



TOPPERS箱庭の新たな展開 ：プラットフォーム化への一歩



森 崇
(永和システムマネジメント)



アジェンダ

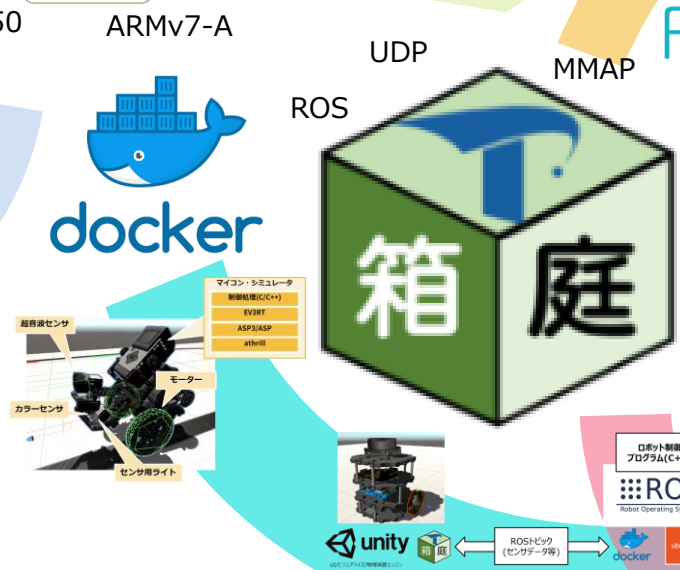
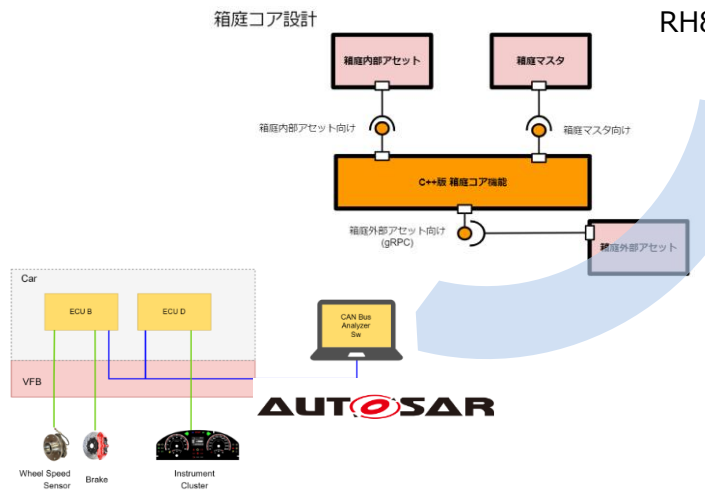
1. 箱庭の軌跡
2. プラットフォーム化への一歩
3. 新アーキテクチャの成果

箱庭の軌跡



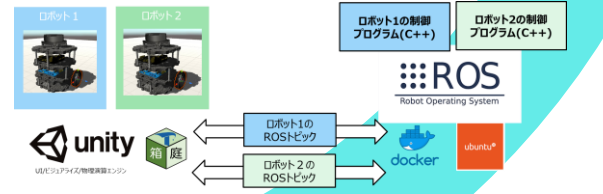
組み込み系エンジニア

ロボットAI系エンジニア



ロボット開発の新しいカタチ

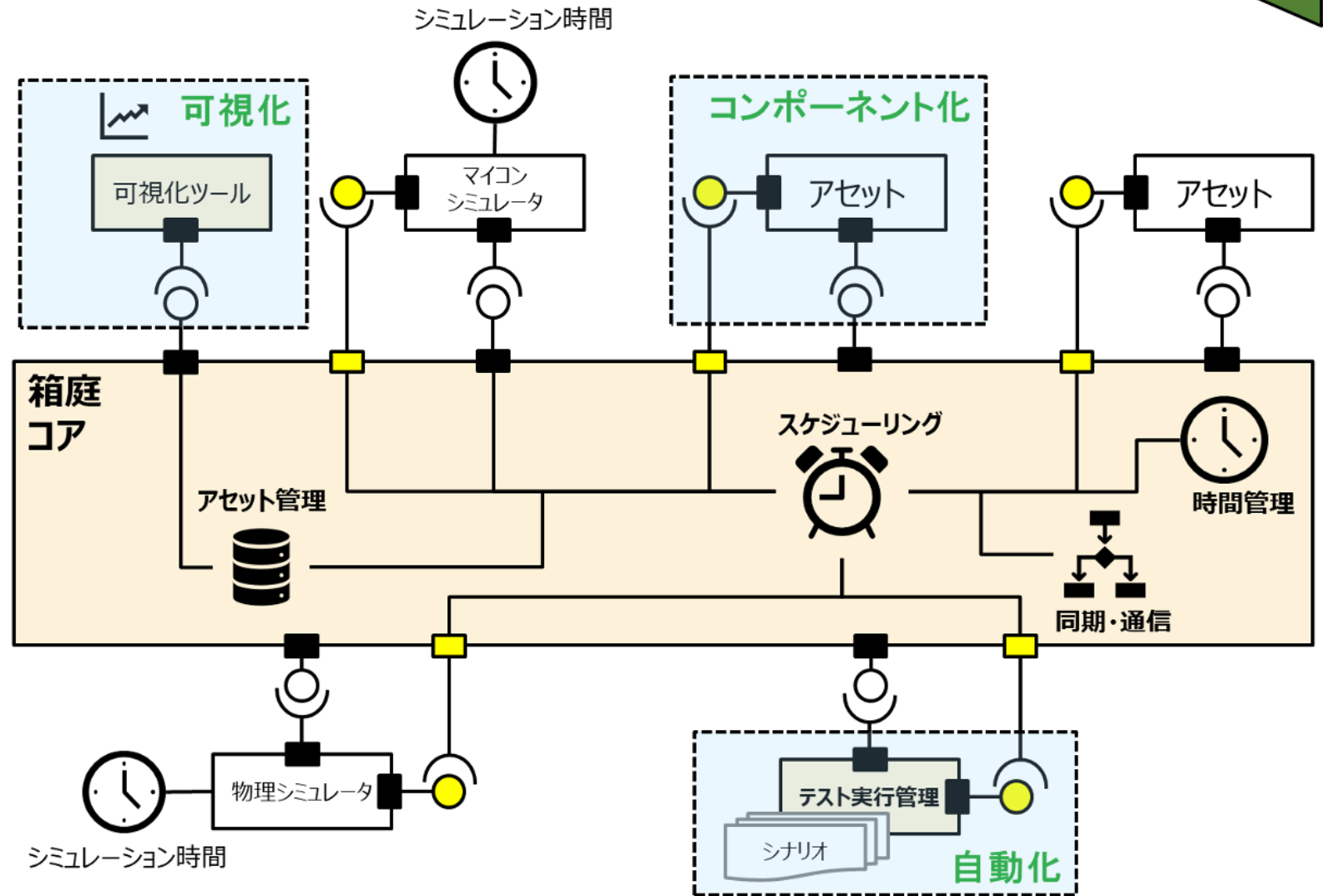
仮想世界と現実世界を同じインフラ環境で開発できるようになる！



ロボット制御系エンジニア

箱庭コア機能の役割

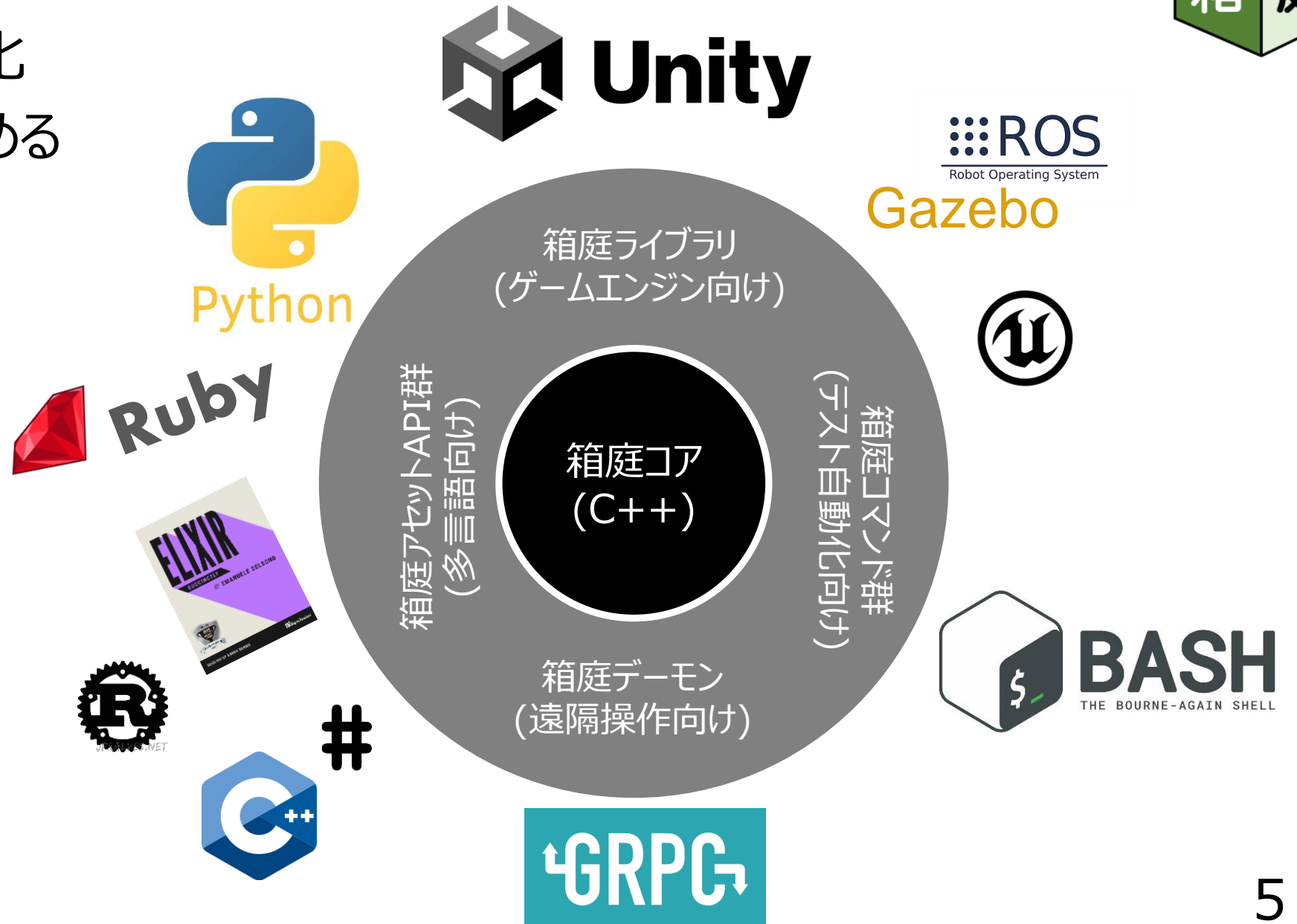
- スケジューリング
- 同期・通信
- 時間管理
- アセット管理





C++版箱庭コア機能の狙い

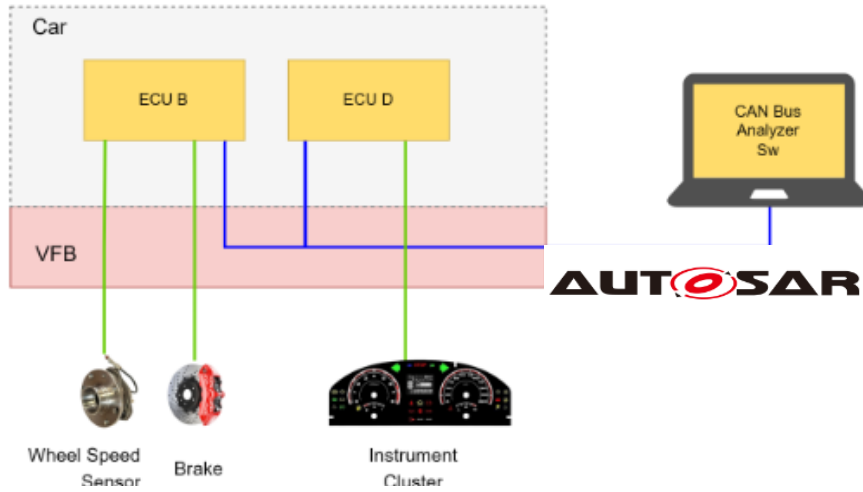
- ・箱庭コア機能を凝集化
- ・汎用性・拡張性を高める
- ・将来への布石





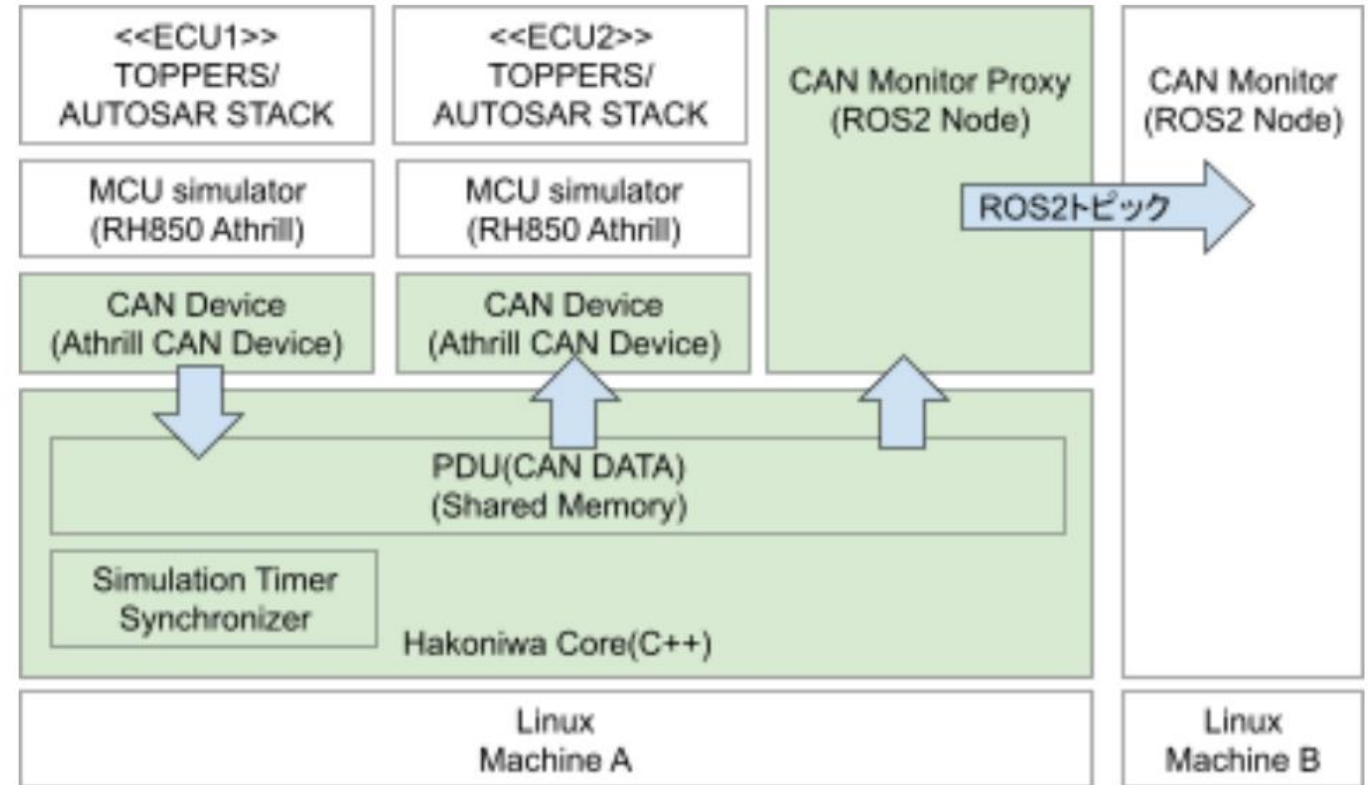
TOPPERS活用アイデア・アプリケーション開発コンテスト

- 2ECU構成で、C++版箱庭コア機能によってスケジューリングされた
- RH850版 AthrillがTOPPERS/AUTOSARスタックを実行し、
- ECU間のCAN通信データをROS2トピックで観測できる



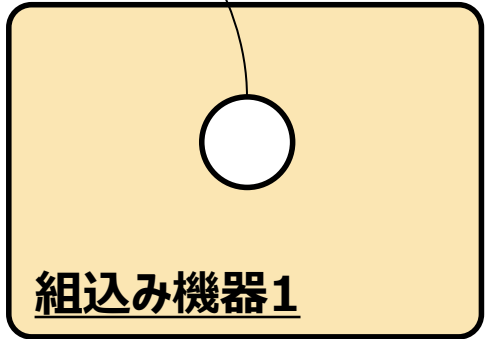
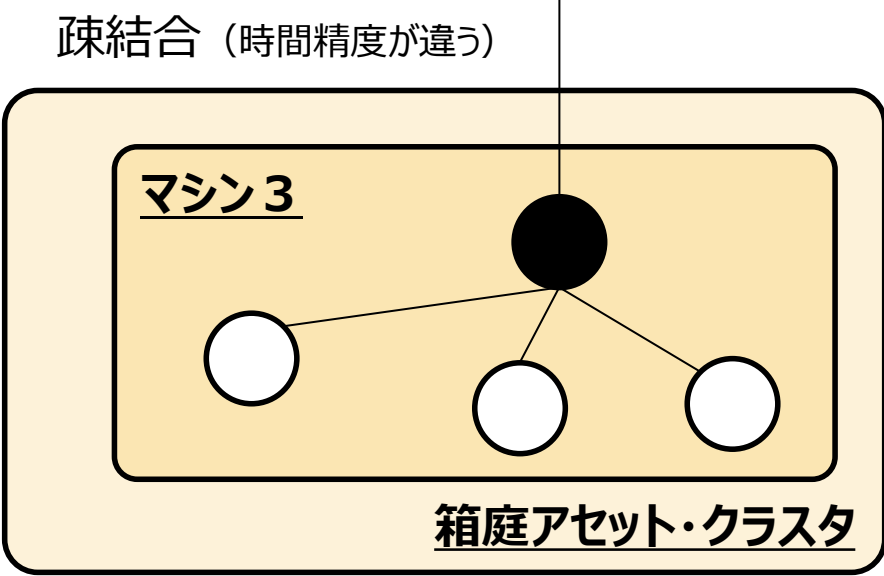
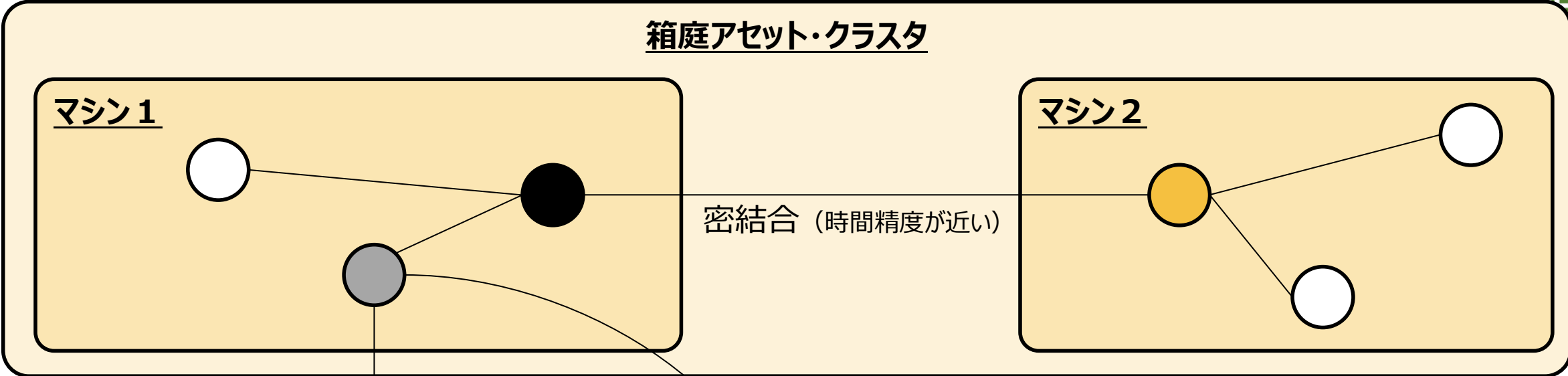
アプリケーション開発部門

金賞
C++版箱庭コア機能
森 崇((株)永和システムマネジメント)
コンテスト応募資料
リポジトリ (コア機能)
リポジトリ (コマンド類)





箱庭の分散シミュレーション構成



- [凡例]
- 箱庭アセット・クラスタ**
シミュレーション時間精度に近いアセットの集合
 - 箱庭コンダクタ(サーバー)**
箱庭アセット・クラスタ内に1個存在し、箱庭時間を管理する
 - 箱庭コンダクタ(クライアント)**
箱庭アセット・クラスタ内に0個以上配置できる
物理的にサーバー側と分離する必要がある場合に利用する
 - 箱庭プロキシ**
接続するクラスタ数が2以上になった場合、
箱庭コンダクタ間の時間同期とPDUデータ通信を実現する
 - 箱庭アセット**
箱庭シミュレーションの最小単位

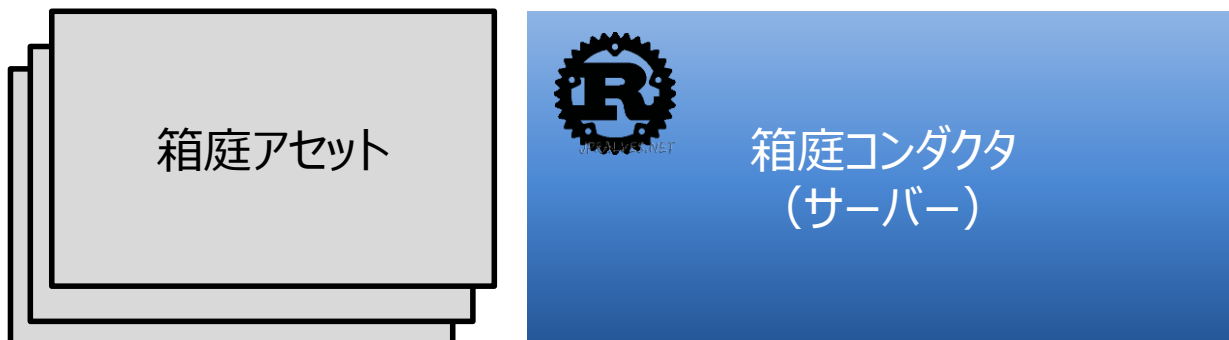


分散シミュレーションを実現する箱庭コンダクタとは

箱庭上にあるアセット間のシミュレーション調停する

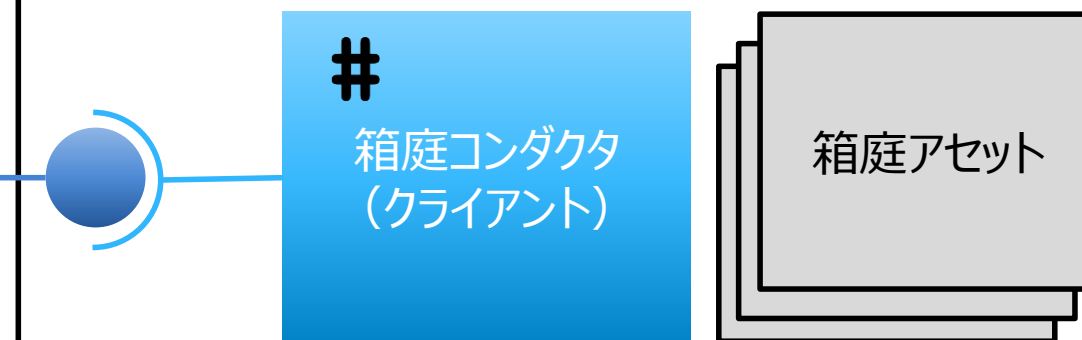
マシン1 (サーバー側)

- gRPCベースのRPCサーバーです
 - 箱庭アセットの登録
 - シミュレーション実行・停止・リセット
 - PDUデータのI/O



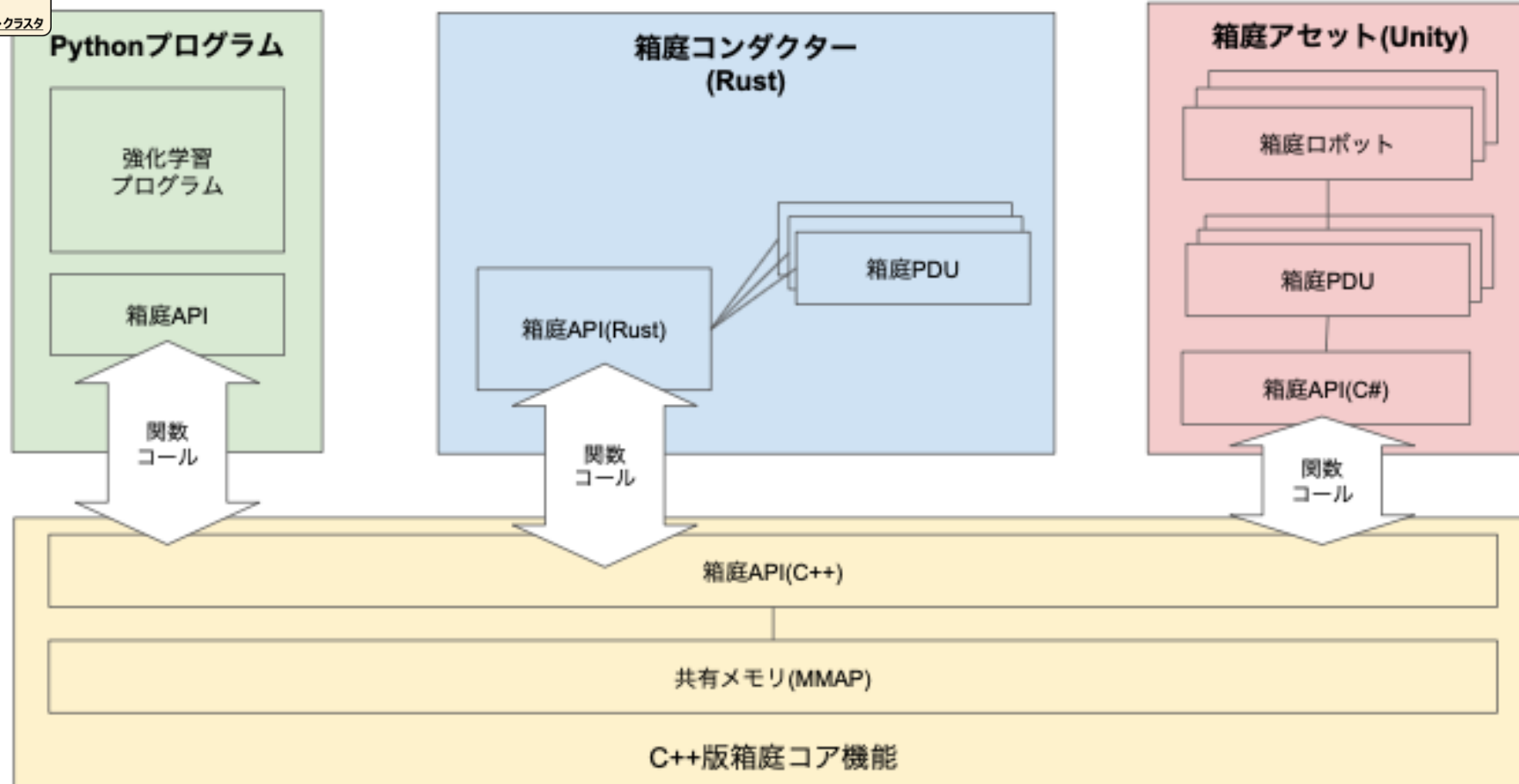
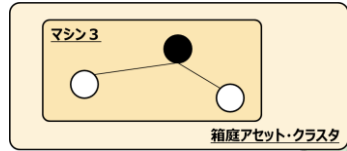
マシン2 (クライアント側)

- gRPCベースのRPCクライアントです
- 箱庭アセットをサーバー側に接続する



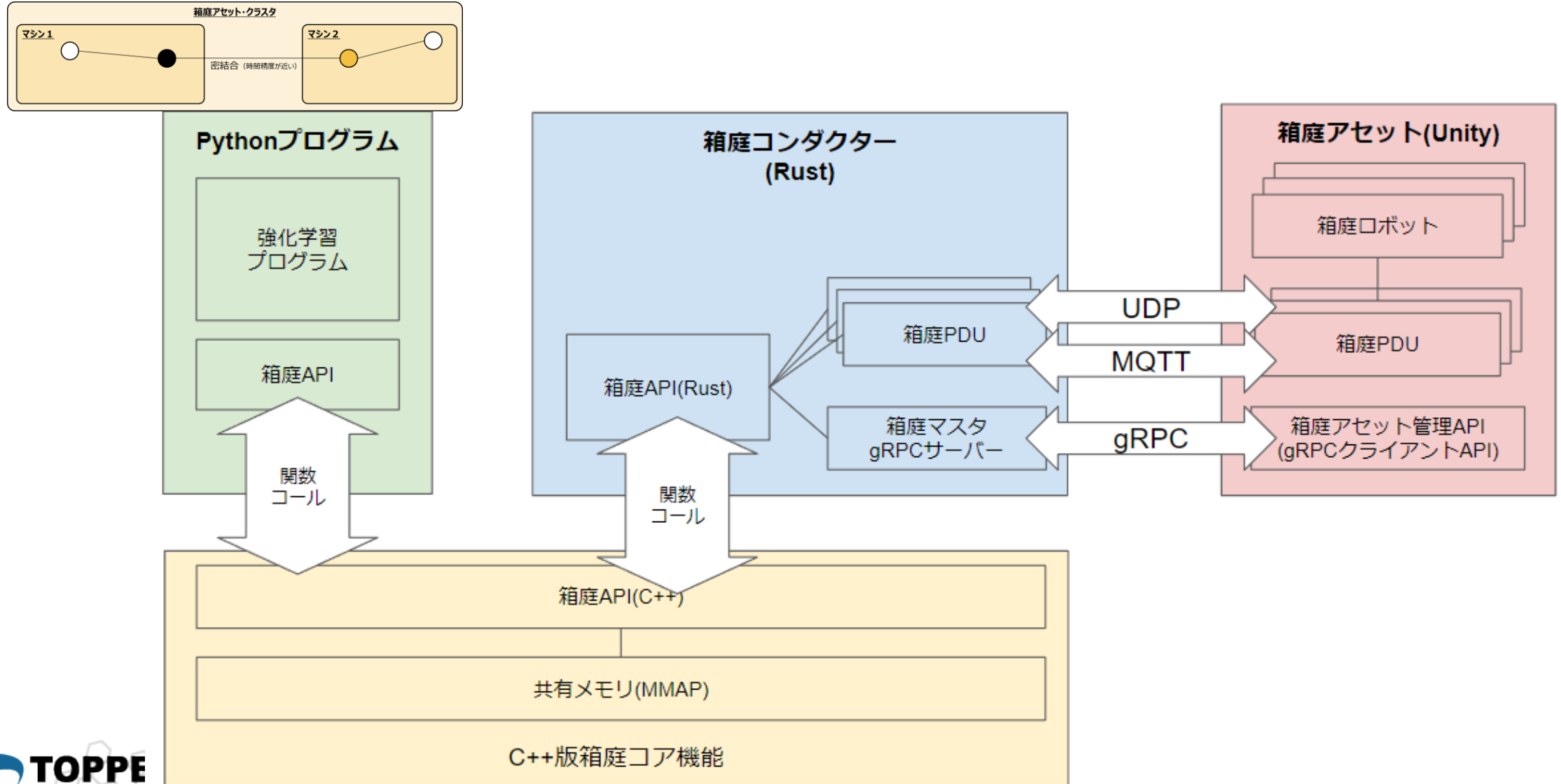
分散シミュレーションの構成例 (1 / 3)

- 1 クラスタ / 1 マシン構成



分散シミュレーションの構成例 (2 / 3)

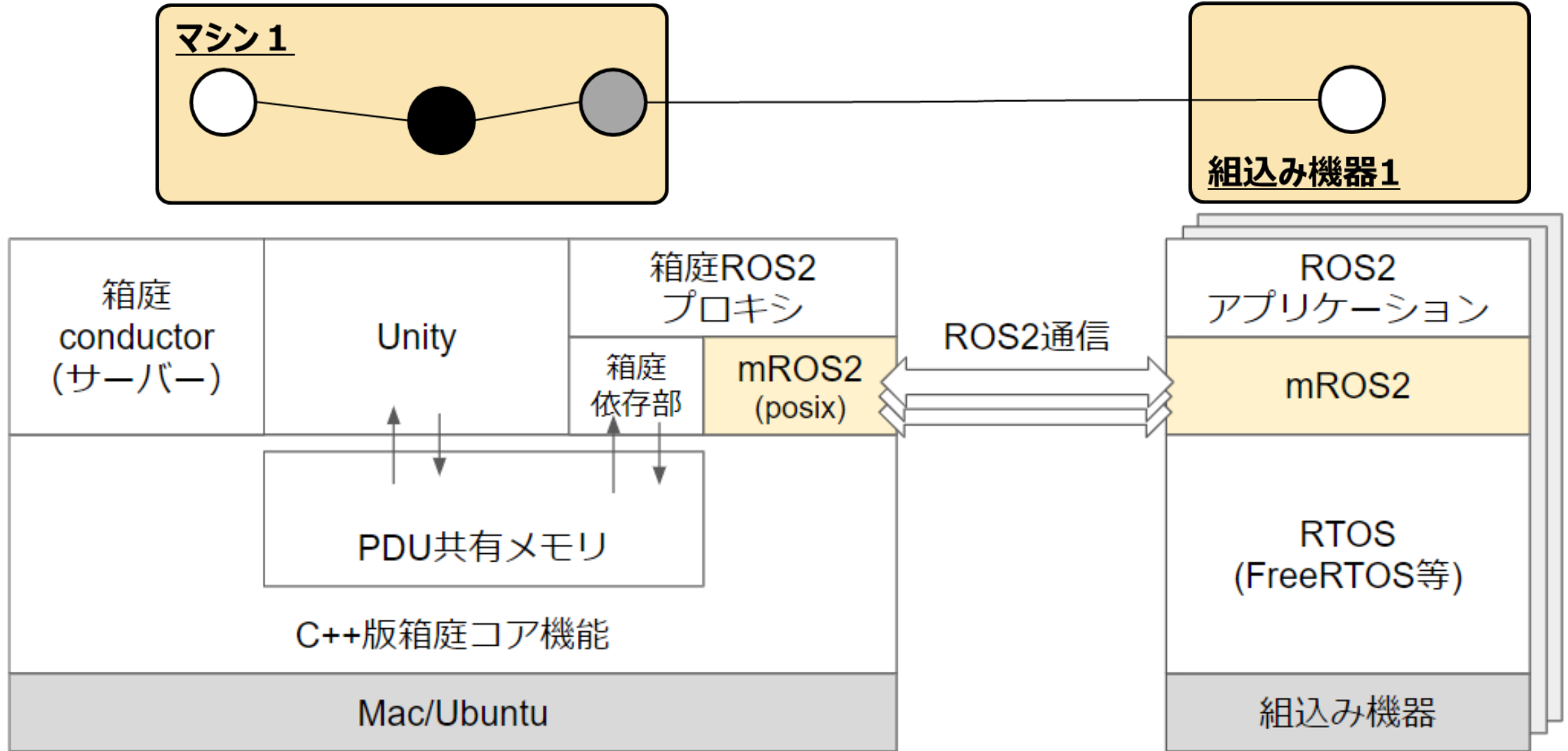
- 1 クラスタ / 2 マシン構成





分散シミュレーションの構成例 (3 / 3)

- 1 クラスタ / 1 組み込み機器構成





箱庭がmROS2を採用する理由

- ① 軽量であること
- ② 汎用OS上でも動作可能であること(mROS2-POSIXの存在)
 - ★ Macで動かしたい！
- ③ 組み込み機器との連携をROS2ベースでやれる



Supported platform

Since this repository itself only maintains the communication layer of mROS 2, we also publish the implementation on the embedded board using this layer and the embedded kernel. Please see each repository to learn how to use it.

name	kernel	board	support status
mros2-asp3-f767zi	TOPPERS/ASP3	STM32 NUCLEO-F767ZI	currently supported up to v0.3.2 (see status)
mros2-mbed	Mbed OS 6	Mbed enabled boards having an Ethernet port (See detail)	well maintained and easy to try
mros2-esp32	ESP-IDF FreeRTOS	ESP32 boards with 2.4 GHz Wi-Fi (See detail)	well maintained
mros2-posix	POSIX (pthread)	any machine that runs Linux	partly maintained

Please let us know if you have a request for support for other boards/kernels, or if you could implement this layer on other platforms.



アジェンダ

1. 箱庭の軌跡
2. **プラットフォーム化への一歩**
3. 新アーキテクチャの成果



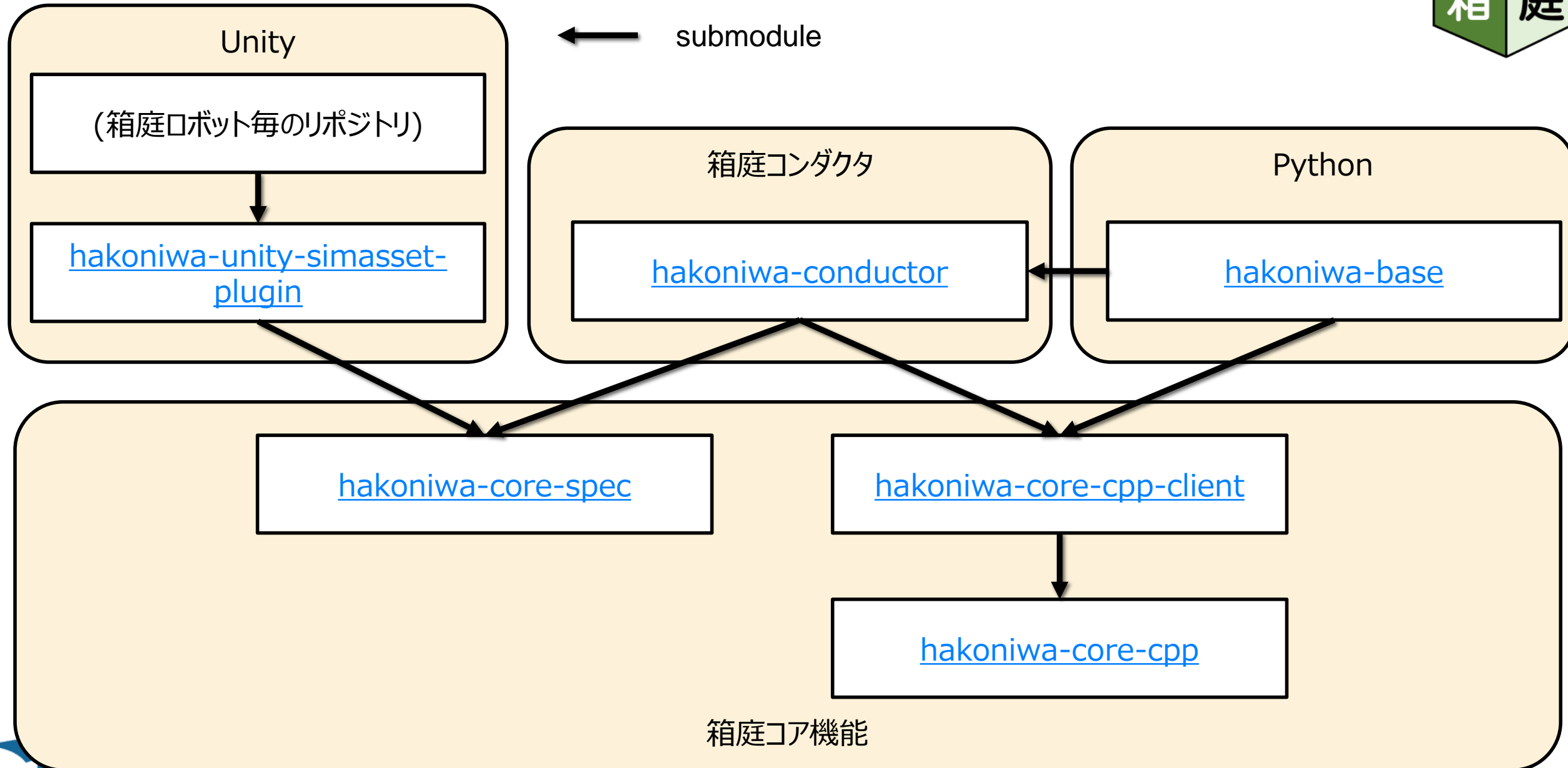
プラットフォーム化への一歩

宣言：今の箱庭をプラットフォーム化します！

課題・動機	今年度の取り組み
<ul style="list-style-type: none">・リポジトリが散らばっており、利用可能なものがよくわからない・箱庭環境を構築するためのノウハウが非公開な状態	<ul style="list-style-type: none">・箱庭リポジトリの整理・箱庭のインテグレーションプロセスの自動化
<ul style="list-style-type: none">・箱庭のロボットを作ることができる人を増やしたい	<ul style="list-style-type: none">・箱庭のロボットを作成するための環境の構築・箱庭のロボットをUnityアセットストアで公開
<ul style="list-style-type: none">・そもそも、箱庭が目指す姿に近づいている？	<ul style="list-style-type: none">・箱庭の技術的な方向性を検討する・特に、LLM/生成系AIが箱庭コンセプト実現にどう寄与するか？を検討したい→箱庭は人間のインスピレーションを加速するP/Fであるべき



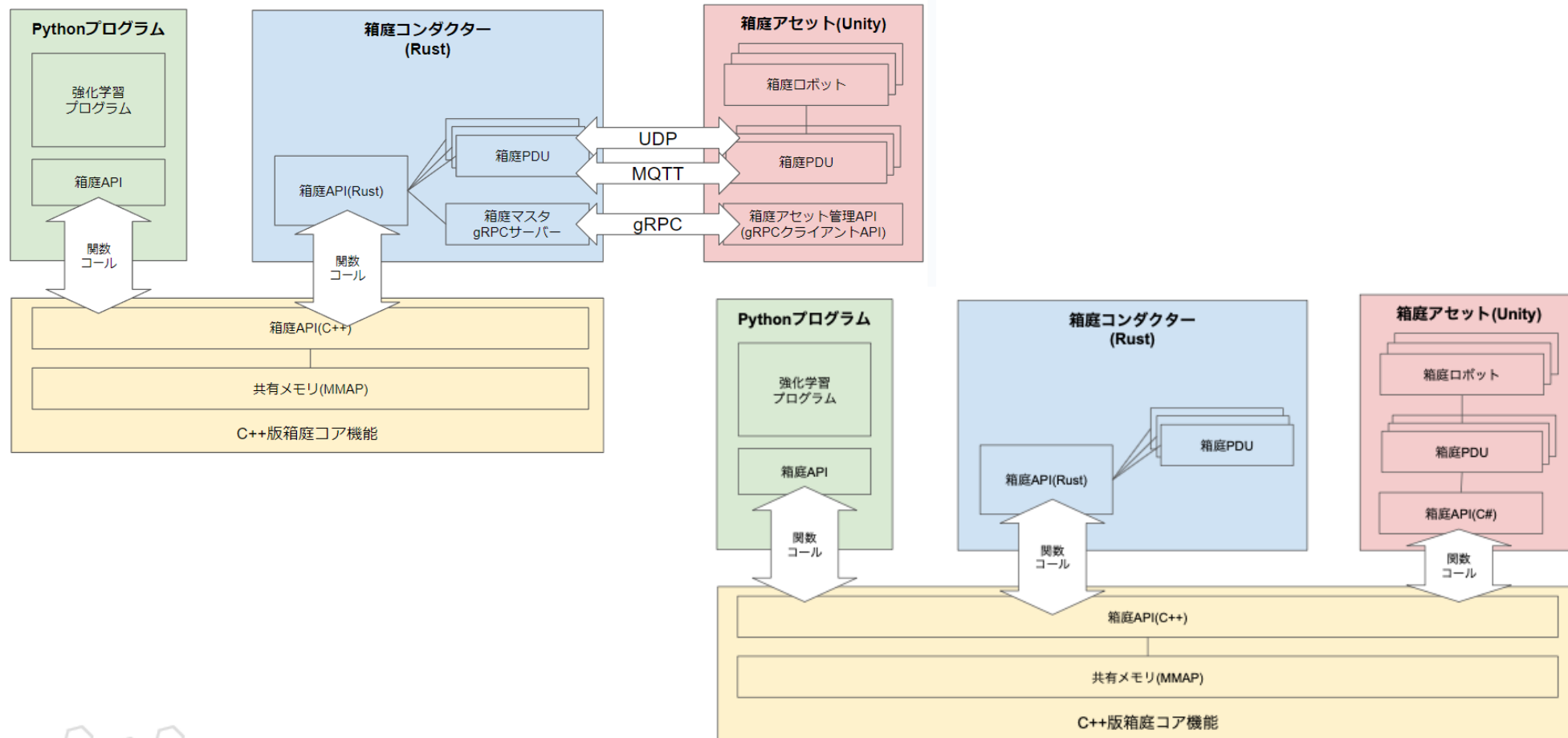
現時点の箱庭リポジトリ構成





箱庭のインテグレーションプロセスの自動化

- 箱庭モジュール組み合わせの自由度を上げる
- OSに応じた組み合わせの自動選定／インストールの仕組み





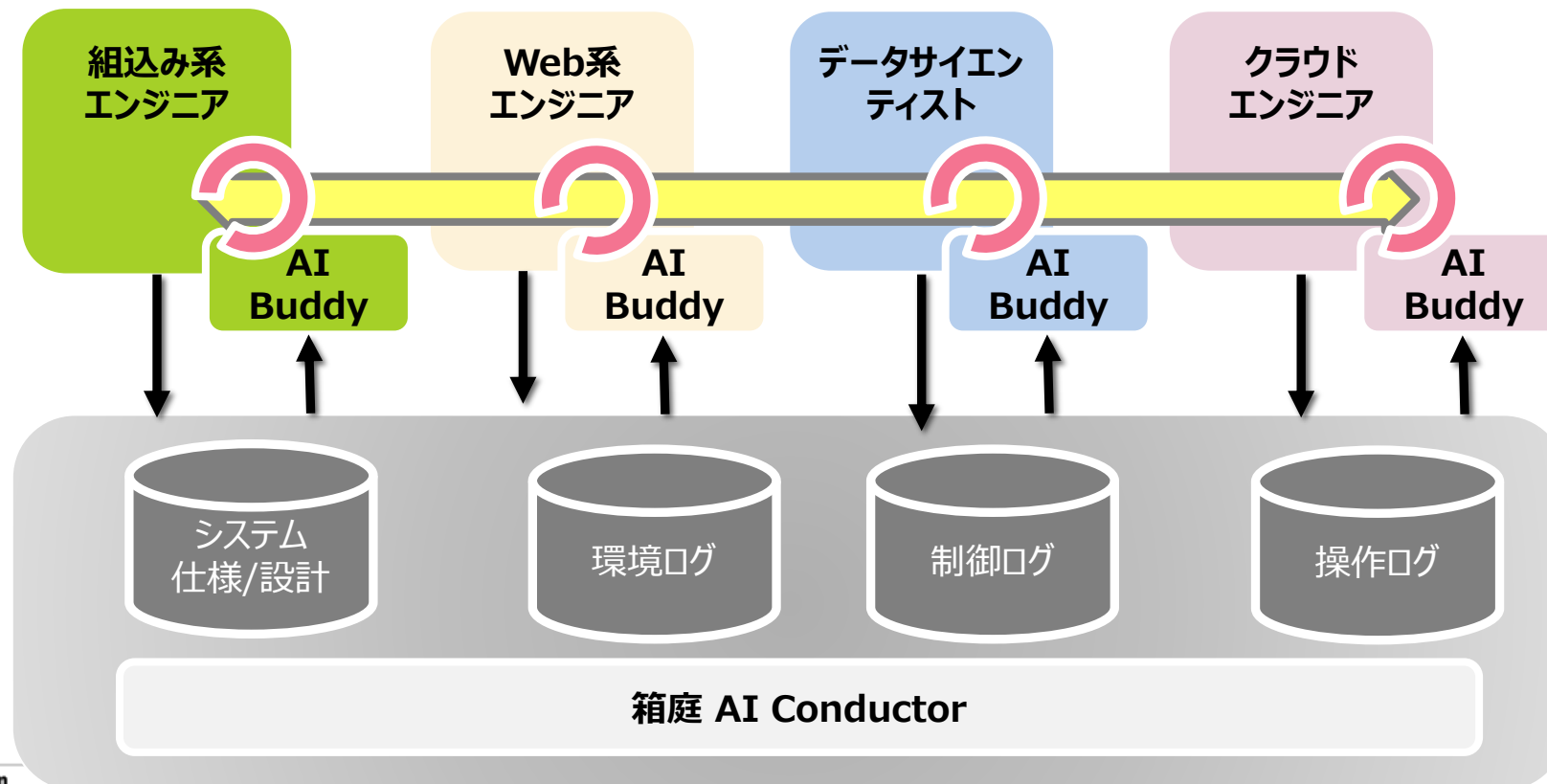
箱庭ロボットの開発環境と箱庭アセットの一般公開

箱庭はロボットの開発／統合に必要な道具を用意します



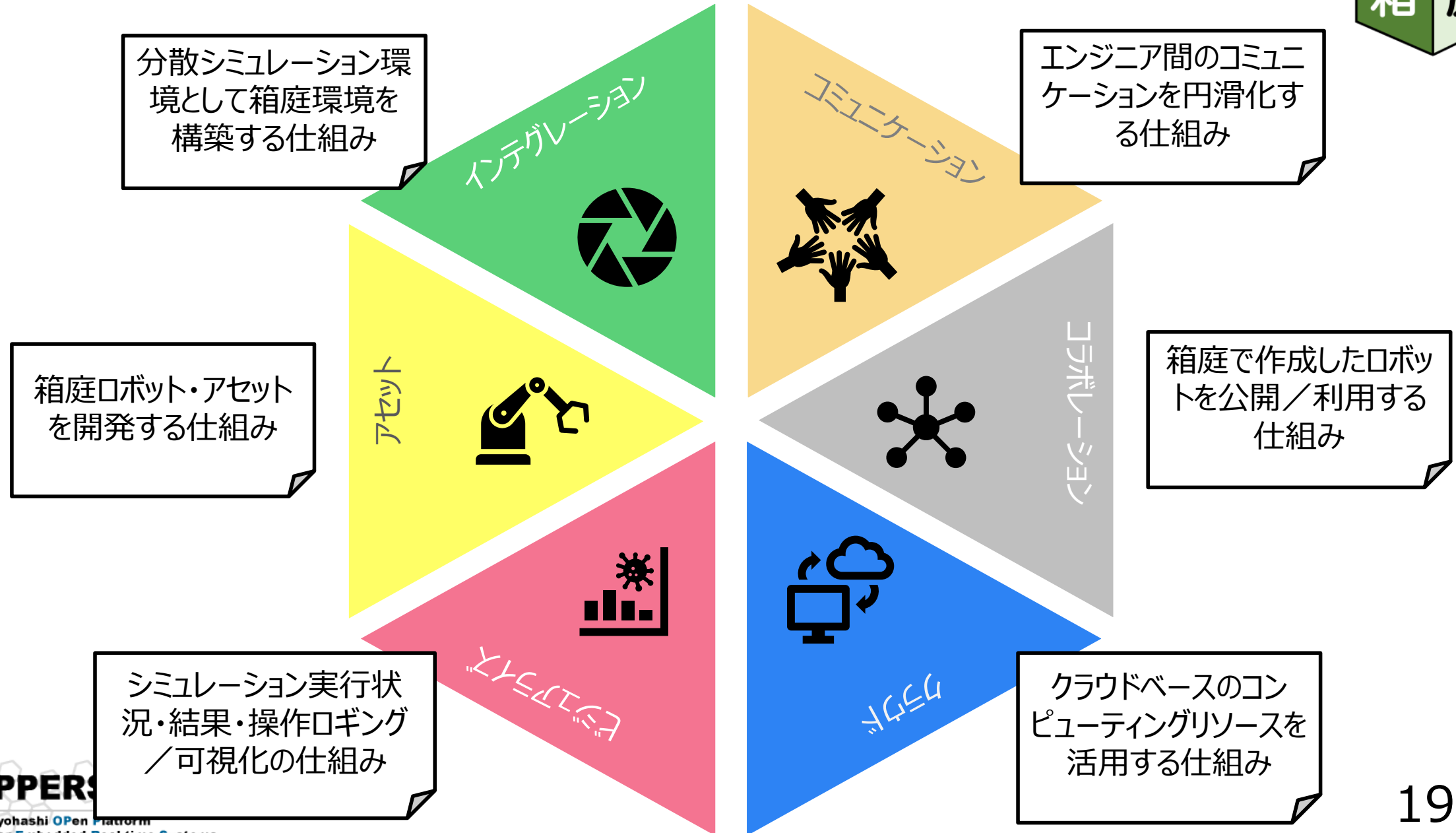
エンジニアのインスピレーションを加速するP/Fとして

- エンジニア間のコミュニケーションを円滑化するための仕組みとしてのAI Buddy
- AI Buddyが開発者とペアで開発するスタイル
 - AI Buddyは、エンジニアと一緒に開発を進める
 - AI Buddyは、箱庭上で実行したロギング情報にアクセスして問題解決を一緒に手伝ってくれる
 - AI Buddyは、職種を越えて、他のエンジニア向けに翻訳してくれる





箱庭プラットフォームの方向性





アジェンダ

1. 箱庭の軌跡
2. プラットフォーム化への一歩
3. **新アーキテクチャの成果**

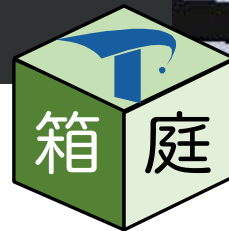
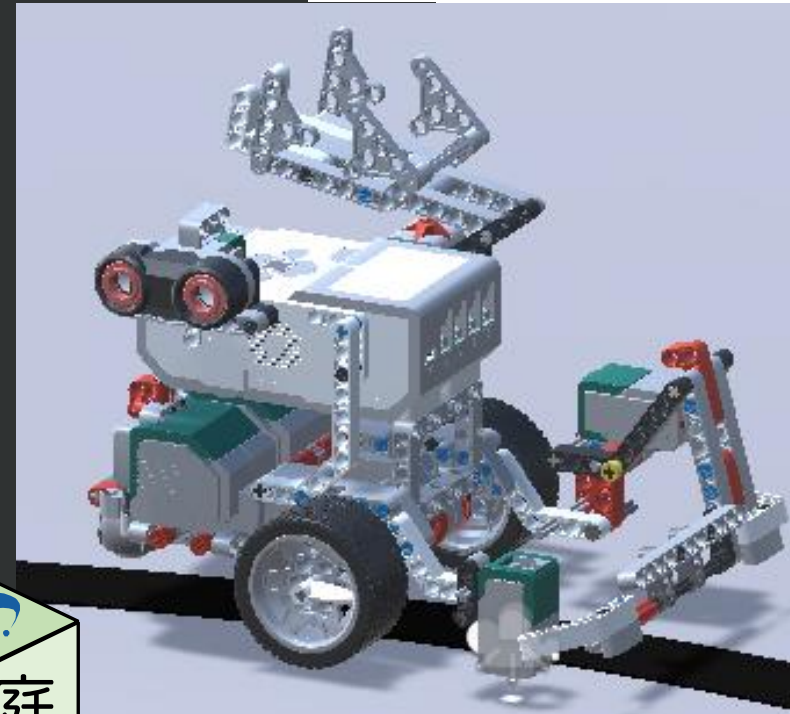
箱庭強化化学習

- UnityとPython使って、ロボットの強化学習を手軽にためせます！



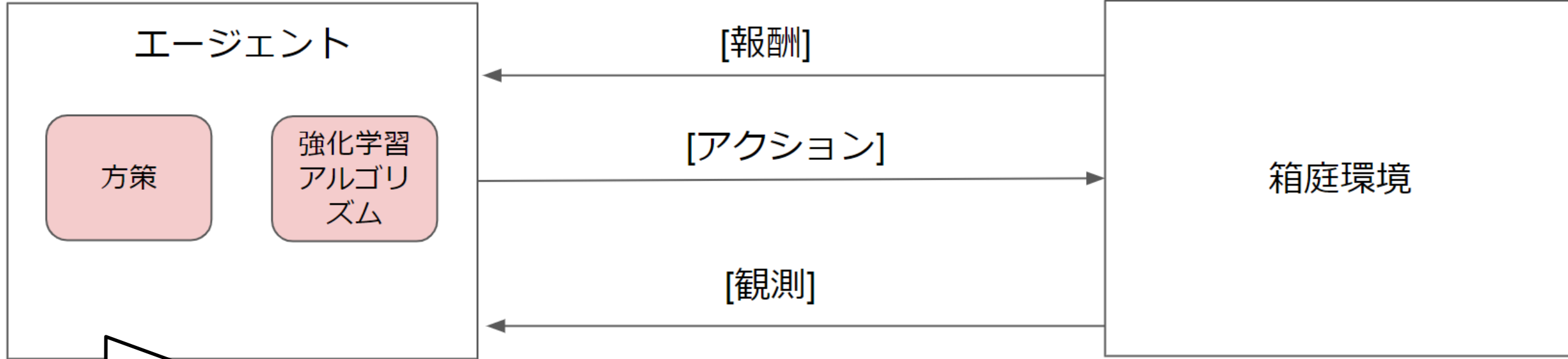
```
ai model
l = qtable_model2.get_model(env.roboto().num_states(), env.roboto().num_actions())
l.load('./dev/ai/qtable_model.csv')

simulation
= env.roboto()
for episode in range(100):
    total_time = 0
    done = False
    state = 0
    total_reward = 0
    while not done and total_time < 4000:
        action = model.get_action(state)
        next_state, reward, done, _ = env.step(action)
        total_reward = total_reward + reward
        model.learn(state, action, reward, next_state)
        state = next_state
        total_time = total_time + 1
    env.reset()
    model.save('./dev/ai/qtable_model.csv')
```





箱庭強化学習のアーキテクチャ



Pythonプログラム

- ※ Qテーブルを使った学習をさせていますが、
- ※ 他の学習アルゴリズムでも対応可能

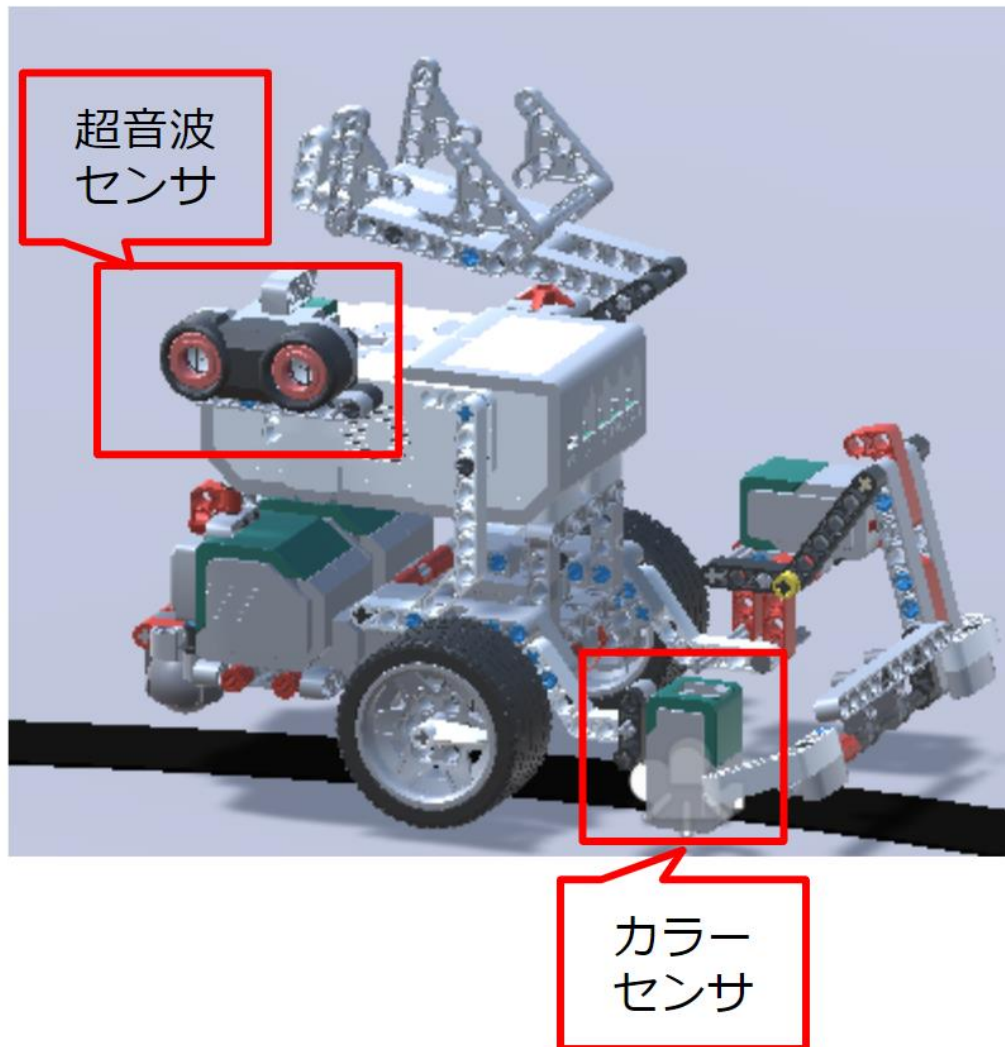
@kanetugu2018 (Mori Takashi) in TOPPERSプロジェクト
投稿日 2023年01月29日 更新日 2023年06月01日 40132 views

Unity + Python + 箱庭でロボットを強化学習させてみよう！

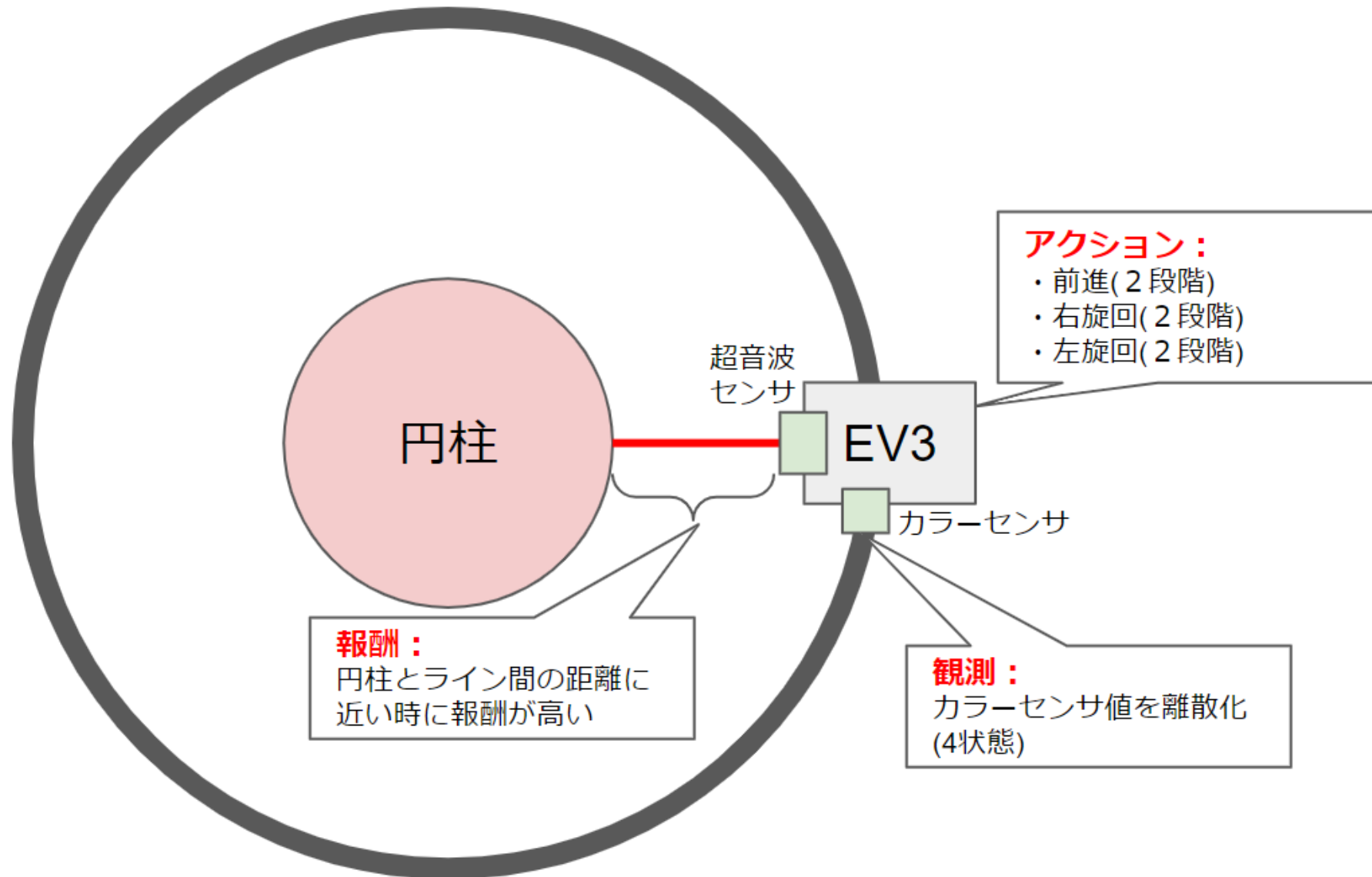
Python, Unity, 強化学習, 箱庭

<https://qiita.com/kanetugu2018/items/ee9b2bf0596a35e42b03>

箱庭強化学習で使用するロボット



強化学習デモ環境の説明



Qテーブル設計(ロボット視点)



状態	前進		右旋回		左旋回	
	Power=50	Power=20	Power=50	Power=20	Power=50	Power=20
黒	??	??	??	??	??	??
黒に近い	??	??	??	??	??	??
白に近い	??	??	??	??	??	??
白	??	??	??	??	??	??

https://github.com/toppers/hakoniwa-base/blob/ai/workspace/dev/ai/qtable_model.csv



強化学習デモ

未学習状態:

```
total_time = 0
done = False
state = 0
total_reward = 0
#100secs
while not done and total_time < 4000:
    action = model.get_action(state)
    next_state, reward, done, _ = env.step(action)
    total_reward = total_reward + reward
    #print("state=" + str(state))
    #print("reward=" + str(reward))
    #print("action=" + str(action))
    #print("done=" + str(done))
    #print("total_time=" + str(total_time))

    model.learn(state, action, reward, next_state)

    state = next_state
    total_time = total_time + 1

env.reset()
print("episode=" + str(episode) + " total_time=" + str(total_time) + " total_reward=" + str(total_reward))
```

学習済み状態:

停止 シミュレーション時間[秒] 0.000

```
str(total_reward))
```

asset_def.txt M sync_mode: true
\$ asset_env.bash episode=10 total_time=163 total_reward=11275
\$ install.bash sync_mode: true
episode=11 total_time=4000 total_reward=440586
sync_mode: true



ChatGPT連携

- ChatGPTにロボットの動作ルール（仕様）をプロンプトとして与え
- センシング情報を入力としてモーター指示値を生成させる

ChatGPTのAPI使って、Unity上の箱庭ロボットを動かしてみた！

Python, Unity, TOPPERS, 箱庭, ChatGPT

<https://qiita.com/kanetugu2018/items/8bcaa2e4ceb464262c19>

Rule1: あなたはロボットです。与えられた環境で自由に移動してください。

Rule2: ずっと同じ場所にとどまらず、動き続けてください。停止してはいけません。

Rule3: 障害物にぶつかってはいけません。

Rule4: 障害物との距離は、3方向（前、右、左）からセンサ値を取得できます。

Rule5: 3方向（前、右、左）のいずれかのセンサ値が、 0.5 以下の場合には衝突直前と判断して回避行動をとってください。回避する場合は、前進してはいけません。

Rule6: 移動するためのモーターがあります。モーター指示値としては、前進と回転の移動が可能です。

Rule7: 前進する場合のパラメータ値は、 $0.0 \sim 50.0$ の範囲となります。値が大きいほど早く進みます。値が0の場合は停止します。

Rule8: 回転する場合のパラメータ値は、 $-10.0 \sim 10.0$ の範囲となります。正の値は、左回転、負の値は、右回転です。値が大きいほど回転速度が高くなります。値が0の場合は回転しません。

Rule9: 回転する場合は、前進のモーター指示値は必ず 0.0 にしてください。

Rule10: モーター指示値が、前進、回転ともに0の場合、移動していないとみなします。

Rule11: モーターの指示値は以下の書式で1データのみ出力してください。理由およびコメントの出力は不要です。

```
{ "x": (value of x according to Rule7), "z": (value of z according to Rule9) }
```

Rule12: 障害物までの距離は、以下の書式で入力します。

```
d_f=<前方向にある障害物までの距離>  
d_r=<右方向の障害物までの距離>  
d_l=<左方向の障害物までの距離>
```

Rule13: Rule12の入力が与えられたら、Rule1~Rule10のルールに従って、Rule11の出力をしてください。

Rule14: 障害物回避する場合は、停止してはいけません。回転して回避することを推奨します。

ドローン



- 箱庭でドローンを作成してGitHub公開しました

- <https://github.com/toppers/hakoniwa-unity-drone-model>



A screenshot of a Twitter post. On the left, there are icons for likes (17), replies (18), and retweets. The post is from @kanetugu2018 (Mori Takashi) in the TOPPERSプロジェクト, posted on May 28, 2023, with 14,495 views. The main text reads 'Unity + Python + 箱庭で自作ドローンを動かしてみる!' and the hashtags are Python, Unity, TOPPERS, ドローン, 箱庭.

<https://qiita.com/kanetugu2018/items/cacbc0b3632e070fb777>

The screenshot displays the Unity 2021.3.17f1 Personal IDE. The main window is split into several panels:

- Explorer:** Shows the project structure with folders like `workspace`, `dev`, and `ai`. The file `cmd.txt` is selected.
- Code Editor:** Displays the content of `cmd.txt` with the text `1 0.0`.
- Terminal:** Shows the output of a bash shell in the `hakoniwa-base` directory. The log includes:


```

tity,deflate,gzip", "accept-encoding"
}, message: AssetInfo { name: "Unity
: Extensions }
asset_notification_start: Got a requ
ata: MetadataMap { headers: {"te": "
type": "application/grpc", "user-age
37.0-dev grpc-c/15.0.0 (windows; cht
encoding": "identity,deflate,gzip",
identity,gzip"} }, message: AssetInf
t" }, extensions: Extensions }

```
- Hierarchy:** Lists the scene objects: `Hakoniwa*`, `ENV`, `GUI`, `Robot`, and `Drone`.
- Game View:** Shows a 3D simulation of a drone in a simple environment. A `START` button and a `Simulation Time [Sec]: 0.000` display are visible.
- Project/Assets:** Shows the `Assets > Prefab > Robots` folder containing several drone-related prefabs.

The status bar at the bottom indicates the environment is `WSL: Ubuntu-22.04` and the current file is `ai*` at `Ln 1, Col 1`.

