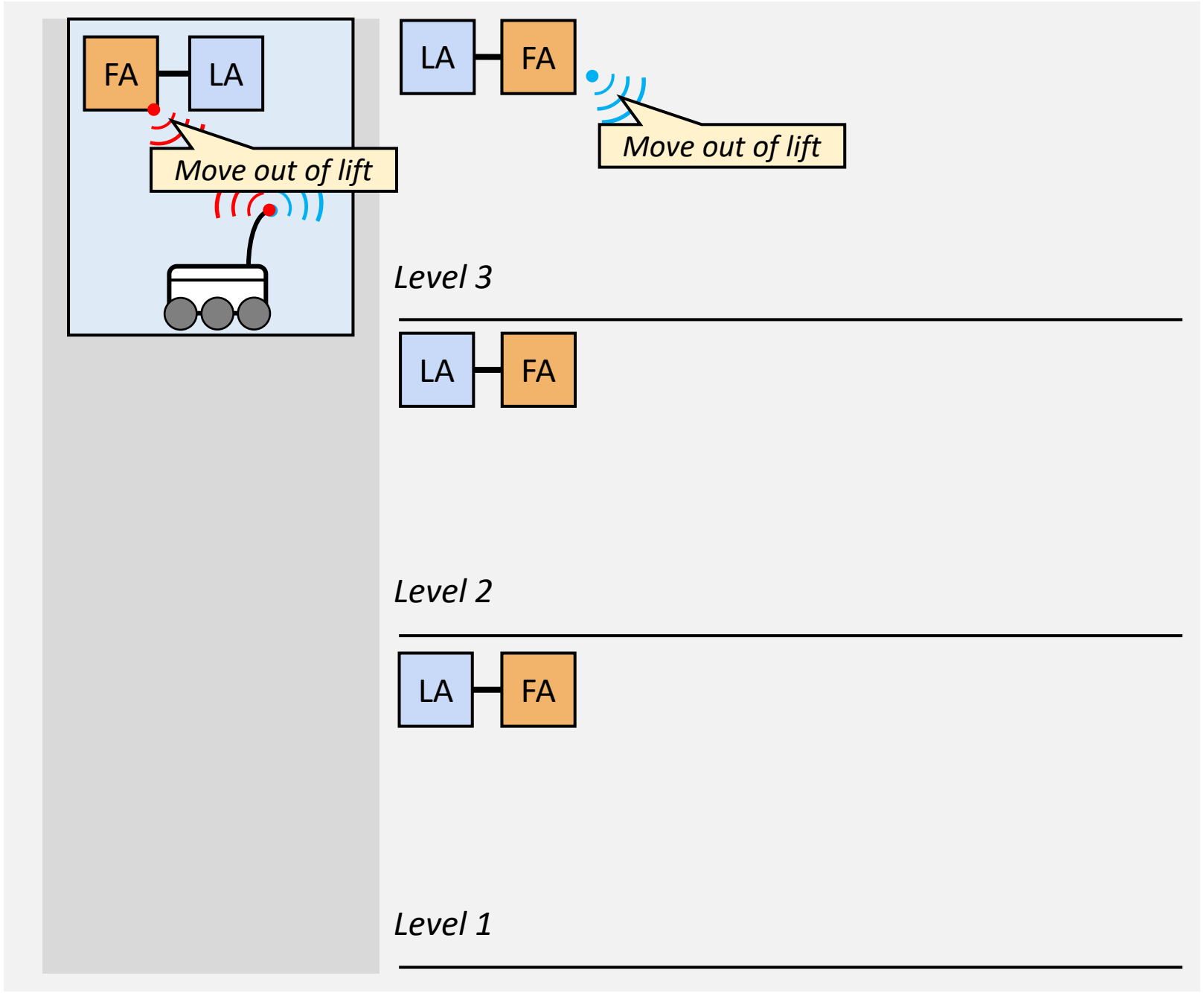
Summary of all RMF communications in a single operational cycle

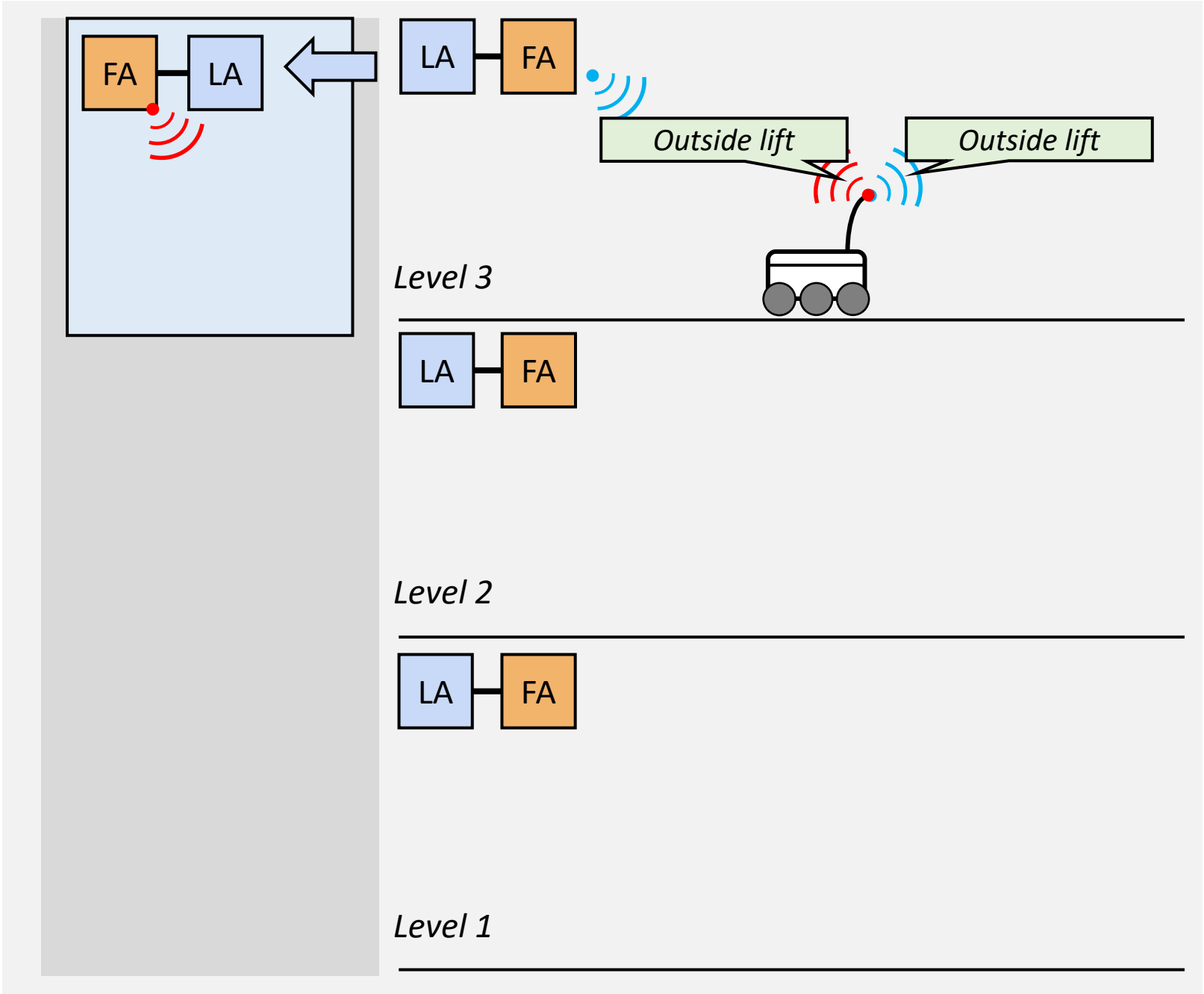

LA
FA
Level 3
FA
LA
LA
FA
Waiting for lift
Level 2
LA
FA
Level 1


LA
FA
Level 3
LA
FA
Level 2
FA
LA
LA
FA
Level 1


LA
FA
Level 3
FA
LA
Move into lift
LA
FA
Move into lift
Level 2
LA
FA
Level 1


LA
FA
Level 3
FA
LA
LA
FA
Inside lift
Inside lift
Level 2
LA
FA
Level 1


FA
LA
Move out of lift
LA
FA
Move out of lift
Level 3
LA
FA
Level 2
LA
FA
Level 1


FA
LA
LA
FA
Outside lift
Outside lift
Level 3
LA
FA
Level 2
LA
FA
Level 1