



## Orchestratorによる大規模RPAプロジェクトの 安定稼働：スケーラブルでレジリエントな RPA基盤実現に向けて

UiPath株式会社

プロダクトソリューション本部

プロダクトマネージャ

**ブランドン・カンタン** BRANDON Quentin

## \$ whoami

氏名

BRANDON Quentin (カンタン)

役割

プロダクトマネージャ

経歴

ゲームサーバ開発

→UiPath(社員番号23)

趣味

好きな言語はJS、趣味はカウチポテト



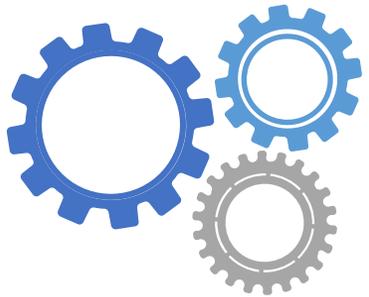
1. Orchestratorの仕組みについて

2. デプロイの事例

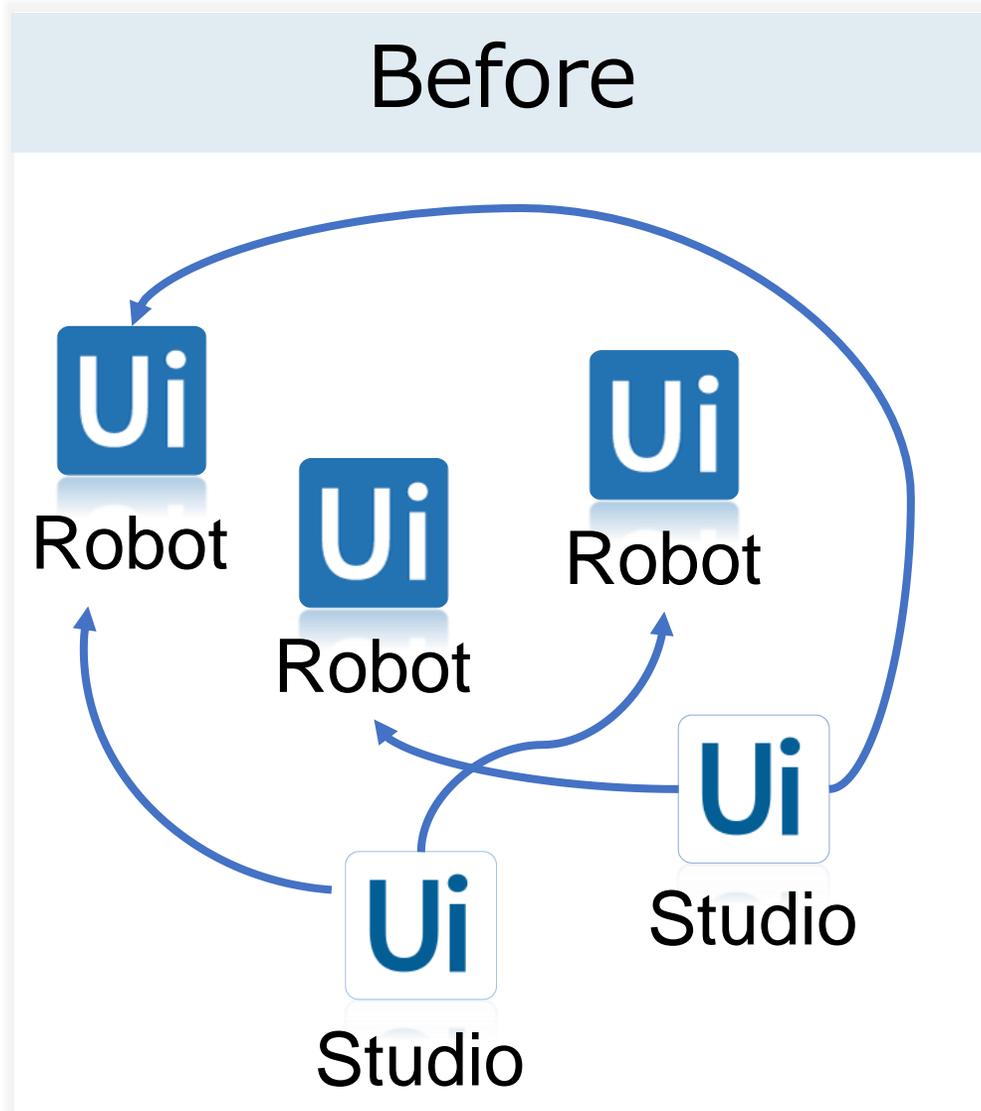
3. 設計・運用について

# 01

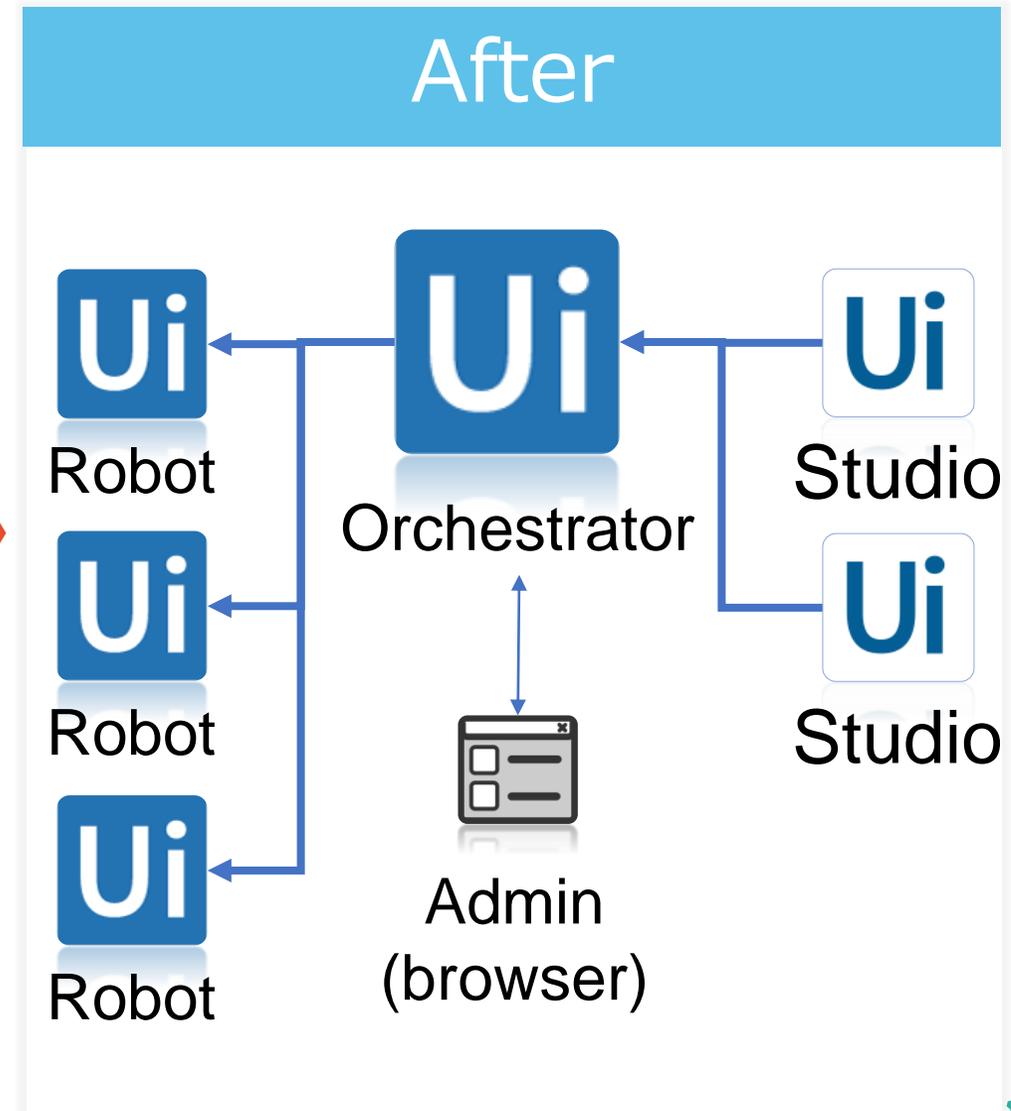
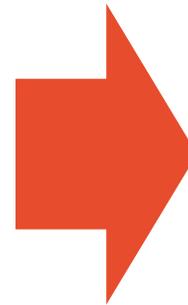
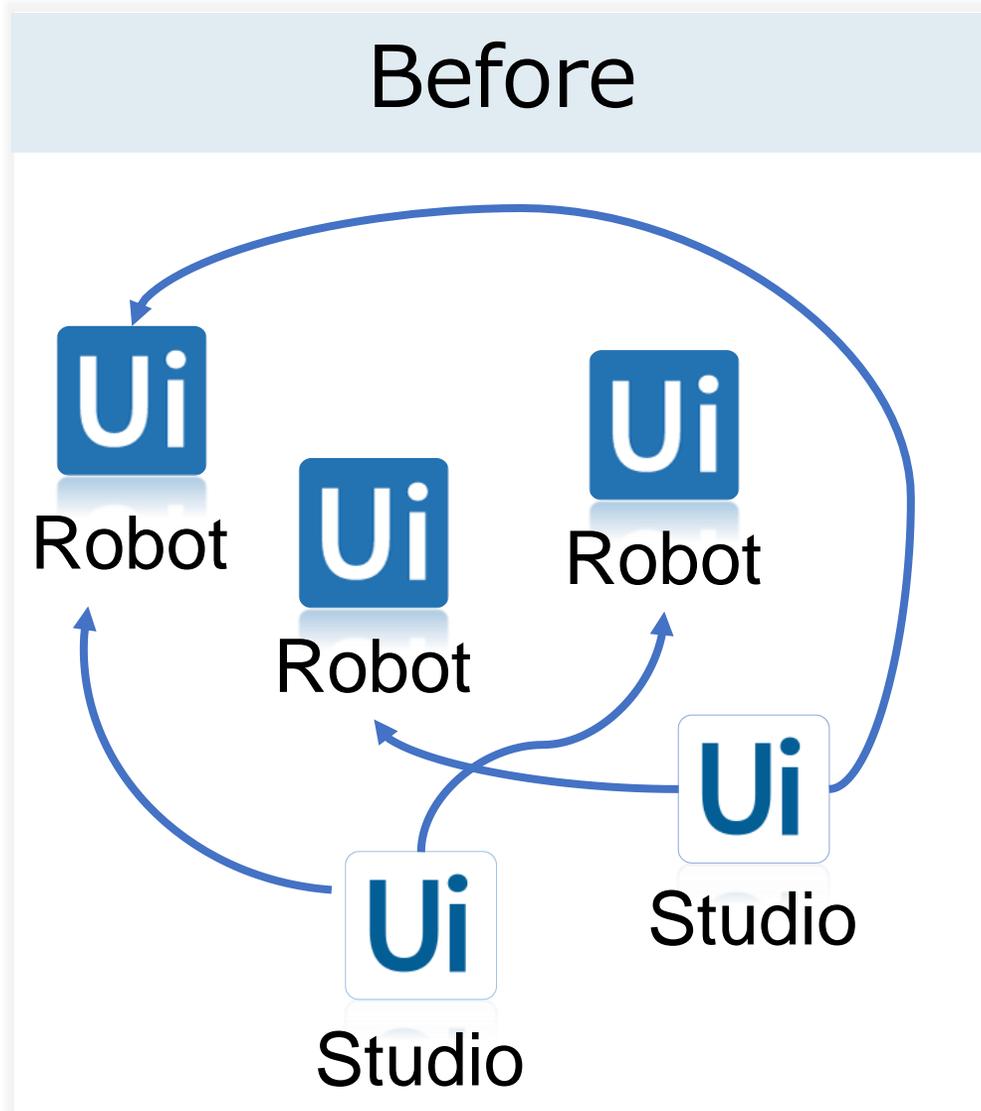
## Orchestratorの仕組みについて



# Why Orchestrator?



# Why Orchestrator?



1. Package\* の集中管理
2. RPA の追加機能 (キュー\*\*、アセットなど)
3. 実行ログ・監査ログ
4. **Unattended ロボット!**

\*PackageはStudioで開発されたワークフローとなります。

\*\*キューはロボットが共有できるトランザクションの仕組み。

## On premise

---

- 標準なインフラで設定可能
- オフライン



## On premise

---

- 標準なインフラで設定可能
- オフライン



## In the Cloud

---

- フレキシブル
- スケーラブル



Orchestrator は WebServer型のバックエンド。

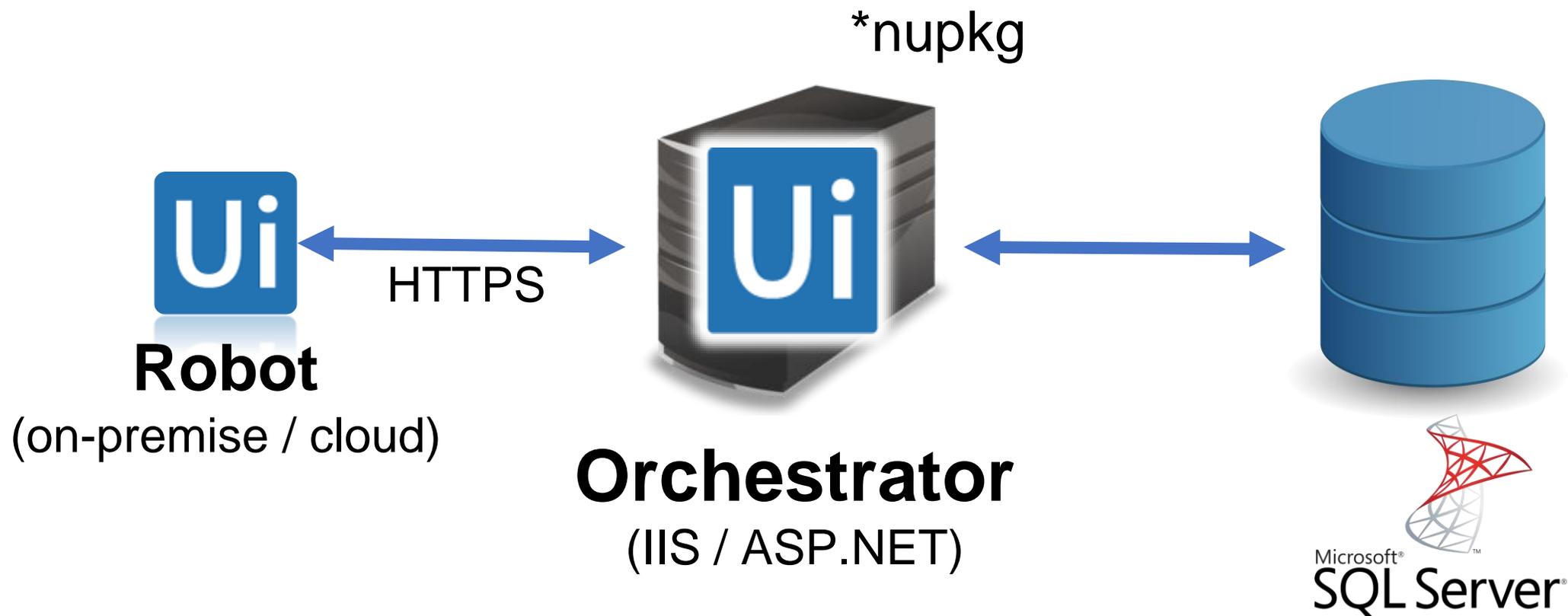


**Web Server**  
(IIS / ASP.NET)

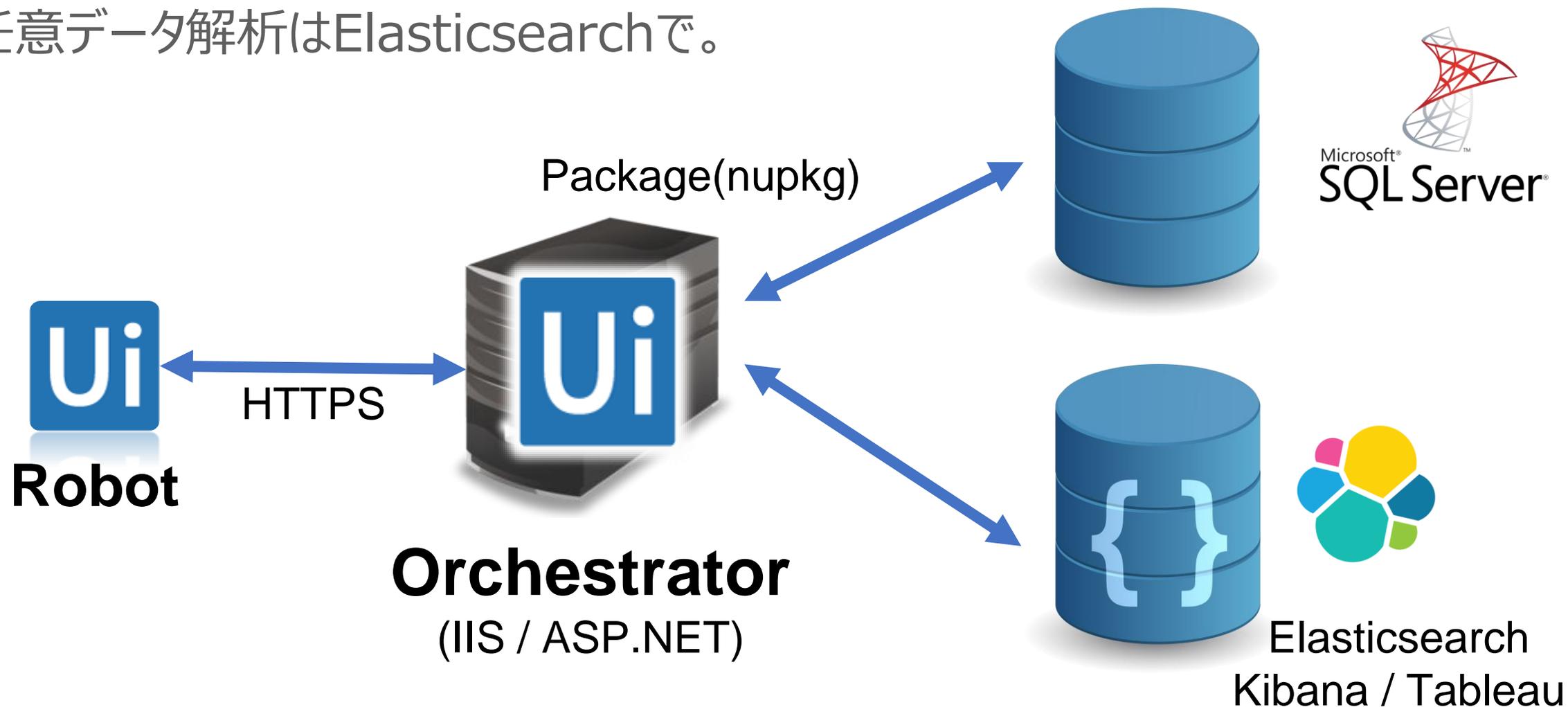


**Database**  
(SQL Server)

Orchestrator は WebServer 形のバックエンド

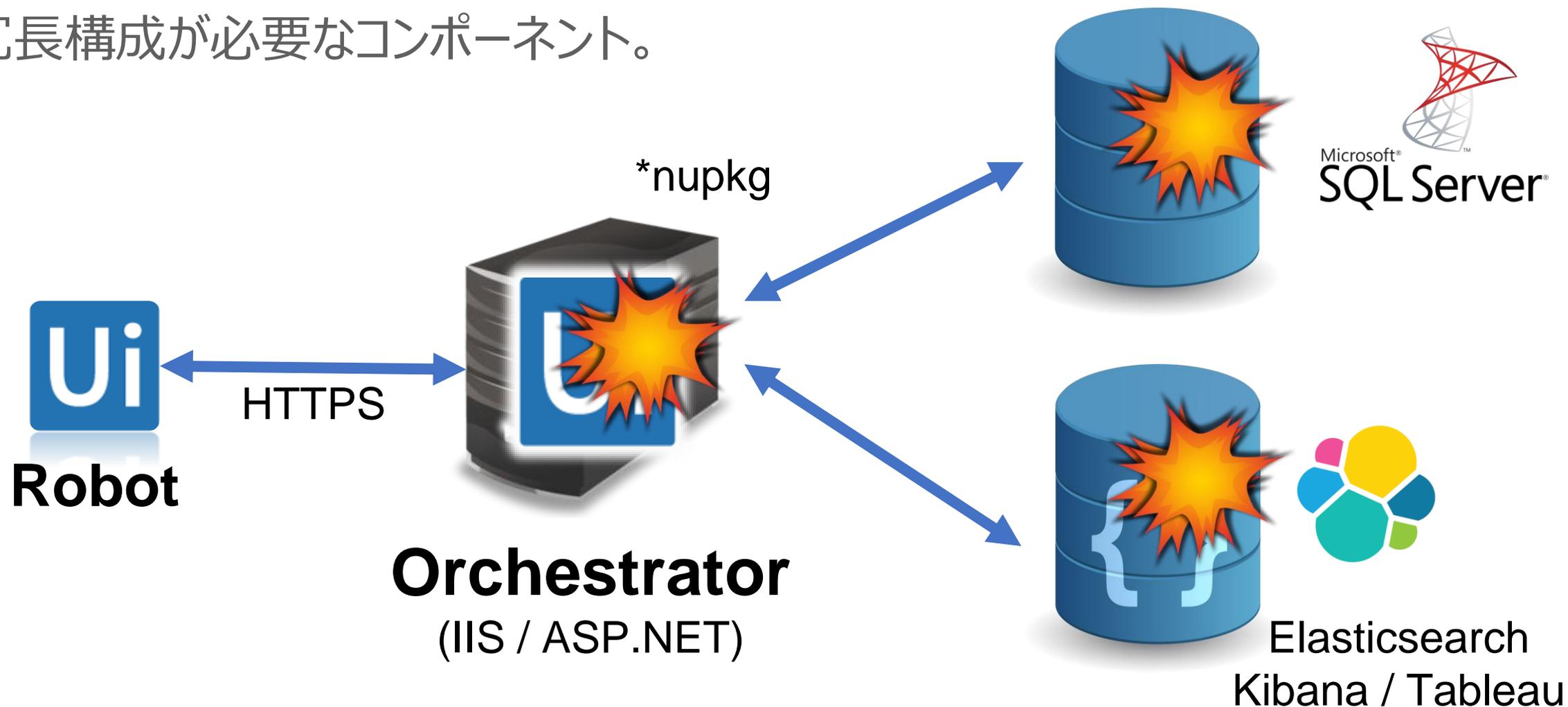


任意データ解析はElasticsearchで。



\*Elasticsearchはログストア・解析NoSQL DBとなります。

冗長構成が必要なコンポーネント。



01

Orchestrator  
Active/  
Standby

02

SQL Server  
高可用性  
クラスター

03

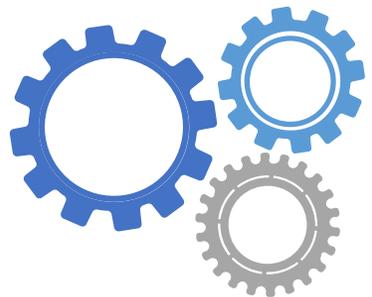
Elasticsearch  
レプリケーション

04

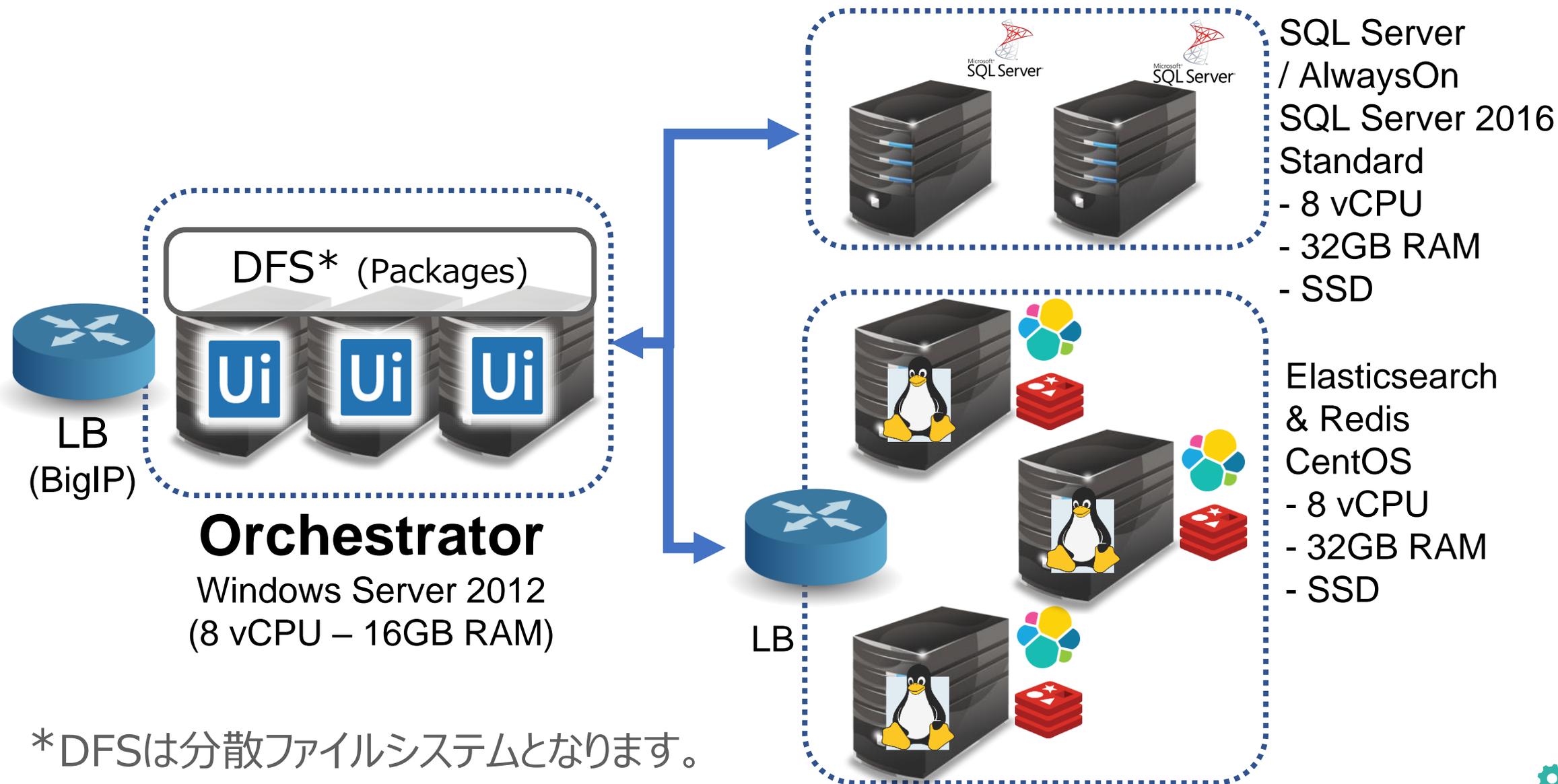
Orchestrator  
スケールアウト

02

## デプロイの事例



# デプロイ例 1 : On premise

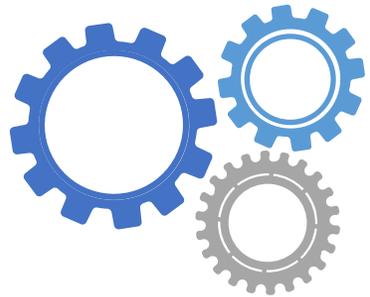


<https://orchestrator.uipath.com/docs/hardware-requirements-orchestrator>

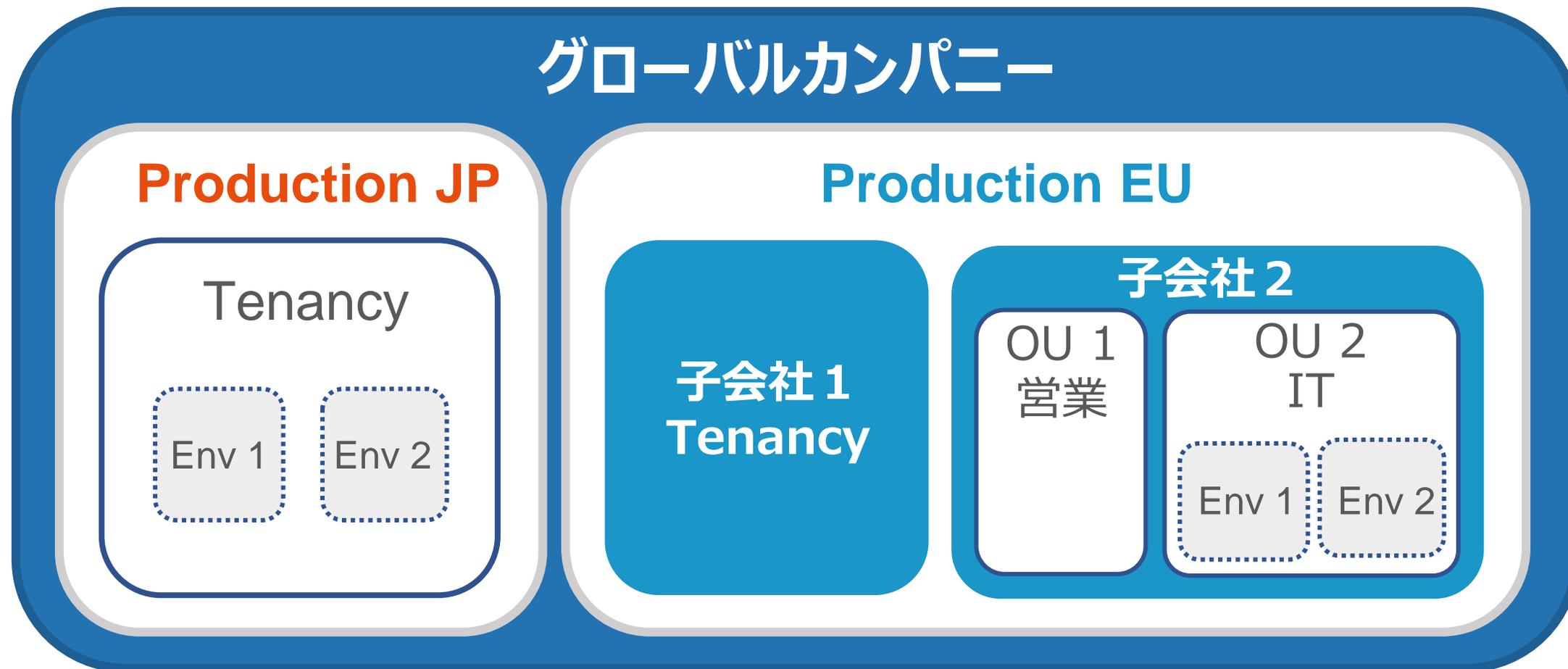


# 03

## 設計・運用について



## Overview



Orchestrator API Reference

<https://orchestrator.uipath.com/reference>

<https://github.com/UiPath/orchestrator-nodejs>

毎月（月末）行います。

[https://www.uipath.com/ja/developer\\_community\\_registration](https://www.uipath.com/ja/developer_community_registration)



**RPAトークと  
ピザやビール  
あります！**



