

# UiPath Orchestrator スターターガイド

2020/05/07

UiPath株式会社

ソリューション本部

セールスコンサルティング第3部



# 免責事項

- 本資料は情報提供及び弊社製品の利用をご検討いただくことを目的とするものです。目的外のご利用はご遠慮ください。
- 本資料に記載されている情報については、弊社では細心の注意を払っておりますが、その正確性や妥当性を保証するものではありません。
- 本資料に記載されている情報は、参考情報として提供されるものであり何らの法的拘束力を有するものではありません。  
最終的な契約条件は、当事者間で有効に締結される契約において確定されます。
- 本資料に関する一切の権利は弊社に帰属します。本資料に関する著作権は、弊社又はその他の権利者に帰属しており、著作権法その他の法令で保護されています。上記の目的を超えて、著作権者及びその他の権利者の許諾を得ることなく、これらの情報を使用(複製、改変、転載などを含みます)することは、著作権法により禁止されております。また、本資料で使用される会社名、製品の商品名及びロゴマークは、弊社、そのグループ会社、又はそれぞれの権利者に帰属している商標又は登録商標です。権利者の許諾を得ることなくこれらを使用することは法令により禁止されておりますので、事前に当社にご連絡の上許諾を得ていただくようお願いいたします。

## 第1章Orchestrator導入にむけて

1. Orchestratorの必要性
2. Orchestratorの種類

## 第2章Orchestrator製品紹介

1. UiPath Platform概要説明
2. Orchestrator導入のメリット
3. Elasticsearch & Kibanaによる実行ログの高度分析
4. Orchestrator主要機能一覧
5. Orchestratorエンティティ説明
6. Orchestrator構成について

## 第3章Orchestratorに関する役立つリンク集

1. 参考リンク

# 第1章

## 1. Orchestratorの 必要性

## 1. Orchestratorの必要性

- 1-1.効果を上げているUiPath導入の姿
- 1-2.UiPath 主要製品ラインアップ
- 1-3.Orchestratorを構成するインフラコンポーネント
- 1-4.Orchestrator導入による効果
- 1-5.Orchestrator導入判断基準
- 1-6.Orchestrator導入検討時期
- 1-7.RPAの導入ステージのステップアップ
- 1-8.ライセンス数に伴うOrchestrator導入構成

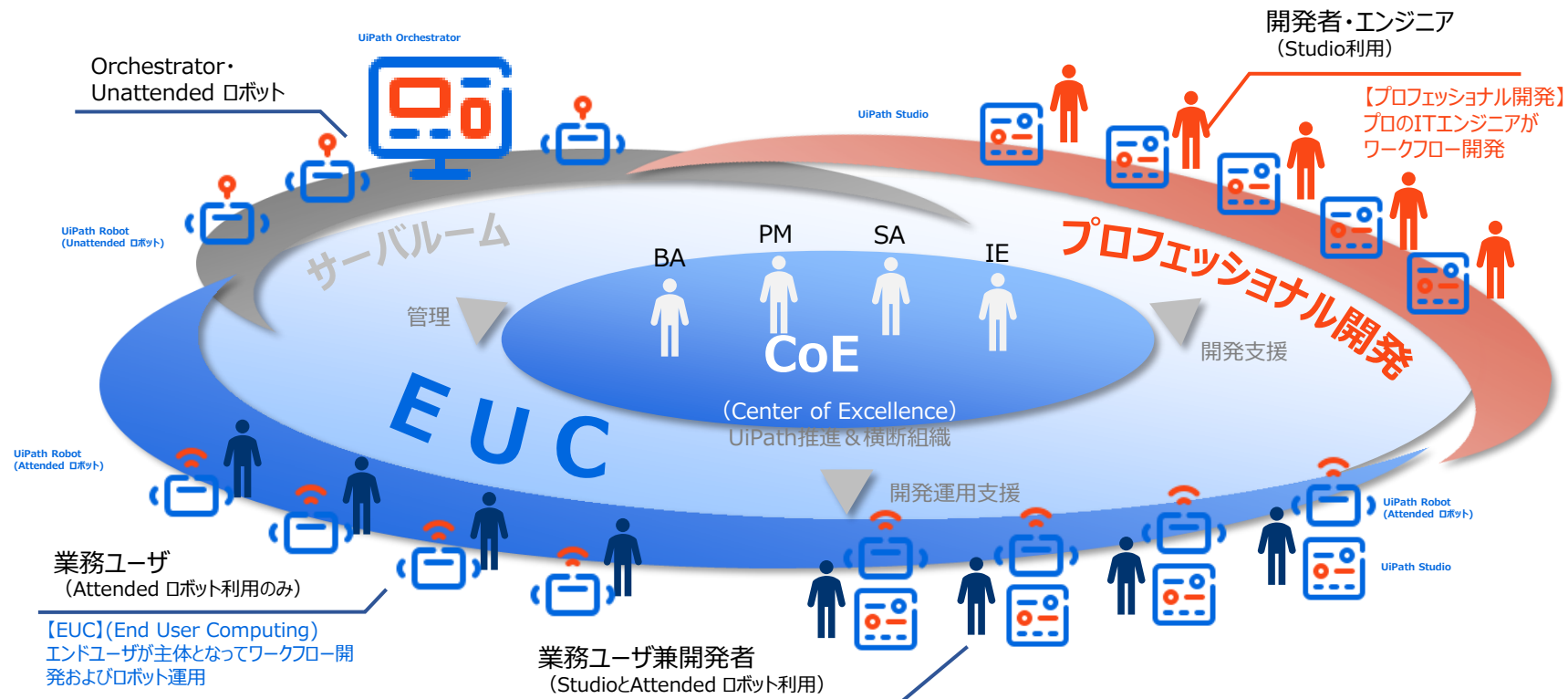
## 2. Orchestratorの種類

- 2-1.規模・導入方式ごとのライセンス選択パターン
- 2-2.StandardとBasic
- 2-3. Cloud版Orchestrator

# 1. Orchestratorの必要性

## 1-1.効果を上げているUiPath導入の姿

- UiPath導入で効果を上げている企業における代表的な共通点
  - COEが確立されている
  - 必要なルールが整備されている
  - 統合管理を行うツール（**=Orchestrator**）とそれを運用保守するメンバーが明確になっている
  - スキルを習得した開発者が十分にいる
  - 開発者以外のユーザーもRPAの概要を理解し、適用領域を考えられている



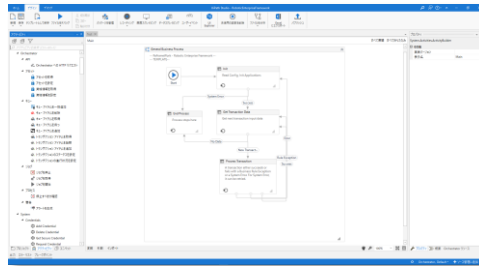
目指すべき効果が出るRPA導入の姿  
(イメージ)

# 1. Orchestratorの必要性

## 1-2.UiPath 主要製品ラインアップ

開発

Studio

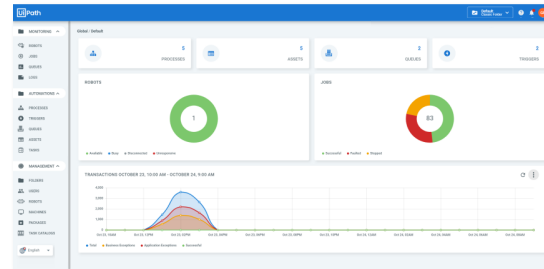


ワークフロー開発ツール

- 優れたコンピュータービジョン
- レコーディング機能
- 複雑な自動化を可能にする  
共通部品やテンプレート

管理

Orchestrator



管理コンソール

- 実行スケジュール管理
- ワークフローの展開とリリース管理
- 実行ログの集中管理
- APIによる外部連携

実行

Robots

Attended  
Robot



Unattended  
Robot



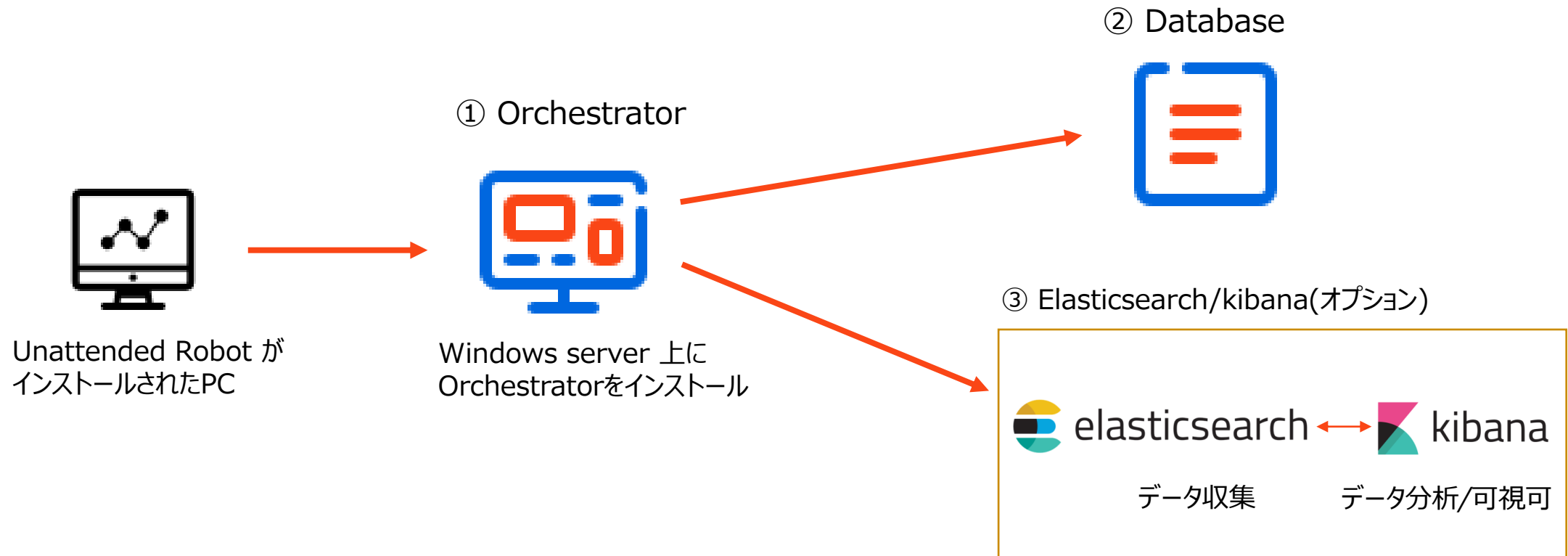
ワークフロー実行ツール

- **Attended Robot**  
ユーザーがデスクトップ上で直接実行を指示するロボット
- **Unattended Robot**  
「スケジュール実行」等の無人実行ができるロボット

# 1. Orchestratorの必要性

## 1-3.Orchestratorを構成するインフラコンポーネント

- ① OrchestratorのWeb/App 部分を担うWindows Server
  - ② OrchestratorのDB部分を担う MS SQL Server
- が最小コンポーネント。①と②の同一サーバー上へのインストールは非推奨。





# 1. Orchestratorの必要性

## 1-4.Orchestrator導入による効果

### Orchestrator導入前

- 1 **改修したワークフローが使われていない**  
アナウンスしたのに、改修前の不具合があるワークフローを使っている…
- 2 **野良ロボ**  
RPAの担当者が把握していないロボットが社内で稼働しているらしい…
- 3 **ログが取れていない/散在している**  
問い合わせ対応やトラブルシューティングに、必要以上に工数が掛かっている…
- 4 **セキュリティ**  
ワークフローの中にユーザー名/パスワードが埋め込まれているが、これって大丈夫…
- 5 **監査対応**  
重要なデータが書き換えられたが、ロボットがやったのか人がやったのかが後から分からない…
- 6 **効率が上がらない**  
スケジュール実行ができないと、夜間や土日など端末の空き時間を効率的に使えない…
- 7 **処理速度が遅い**  
業務量が多すぎて1台のロボットでは非常に時間がかかってしまう…
- 8 **業務が複数部署にまたがる**  
ロボットが異なるため、後続部署にデータを渡すところが人手になってしまっている…
- 9 **ロボットを増やすのが大変**  
正しいバージョンのワークフローを適切なロボットへ渡して設定しないとイケない…
- 10 **全社で効率的なロボット運用ができていない**  
閑散期の部署でも固定数のロボットが割り当てられてしまっている…

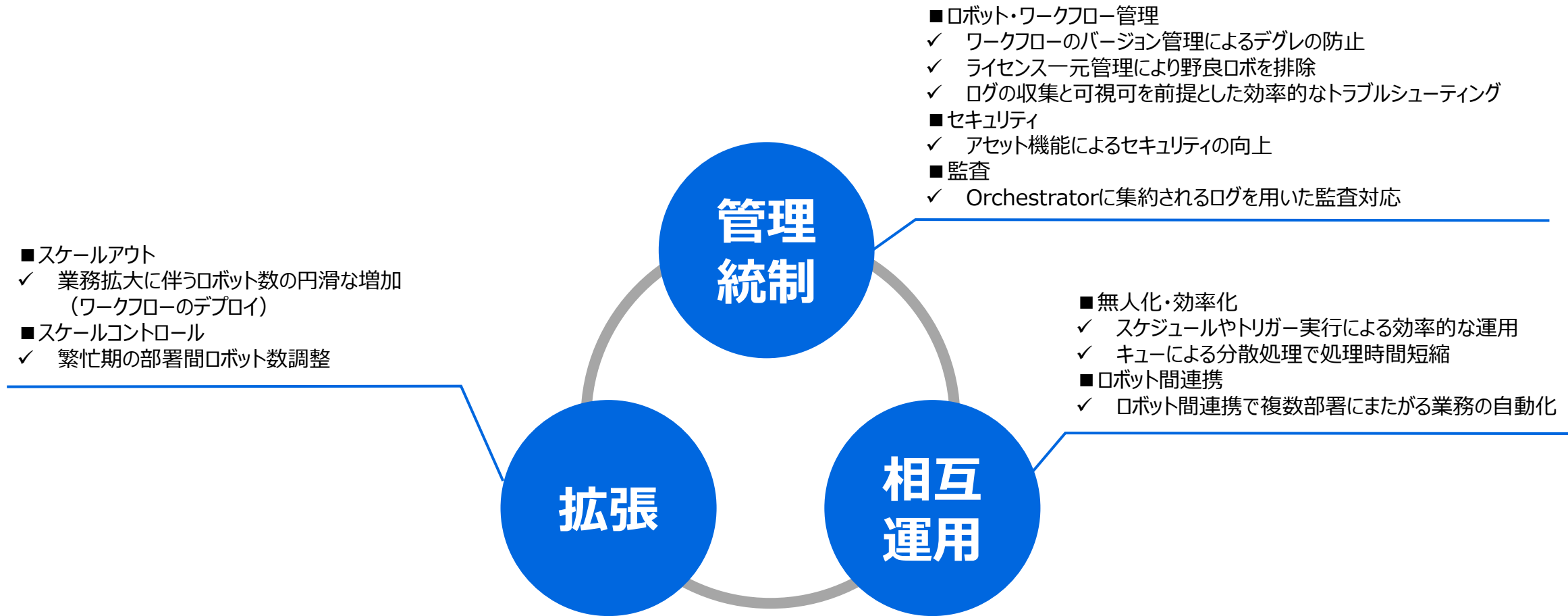
### Orchestrator導入後

- ✓ **ワークフローのバージョン管理によるデグレの防止**
- ✓ **ライセンス一元管理により野良ロボを排除**
- ✓ **ログの収集と可視可を前提とした効率的なトラブルシューティング**
- ✓ **アセット機能によるセキュリティの向上**
- ✓ **Orchestratorに集約されるログを用いた監査対応**
- ✓ **スケジュールやトリガー実行による効率的な運用**
- ✓ **キューによる分散処理で処理時間短縮**
- ✓ **ロボット間連携で複数部署にまたがる業務の自動化**
- ✓ **業務拡大に伴うロボット数の円滑な増加（ワークフローデプロイ）**
- ✓ **繁忙期の部署間ロボット数調整**

# 1. Orchestratorの必要性

## 1-5.Orchestrator導入判断基準

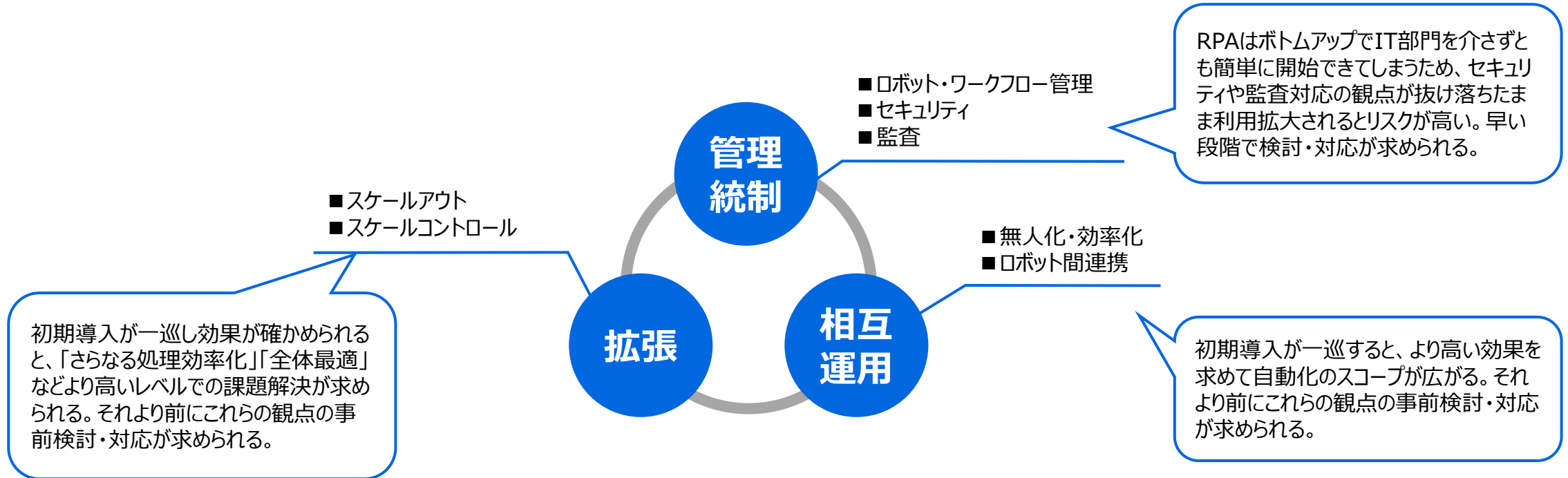
- 前頁の効果を整理すると、Orchestrator導入の判断基準は以下 3 点に集約される



# 1. Orchestratorの必要性

## 1-6.Orchestrator導入検討時期

- Orchestrator導入の観点求められる時期を示す



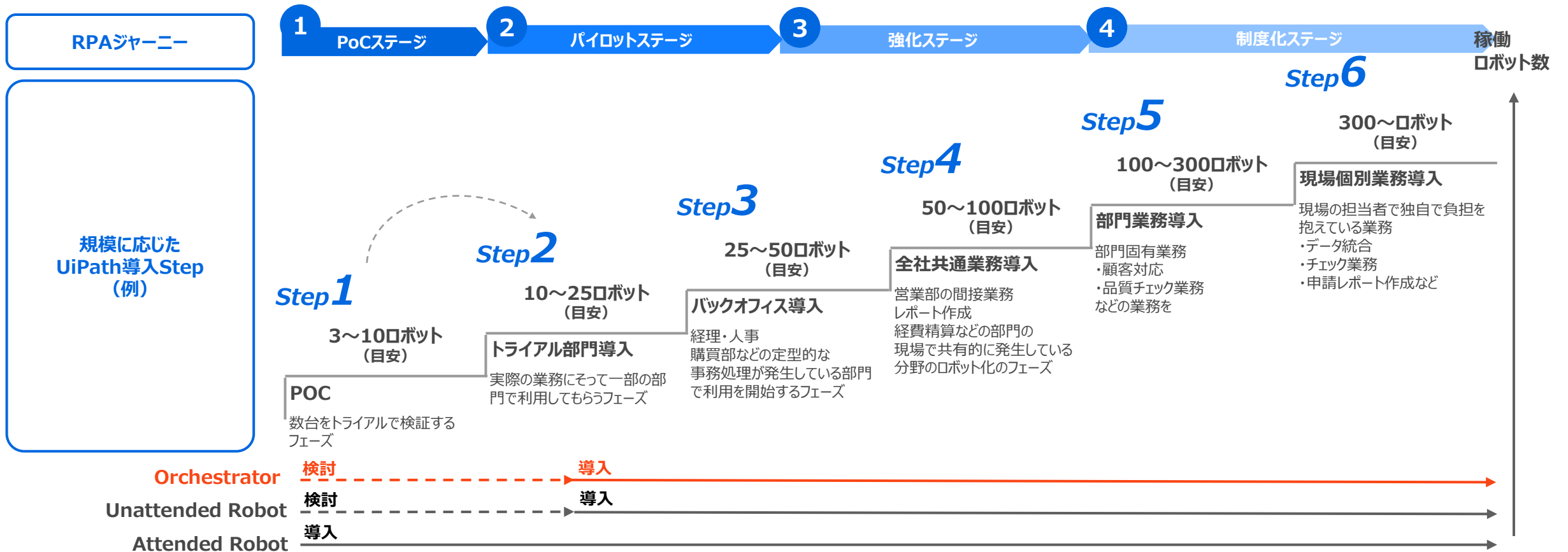
各観点を後から対応しようとする、ワークフローの作りを見直したり、問題発生対応と同時に遂行することを求められたりして対応負荷が増す可能性がある。

**できるだけ早い時期にOrchestratorを含むRPA基盤の検討・構築が重要である。**

# 1. Orchestratorの必要性

## 1-7.RPAの導入ステージのステップアップ

RPAジャーニーでは以下のステップでRPAの導入エリアを拡張させていくことが可能です  
(以下モデルケースとしての導入拡張例)



# 1. Orchestratorの必要性

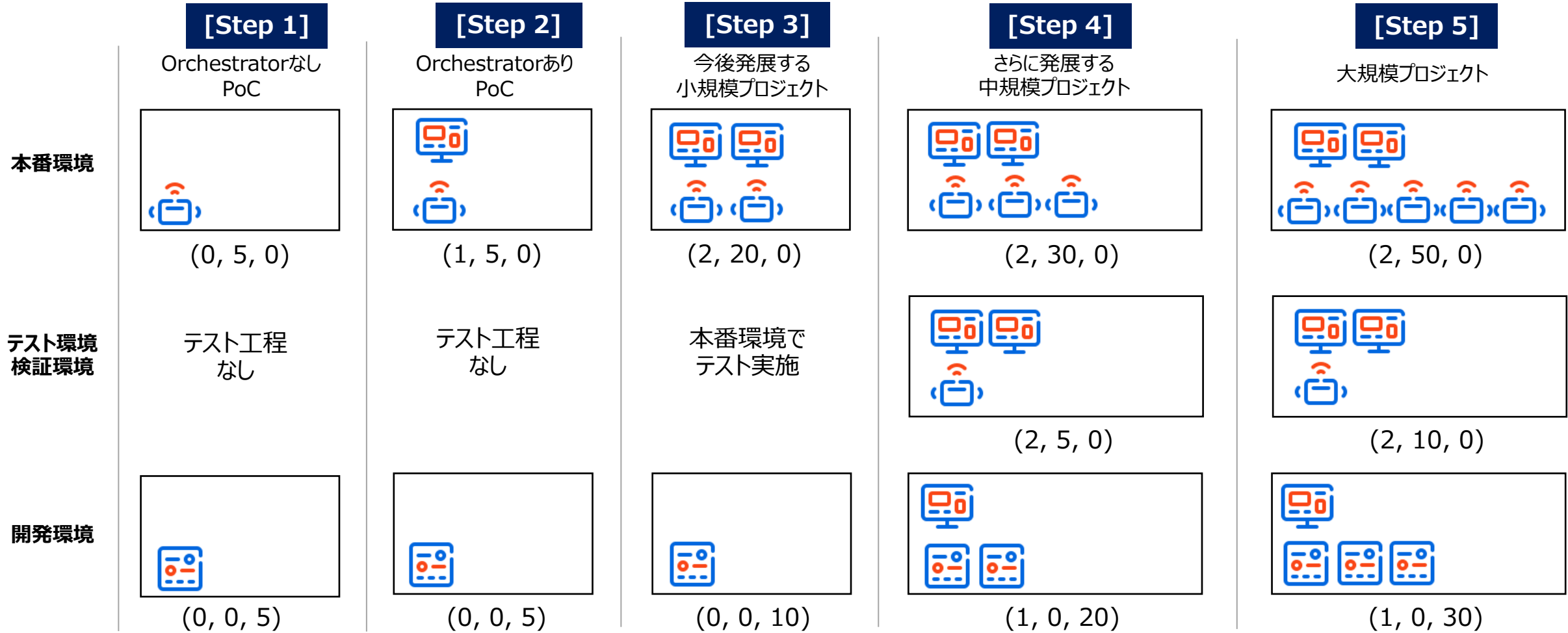
## 1-8.ライセンス数に伴うOrchestrator導入構成

※この図はUiPathの考える典型的な購入数の例を示しており、実際の購入数はお客様のプロジェクト要件ごとに異なります。

【凡例】



ライセンス数表記:(Orchestrator,Robot,Studio)



# 第1章

## 2. Orchestratorの 種類

## 1. Orchestratorの必要性

- 1-1.効果を上げているUiPath導入の姿
- 1-2.UiPath 主要製品ラインアップ
- 1-3.Orchestratorを構成するインフラコンポーネント
- 1-4.Orchestrator導入による効果
- 1-5.Orchestrator導入判断基準
- 1-6.Orchestrator導入検討時期
- 1-7.RPAの導入ステージのステップアップ
- 1-8.ライセンス数に伴うOrchestrator導入構成

## 2. Orchestratorの種類

- 2-1.規模・導入方式ごとのライセンス選択パターン
- 2-2.StandardとBasic
- 2-3.Cloud版Orchestrator

## 2. Orchestratorの種類

### 2-1. 規模・導入方式ごとのライセンス選択パターン

Orchestrator導入方式		Orchestratorの種類	
サービス利用 (SaaS)	SaaS	Cloud版Orchestrator	
自社構築 (クラウド、 オンプレミス)	クラウド	Orchestrator Basic ※50ロボットまで接続可能	Orchestrator Standard
	オンプレミス		



## 2. Orchestratorの種類

### 2-2.StandardとBasic

- UiPath Orchestrator Basicとは  
Orchestratorの入門的な位置づけのライセンス  
Orchestrator Standardとの機能的な差違は無いが、管理できるRobot/Studioの数や、テナント分割の可否が異なる

製品名	Orchestrator Basic	Orchestrator Standard
対象導入規模	エントリー	エンタープライズ
Robotの最大接続可能数	5 (Attended Robot, Unattended Robotの合算で5)	制限なし
Studioの最大接続可能数	5	制限なし
最大テナント数	1 (本番用で1、非本番用は数に含まず)	制限なし
Orchestrator Standardへのアップグレード	ライセンス期間中でもアップグレード用日割金額のライセンス購入で切替可能	-

## 2. Orchestratorの種類

### 2-3.Cloud版Orchestrator

- Cloud版Orchestratorの代表的な機能

分類	機能概要
スケジュール管理	Unattended Robotを使用したスケジュール実行
統合運用管理	ライセンスの集中管理
	パッケージのバージョン管理
	実行結果/ログの集中管理
ガバナンス	ユーザーおよび組織ごとのアクセスコントロール
	監査証跡の記録/管理
	パスワードの集中管理

## 2. Orchestratorの種類

### 2-3.Cloud版Orchestrator

- 一部の機能はCloud版Orchestratorに適した形で将来的な提供を検討中

カテゴリ	機能
設定の柔軟性	資格情報ストアのプラグイン
	Orchestrator実行ログのターゲット設定
運用保守補助	ジョブ失敗時のレコーディング
外部連携	Azure AD連携
	Elasticsearch/Kibana
UiPath製品統合	Insights

- 最新の対応状況は下記ページに掲載  
<https://docs.uipath.com/cloudplatform/lang-ja/docs/on-premises-vs-cloud-platform-orchestrator-features>

# 第2章




## 1. UiPath Platform概要説明



- 1. UiPath Platform 概要説明**
  - 1-1. 本資料で使用する用語
  - 1-2. UiPath Platform の構成
- 2. Orchestrator 導入のメリット**
  - 2-1. Orchestrator 導入のメリット一覧
  - 2-2. 効率的なパッケージのデプロイ
  - 2-3. 効率的なライブラリの配布
  - 2-4. 効率的・セキュアなアセットの管理
  - 2-5. 効率的なロボット、マシン、パッケージの管理
  - 2-6. 効率的なライセンス配分の運用
  - 2-7. スケジューリング実行
  - 2-8. 分散処理実行
  - 2-9. 外部システムからの実行
  - 2-10. 実行ログの統合管理
  - 2-11. サマリー
- 3. Elasticsearch & Kibana による実行ログの高度分析**
  - 3-1. Elasticsearch & Kibana とは
  - 3-2. Elasticsearch & Kibana での実行ログ集約分析
  - 3-3. ユースケース
  - 3-4. 『ログフィールドを追加』アクティビティによるログ拡張
- 4. Orchestrator 主要機能一覧**
  - 4-1. Orchestrator の主要機能一覧
- 5. Orchestrator エンティティ説明**
  - 5-1. Orchestrator の主要エンティティ一覧
  - 5-2. マシン
  - 5-3. ロボット
  - 5-4. ロボットグループ
  - 5-5. パッケージ
  - 5-6. プロセス
  - 5-7. ジョブ
  - 5-8. 時間トリガー
  - 5-9. キュートリガー
  - 5-10. キュー
  - 5-11. アセット
  - 5-12. ライブラリ
- 6. Orchestrator 構成について**
  - 6-1. Orchestrator の基本構成要素
  - 6-2. 冗長構成のイメージ
  - 6-3. Orchestrator クラウド (AWS) 構成とそのメリット
  - 6-4. AWS 構成時のネットワークについて

# 1. UiPath Platform概要説明

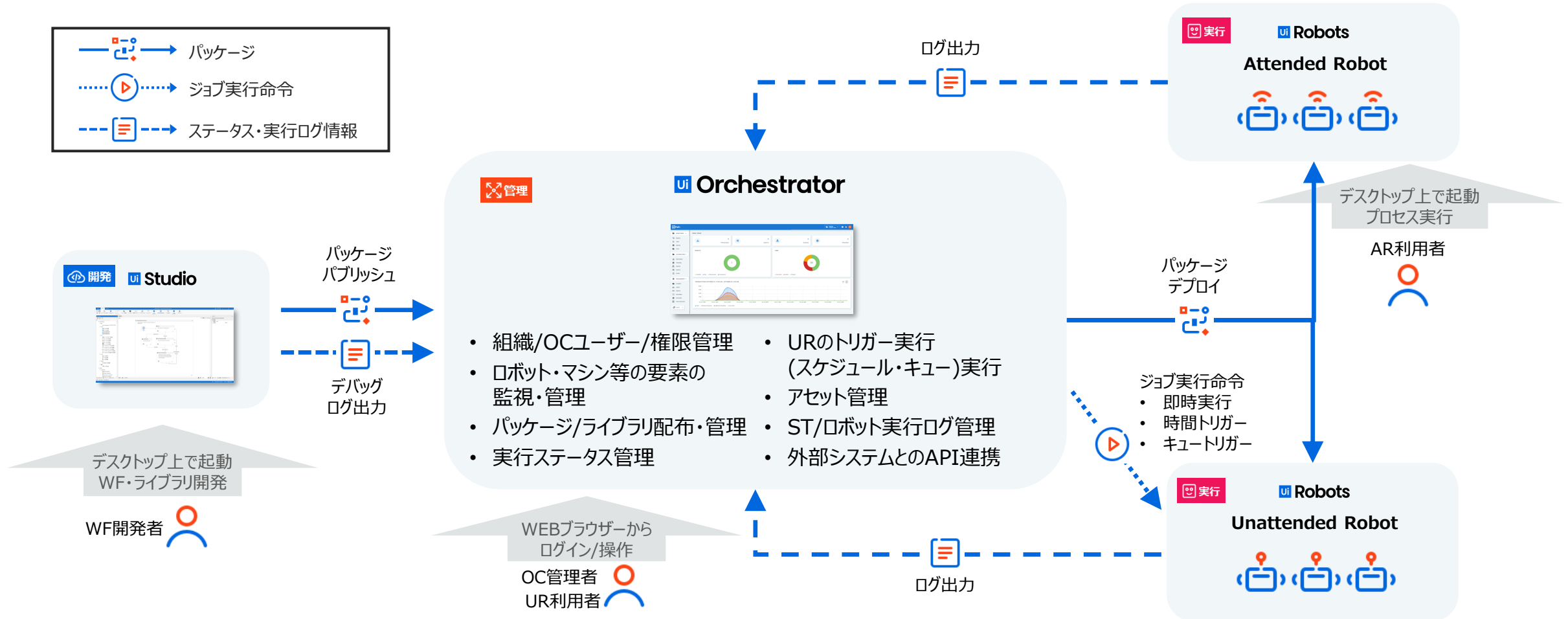
## 1-1.本資料で使用する用語

用語	略語	説明
 <b>Orchestrator</b>	OC	UiPath Robot の統合管理を行うためのWebアプリケーション。 Robot が参照するリソースの管理や動作状況の監視、実行環境への Robot のデプロイなどを行う。
 <b>Studio</b>	ST	Robot が実行する自動化処理（Workflow）を作成するための統合開発環境（IDE）。
 <b>Robots</b>	Robot / ロボット	UiPath Studio で作成された Workflow に従い、コンピュータ上でプロセス（自動化処理）を実行するソフトウェア。
Attended Robot	AR	Robot 種別の一つ。Orchestrator による起動ではなく、ユーザー（人手）により起動されるタイプの Robot。
Unattended Robot	UR	Robot 種別の一つ。人手を介さず、Orchestrator から起動される Robotを表します。
Queue または キュー	(N/A)	Robot が Orchestrator に接続されている場合に利用可能。無制限に項目を保持できる Orchestrator の収納機能であり、Queue へのアイテムの設定やステータスの変更、処理を行う場合は、専用のアクティビティを使用する。
Asset または アセット	(N/A)	Robot が Orchestrator に接続されている場合に利用可能。Orchestrator で管理される各自動化プロジェクトで使用可能な共有変数または認証情報を表す。Asset を作成し特定の情報を格納することで、Robot は容易にその情報へアクセス可能。
Process または プロセス	(N/A)	自動化処理実行時インスタンス。
Workflow または ワークフロー	WF	自動化処理の流れを GUI で表現・定義したものです。

# 1. UiPath Platform概要説明

## 1-2.UiPath Platformの構成

- Orchestratorを導入した場合の簡易構成図は以下の通り



# 第2章

## 2. Orchestrator導入のメリット



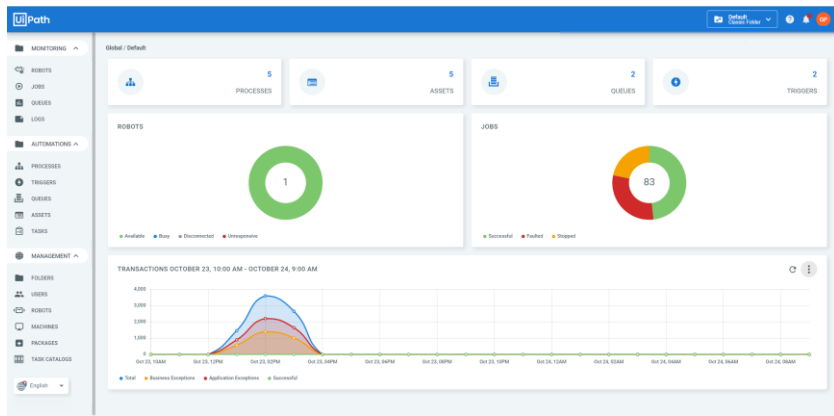
- 1. UiPath Platform 概要説明**
  - 1-1. 本資料で使用する用語
  - 1-2. UiPath Platform の構成
- 2. Orchestrator 導入のメリット**
  - 2-1. Orchestrator 導入のメリット一覧
  - 2-2. 効率的なパッケージのデプロイ
  - 2-3. 効率的なライブラリの配布
  - 2-4. 効率的・セキュアなアセットの管理
  - 2-5. 効率的なロボット、マシン、パッケージの管理
  - 2-6. 効率的なライセンス配分の運用
  - 2-7. スケジューリング実行
  - 2-8. 分散処理実行
  - 2-9. 外部システムからの実行
  - 2-10. 実行ログの統合管理
  - 2-11. サマリー
- 3. Elasticsearch & Kibana による実行ログの高度分析**
  - 3-1. Elasticsearch & Kibana とは
  - 3-2. Elasticsearch & Kibana での実行ログ集約分析
  - 3-3. ユースケース
  - 3-4. 『ログフィールドを追加』アクティビティによるログ拡張
- 4. Orchestrator 主要機能一覧**
  - 4-1. Orchestrator の主要機能一覧
- 5. Orchestrator エンティティ説明**
  - 5-1. Orchestrator の主要エンティティ一覧
  - 5-2. マシン
  - 5-3. ロボット
  - 5-4. ロボットグループ
  - 5-5. パッケージ
  - 5-6. プロセス
  - 5-7. ジョブ
  - 5-8. 時間トリガー
  - 5-9. キュートリガー
  - 5-10. キュー
  - 5-11. アセット
  - 5-12. ライブラリ
- 6. Orchestrator 構成について**
  - 6-1. Orchestrator の基本構成要素
  - 6-2. 冗長構成のイメージ
  - 6-3. Orchestrator クラウド (AWS) 構成とそのメリット
  - 6-4. AWS 構成時のネットワークについて

# 2. Orchestrator導入のメリット

## 2-1.Orchestrator導入のメリット一覧

Orchestratorを導入すると各フェーズにて以下のようなメリットを享受することができます

# Ui Orchestrator



開発	効率的なパッケージのデプロイ
	効率的なライブラリの配布
	効率的・セキュアなアセットの管理
管理	効率的なライセンス配分・管理
	効率的なロボット、マシン、パッケージの管理
実行	スケジューリング実行
	分散処理実行
	外部システムからの実行
	実行ログの統合管理・分析

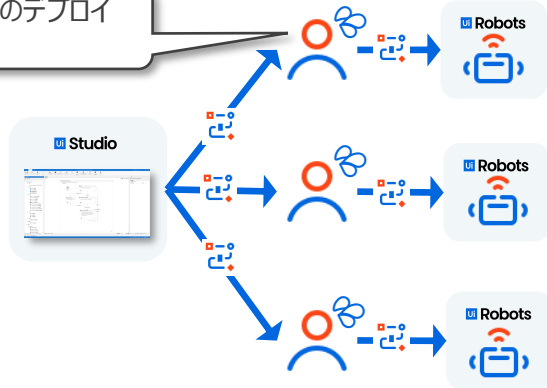
## 2. Orchestrator導入のメリット

### 2-2. 効率的なパッケージのデプロイ

時間のかかる『パッケージのデプロイ』手順が、Orchestrator導入に効率的に！

#### Orchestrator導入前

パッケージのデプロイ  
大変



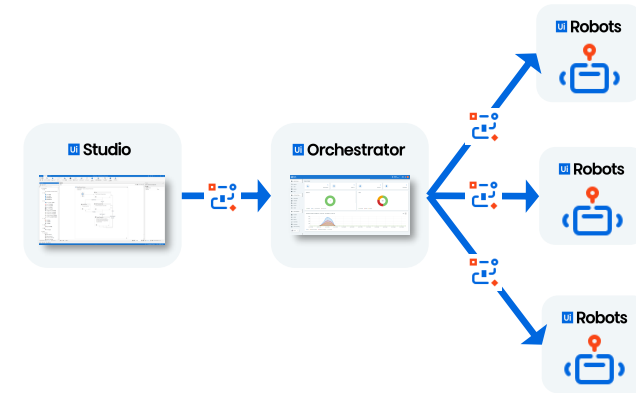
STでパブリッシュしたパッケージを、ユーザーが手動でロボットがインストールされた各マシンにパッケージを配置

#### 問題点

- 運用規定外のパッケージのデプロイの検知不能
- デプロイ先のロボット数の分だけ手動配置する必要有
- パッケージのアップデートをする度に上記処理を行う必要有

Orchestrator  
導入

#### Orchestrator導入後



STからOCへ直接パブリッシュ、OCから任意のRobotに一括デプロイが可能に！

#### 改善点

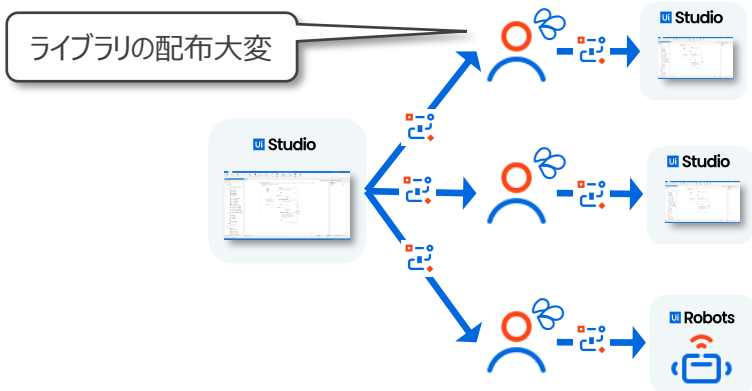
- OCの制御によって規定に沿ったパブリッシュ/デプロイが可能
- 大量のロボット数に一括でデプロイが可能
- パッケージの一括アップデートが可能

## 2. Orchestrator導入のメリット

### 2-3. 効率的なライブラリの配布

時間のかかる『ライブラリの配布』手順が、Orchestrator導入に効率的に！

#### Orchestrator導入前



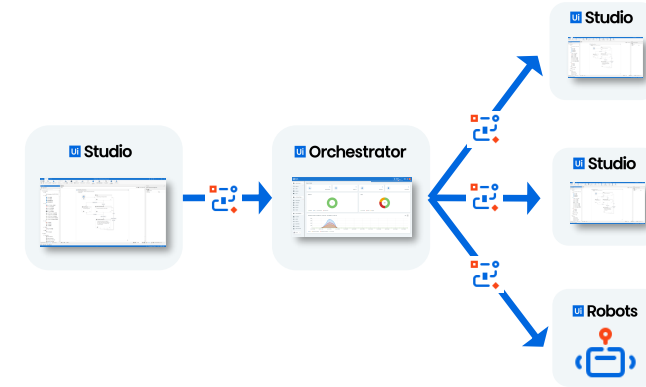
STでパブリッシュしたライブラリを、ユーザーが手動でSTがインストールされた各マシンにパッケージを配置

#### 問題点

- 配布先のStudio/ロボット数の分だけ手動配置する必要有
- 実行時にWFで使用するライブラリパッケージがロボット端末上に存在しない場合、実行不能となる

#### Orchestrator 導入

#### Orchestrator導入後



STからOCへ直接パブリッシュ、STの「パッケージを管理」から直接インストール可能！

#### 改善点

- STの「パッケージを管理」から直接インストール可能
- 実行時にWFで使用するライブラリパッケージがロボット端末上に存在しない場合、OCから自動で取得してからWFを実行

## 2. Orchestrator導入のメリット

### 2-4. 効率的・セキュアなアセットの管理

異なる自動化プロジェクト間で使用可能な共有変数・認証情報を保存できる『アセット』が利用可能に！

#### Orchestrator導入前



異なるWF内で「システムURL」などの同じような文字列が利用されている場合、その文字列に変更があった場合全てのWFを修正しなければならない。

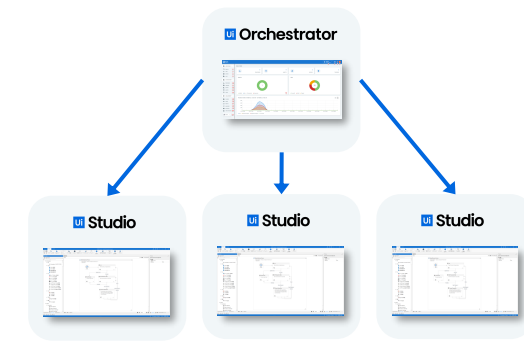
また資格情報などの機密性の高い情報をワークフロー内にベタ書きするのは内部統制上あまりふさわしくない。

#### 問題点

- 複数のWF内で使用する可能性のある『サイトURL』などの文字列に変更があると全てのWFの編集が必要
- 『パスワードを含む各種システムのログイン情報』などを、ガバナンスを保ちながら管理することが難しい。

#### Orchestrator 導入

#### Orchestrator導入後



異なる自動化プロジェクト間で使用可能な共有変数・認証情報を保存できるアセットで利用可能に！

#### 改善点

- 複数のWF内で使用する可能性のある『サイトURL』などの文字列を一元管理可能
- 『パスワードを含む各種システムのログイン情報』など、機密性の高い情報もセキュアに一元管理可能

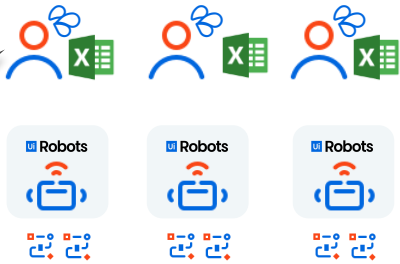
## 2. Orchestrator導入のメリット

### 2-5. 効率的なロボット、マシン、パッケージの管理

煩雑な『ロボット、パッケージ等の管理』が、Orchestrator導入により効率的に運用可能に！

#### Orchestrator導入前

Excelでの  
管理大変。。



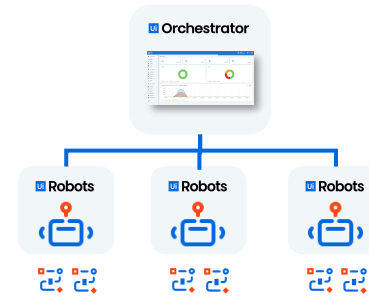
「どのマシンにどのワークフローが実行できるロボットがインストールされているか」を管理するためには、管理台帳Excelファイルなどを随時手動で更新しながら管理をするなど煩雑な作業が必要

#### 問題点

- 台帳の記入漏れの可能性有
- RPA運用規定外の操作の防止不能
- リアルタイム監視は不可能
- 管理対象の要素が増えるとさらに管理が難しく

Orchestrator  
導入

#### Orchestrator導入後



OC画面にて各種ロボット、マシンなどのリアルタイムの状態を確認可能に！

#### 改善点


- 管理のための台帳記入などの業務が不要
- OCの制御によって規定に沿ったRPA運用が可能
- 管理対象要素のリアルタイムの監視が可能
- 管理対象要素の数に関係なく統制の効いた管理が可能

## 2. Orchestrator導入のメリット

### 2-6. 効率的なライセンス配分の運用

各部署の閑散期・繁忙期に合わせた効率的なライセンス配分が可能に！

#### Orchestrator導入前

各端末でのディアクティベート・アクティベート作業大変。。。 

##### 経理部



##### 人事部



##### 総務部



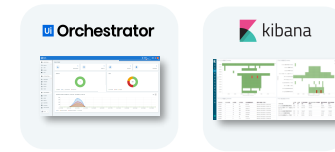
各部門でロボットがどのくらい使われているのかは現場の人にヒアリングしたりするしか調査のしようがなく、適切なライセンス配分をするための情報を得るのが難しい。また部門間（PC端末間）でのライセンスの以降を行う時も、各端末上でディアクティベート、アクティベート作業を行わなければならない。

#### 問題点

- 各ST・ロボットの使用状況の監視不能
- 部署ごとのST/ロボットライセンス数の配分・変更は全て各端末上で手作業で行う必要有

Orchestrator  
導入

#### Orchestrator導入後



##### 経理部



##### 人事部



##### 総務部



各部門のリアルタイム・過去の利用状況はOC画面やKibana\*画面から分析可能。各部署のロボットの有効化・無効化の操作をOC画面上で完結できるため、ライセンスの有効活用が効率的に行える！

#### 改善点

- リアルタイム&過去のST・ロボットの使用状況が監視可能
- 部署ごとのST/ロボットライセンス数の配分・変更をOC上の操作のみで完結可能

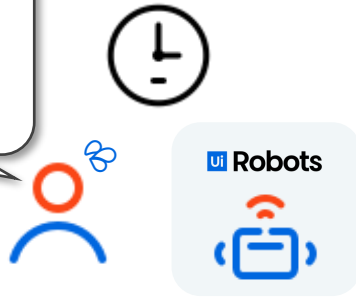
## 2. Orchestrator導入のメリット

### 2-7.スケジューリング実行

設定した時間帯に定期的にプロセスを実行可能！人間の操作を介さない完全自動化が可能に！

#### Orchestrator導入前

- 実行ボタン押す
- PC前での監視  
大変。。。



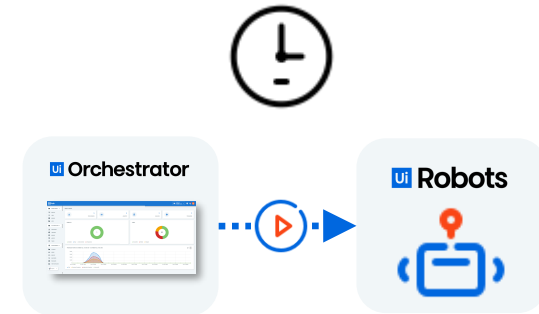
定期的に行うことがわかっているロボットでも、ユーザーがPC端末のデスクトップからロボットトレイから実行する必要がある

#### 問題点

- 人間による実行忘れによる統制上の問題
- ロボットが他プロセスを実行中は終了を待たなければならない
- ユーザーの実行操作が必要 かつ 実行中にユーザーはPC前で監視する必要有

Orchestrator  
導入

#### Orchestrator導入後



人間の操作を介さない完全自動化が可能に！

#### 改善点

- 人間による実行忘れによる統制上の問題を防止
- スケジュールの時間が重複した場合でも、前のプロセス完了するまで保留がし全てのプロセスの完了を保証
- 人間の操作を介さないプロセスの完全自動化実行が可能

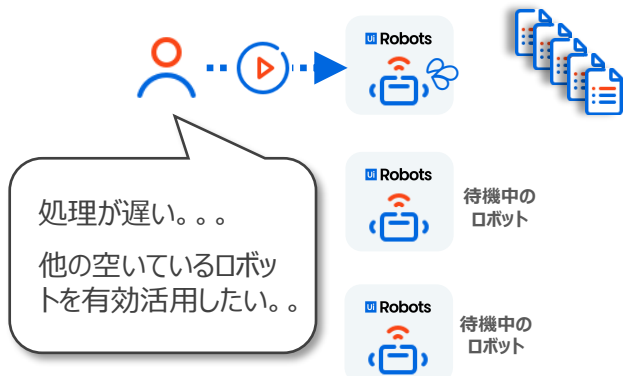


# 2. Orchestrator導入のメリット

## 2-8.分散処理実行

『キュー』を使った分散処理実行により効率的なロボット運用と自動化プロセス完了の高速化を実現！

### Orchestrator導入前



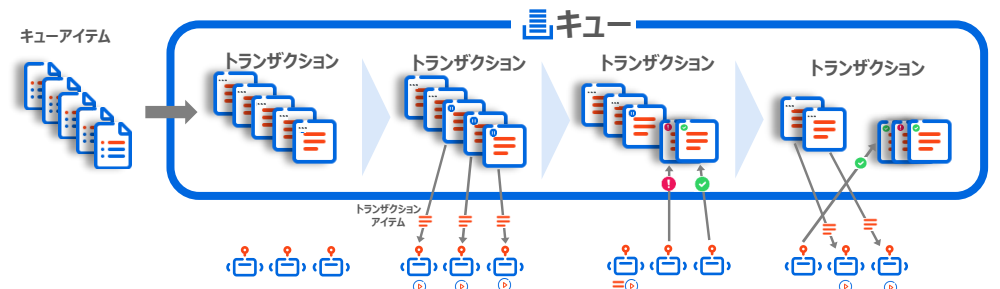
大量のデータを扱うようなワークフローでも一つのロボットに任せる必要があったため、非常に長い時間がかかります。

#### 問題点

- 実行ロボットに問題が起きた時のリカバリが難しい
- ワークフロー内の問題の発見が遅延する可能性
- 待機中の空きロボットを有効活用できない
- 1台のロボットの負荷が高くなってしまふ

Orchestrator  
導入

### Orchestrator導入後



分散処理実行により効率的なロボット運用と自動化プロセス完了の高速化を実現！

#### 改善点

- ロボット端末の通信障害によるプロセス実行停止リスクの分散
- リードタイム削減によるWFの問題早期発見
- 待機中のロボットを有効活用することが可能
- 複数のロボットによる分散処理実行が可能

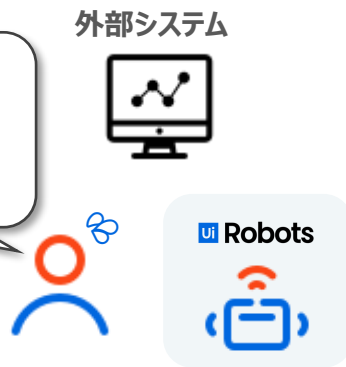
## 2. Orchestrator導入のメリット

### 2-9. 外部システムからの実行

外部システムからAPI経由で人間の操作を介さない完全自動化が可能に！

#### Orchestrator導入前

- 実行ボタン押す
- PC前での監視  
大変。。。



外部システムの何らかの処理をトリガーに実行すべきプロセスがあっても人間が実行命令操作をする必要有

#### 問題点

- 人間による実行忘れによる統制上の問題
- ロボットが他プロセスを実行中は終了を待たなければならない
- ユーザーの実行操作が必要 かつ 実行中にユーザーはPC前で監視する必要有

Orchestrator  
導入

#### Orchestrator導入後



外部システムからAPI経由でOrchestratorにプロセス実行命令可能！

#### 改善点

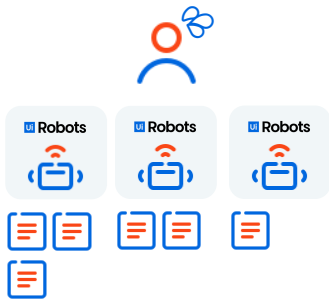
- 人間による実行忘れによる統制上の問題を防止
- ジョブ実行時間が重複した場合でも、前のプロセス完了するまで保留がし全てのプロセスの完了を保証
- 人間の操作を介さないプロセスの完全自動化実行が可能

## 2. Orchestrator導入のメリット

### 2-10. 実行ログの統合管理

ST/ロボットの実行ログを一元管理！開発・保守が効率的に！

#### Orchestrator導入前



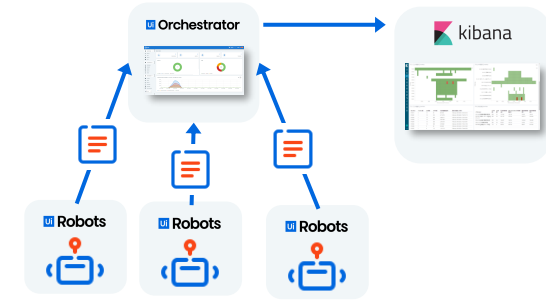
ST/ロボットの実行ログは各端末のローカルフォルダに出力されるため管理が煩雑に  
ログの改ざんも可能

#### 問題点

- プロセス実行時にエラーが発生しても検知が難しい
- ログ調査時PC端末にログインする必要有
- 多数のST/ロボットの端末がある場合運用が煩雑に
- 高度なログ分析は不可能

Orchestrator  
導入

#### Orchestrator導入後



ST/ロボットの実行ログを一元管理！開発・保守が効率的に！

#### 改善点

- プロセス実行時にエラーが発生した時に管理者に通知
- OC上でのログ調査により早期問題解決を促進
- OC画面上で複数のST/ロボット実行ログを一元管理が可能
- ビジネス情報をログに追加し、OCに情報を集約、KibanaでBI分析が可能に

# 2. Orchestrator導入のメリット

## 2-11.サマリー

メリット \ 観点		管理統制	拡張	相互運用
開発	効率的なパッケージのデプロイ	<ul style="list-style-type: none"> <li>OCの制御によって規定に沿ったパブリッシュ/デプロイが可能</li> </ul>	<ul style="list-style-type: none"> <li>大量のロボット数に一括でデプロイが可能</li> <li>パッケージの一括アップデートが可能</li> </ul>	
	効率的なライブラリの配布		<ul style="list-style-type: none"> <li>STの「パッケージを管理」から直接インストール可能</li> <li>実行時にWFで使用するライブラリパッケージがロボット端末上に存在しない場合、OCから自動で取得してからWFを実行</li> </ul>	
	効率的・セキュアなアセットの管理	<ul style="list-style-type: none"> <li>『パスワードを含む各種システムのログイン情報』など、機密性の高い情報もセキュアに一元管理</li> </ul>	<ul style="list-style-type: none"> <li>複数のWF内で使用する可能性のある『サイトURL』などの文字列を一元管理可能</li> </ul>	
管理	効率的なライセンス配分・管理	<ul style="list-style-type: none"> <li>リアルタイムの各ライセンスの使用状況が監視可能</li> </ul>	<ul style="list-style-type: none"> <li>部署ごとのST/ロボットライセンス数の配分・変更をOC上の操作のみで完結可能</li> </ul>	
	効率的なロボット、マシン、パッケージの管理	<ul style="list-style-type: none"> <li>管理のための台帳記入などの業務が不要</li> <li>OCの制御によって規定に沿ったRPA運用が可能</li> <li>管理対象要素のリアルタイムの監視が可能</li> </ul>	<ul style="list-style-type: none"> <li>管理対象要素の数に関係なく統制の効いた管理が可能</li> </ul>	
実行	スケジューリング実行	<ul style="list-style-type: none"> <li>人間による実行忘れによる統制上の問題を防止</li> </ul>	<ul style="list-style-type: none"> <li>スケジュールの時間が重複した場合でも、前のプロセス完了するまで保留がし全てのプロセスの完了を保証</li> </ul>	<ul style="list-style-type: none"> <li>人間の操作を介さないプロセスの完全自動化実行が可能</li> </ul>
	分散処理実行	<ul style="list-style-type: none"> <li>ロボット端末の通信障害によるプロセス実行の停止リスクの分散</li> <li>リードタイム削減によるWFの問題早期発見</li> </ul>	<ul style="list-style-type: none"> <li>待機中のロボットを有効活用することが可能</li> </ul>	<ul style="list-style-type: none"> <li>複数のロボットによる分散処理実行が可能</li> </ul>
	外部システムからの実行	<ul style="list-style-type: none"> <li>人間による実行忘れによる統制上の問題を防止</li> </ul>		<ul style="list-style-type: none"> <li>人間の操作を介さないプロセスの完全自動化実行が可能</li> </ul>
	実行ログの統合管理・分析	<ul style="list-style-type: none"> <li>プロセス実行時にエラーが発生した時に管理者に通知</li> <li>OC上でのログ調査により早期問題解決を促進</li> </ul>	<ul style="list-style-type: none"> <li>OC上の画面上で複数のST/ロボット実行ログを一元管理が可能</li> </ul>	<ul style="list-style-type: none"> <li>ビジネス情報をログに追加し、OCに情報を集約、KibanaでBI分析が可能に</li> </ul>

## 第2章

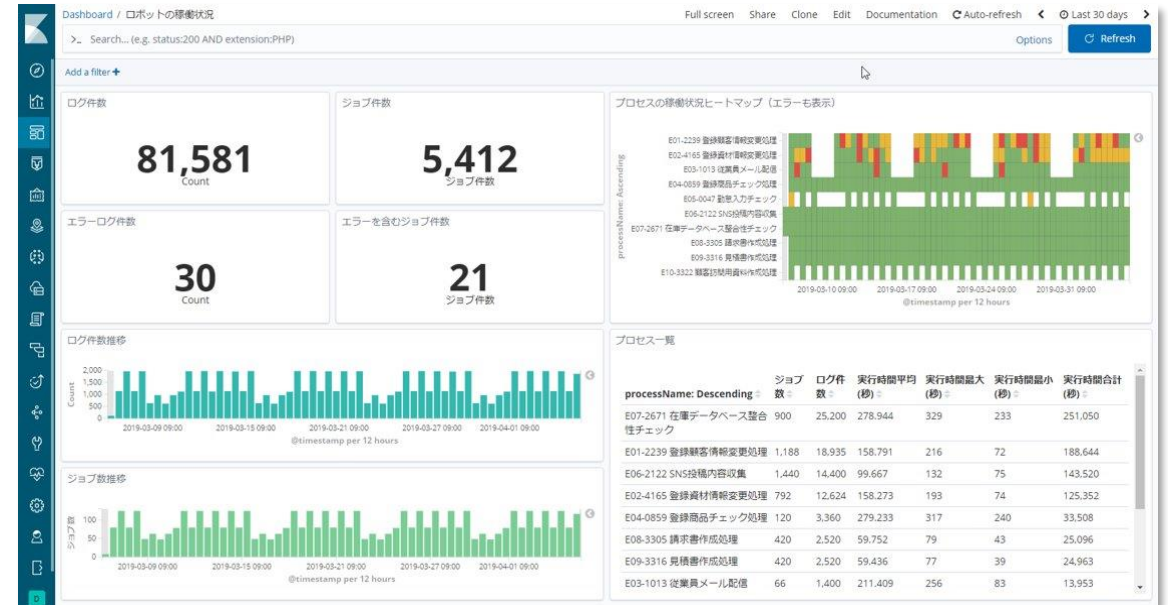
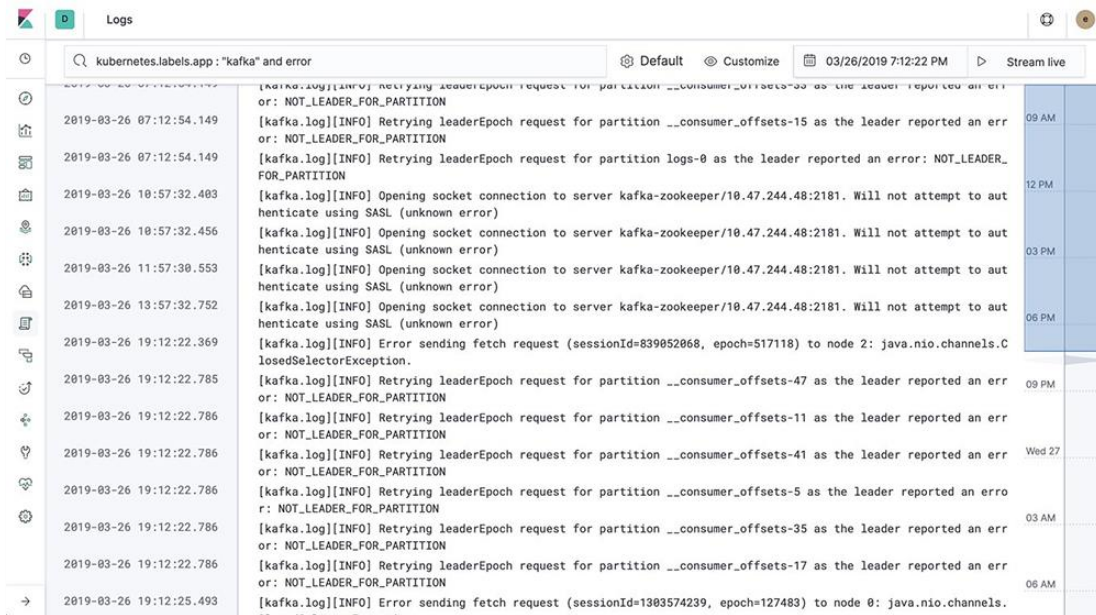
# 3. Elasticsearch & Kibanaによる実行ログの高度分析

1. **UiPath Platform 概要説明**
  - 1-1. 本資料で使用する用語
  - 1-2. UiPath Platform の構成
2. **Orchestrator 導入のメリット**
  - 2-1. Orchestrator 導入のメリット一覧
  - 2-2. 効率的なパッケージのデプロイ
  - 2-3. 効率的なライブラリの配布
  - 2-4. 効率的・セキュアなアセットの管理
  - 2-5. 効率的なロボット、マシン、パッケージの管理
  - 2-6. 効率的なライセンス配分の運用
  - 2-7. スケジューリング実行
  - 2-8. 分散処理実行
  - 2-9. 外部システムからの実行
  - 2-10. 実行ログの統合管理
  - 2-11. サマリー
3. **Elasticsearch & Kibana による実行ログの高度分析**
  - 3-1. Elasticsearch & Kibana とは
  - 3-2. Elasticsearch & Kibana での実行ログ集約分析
  - 3-3. ユースケース
  - 3-4. 『ログフィールドを追加』アクティビティによるログ拡張
4. **Orchestrator 主要機能一覧**
  - 4-1. Orchestrator の主要機能一覧
5. **Orchestrator エンティティ説明**
  - 5-1. Orchestrator の主要エンティティ一覧
  - 5-2. マシン
  - 5-3. ロボット
  - 5-4. ロボットグループ
  - 5-5. パッケージ
  - 5-6. プロセス
  - 5-7. ジョブ
  - 5-8. 時間トリガー
  - 5-9. キュートリガー
  - 5-10. キュー
  - 5-11. アセット
  - 5-12. ライブラリ
6. **Orchestrator 構成について**
  - 6-1. Orchestrator の基本構成要素
  - 6-2. 冗長構成のイメージ
  - 6-3. Orchestrator クラウド (AWS) 構成とそのメリット
  - 6-4. AWS 構成時のネットワークについて

# 3. Elasticsearch & Kibanaによる実行ログの高度分析

## 3-1. Elasticsearch & Kibanaとは

- Elasticsearch は Elastic社が開発により進められているOSSの分散型RESTful検索/分析エンジン。
- Kibanaも同社が開発をしているElasticsearch用のデータ視覚化ダッシュボードOSS



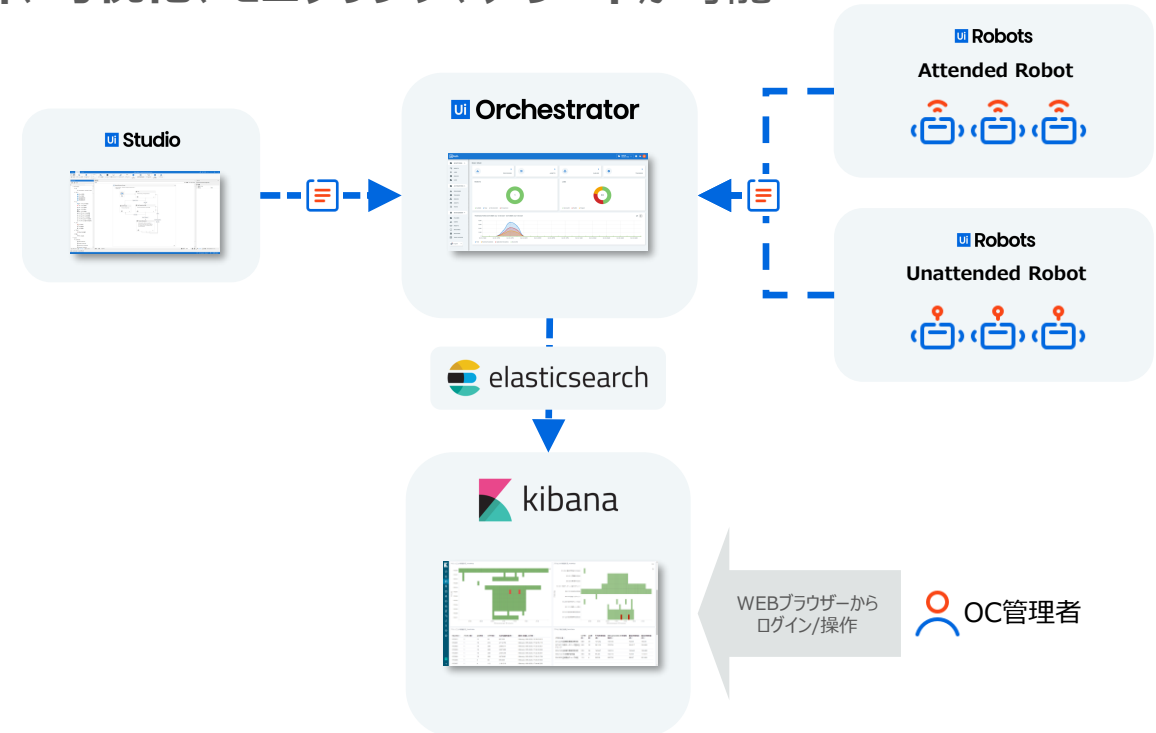
# 3. Elasticsearch & Kibanaによる実行ログの高度分析

## 3-2.Elasticsearch & Kibanaでの実行ログ集約分析

- Orchestrator で管理する Robot から送信されるすべてのログデータを Elasticsearch に格納することが可能
- SQL Server とは違い、Elasticsearch（全文検索エンジン）は**大量のデータの格納や分析**に向く（SQL Server に大量のログ書き込みが発生するとOrchestratorのパフォーマンスが落ちるため、UiPath社としてはElasticsearchの導入を強く推奨）
- Kibana は Elasticsearch のモジュールで、**データの分析、可視化、モニタリング、アラート**が可能

### Elasticsearch/Kibana の有用な機能

- 文書横断的な全文検索
- 時系列データの検索・分析
  - 指定した時間範囲のデータを取得
  - 条件に合致するレコードの取得
  - 数値の集計（平均値や最大値などの統計情報）
- 大量のデータを扱える
  - 複数台のクラスタ構成が可能
  - 数百台のクラスタでペタバイトクラスのデータを扱える
- データの可視化とダッシュボードのカスタマイズ
  - ロボットの稼働状況をグラフィカルに把握
  - 実行頻度の少ないマシンの把握

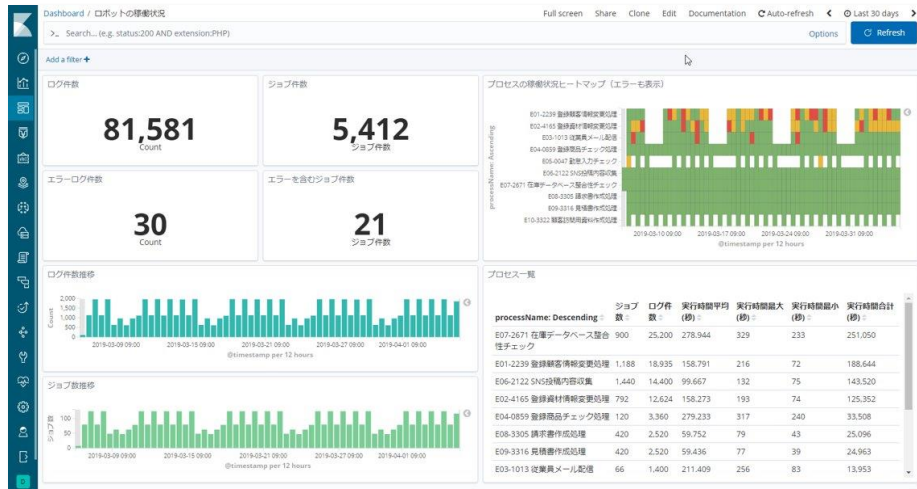




# 3. Elasticsearch & Kibanaによる実行ログの高度分析

## 3-3.ユースケース

- Elasticsearch & Kibanaでの実行ログ集約分析のユースケースは以下の通り



### 障害の原因調査効率化

- ロボの実行状況を速やかに把握すると共に、エラー発生時は大量の実行ログ内から特定のログ情報を検索することが可能であり、障害時の原因調査を効率化

### ロボットの資源最適化

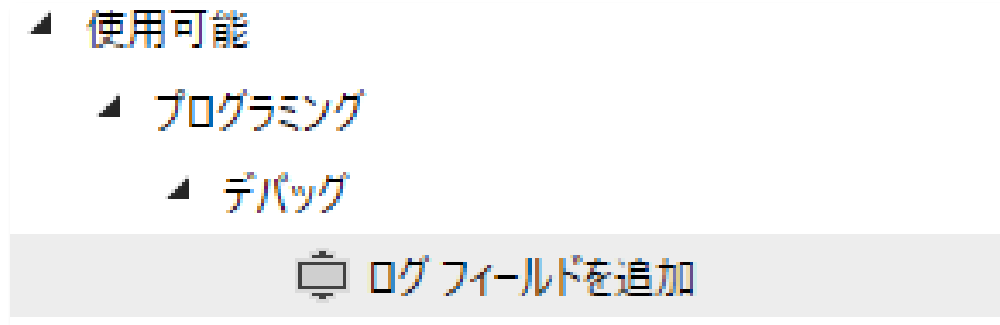
- ロボの未稼働時間を時系列グラフで可視化し、プロセスの活用状況及びロボット端末（PC）の空き時間を把握することが容易
- 本情報を活用し、各ロボの実行計画を最適化することでライセンス費用を抑制

### 業務影響のある障害を回避

- 長期に保存したログを利用して、前週比や前月比の各ロボのログを比較し異常な動きを早期に検知
- エラーログアラート機能により障害の重症化を回避

# 3. Elasticsearch & Kibanaによる実行ログの高度分析

## 3-4. 「ログフィールドを追加」アクティビティによるログ拡張



『ログフィールドを追加』アクティビティを使って

- 組織
- 対象システム名称
- 対象部署
- 処理区分（参照、更新）
- トランザクション情報（ID、識別パラメータ、トランザクション処理時間）

のような情報をログに追加することができる。

追加ログフィールドの情報を使ってElasticsearch/Kibana でフィルタリング・全文検索が可能になるため

- トラブルシューティング
- BI分析

などの場面で有効活用が可能になる。

# 第2章

## 4. Orchestrator 主要機能一覽

1. **UiPath Platform 概要説明**
  - 1-1. 本資料で使用する用語
  - 1-2. UiPath Platform の構成
2. **Orchestrator 導入のメリット**
  - 2-1. Orchestrator 導入のメリット一覧
  - 2-2. 効率的なパッケージのデプロイ
  - 2-3. 効率的なライブラリの配布
  - 2-4. 効率的・セキュアなアセットの管理
  - 2-5. 効率的なロボット、マシン、パッケージの管理
  - 2-6. 効率的なライセンス配分の運用
  - 2-7. スケジューリング実行
  - 2-8. 分散処理実行
  - 2-9. 外部システムからの実行
  - 2-10. 実行ログの統合管理
  - 2-11. サマリー
3. **Elasticsearch & Kibana による実行ログの高度分析**
  - 3-1. Elasticsearch & Kibana とは
  - 3-2. Elasticsearch & Kibana での実行ログ集約分析
  - 3-3. ユースケース
  - 3-4. 『ログフィールドを追加』アクティビティによるログ拡張
4. **Orchestrator 主要機能一覧**
  - 4-1. Orchestrator の主要機能一覧
5. **Orchestrator エンティティ説明**
  - 5-1. Orchestrator の主要エンティティ一覧
  - 5-2. マシン
  - 5-3. ロボット
  - 5-4. ロボットグループ
  - 5-5. パッケージ
  - 5-6. プロセス
  - 5-7. ジョブ
  - 5-8. 時間トリガー
  - 5-9. キュートリガー
  - 5-10. キュー
  - 5-11. アセット
  - 5-12. ライブラリ
6. **Orchestrator 構成について**
  - 6-1. Orchestrator の基本構成要素
  - 6-2. 冗長構成のイメージ
  - 6-3. Orchestrator クラウド (AWS) 構成とそのメリット
  - 6-4. AWS 構成時のネットワークについて

# 4. Orchestrator主要機能一覧

## 4-1.Orchestratorの主要機能一覧(1/2)

カテゴリ	機能	概要
ダッシュボード	ダッシュボード	<ul style="list-style-type: none"><li>Robot、ジョブ等の状況を一覧で表示</li></ul>
リリース管理	ワークフロー管理	<ul style="list-style-type: none"><li>ワークフローパッケージの集中管理</li><li>ワークフローパッケージのバージョン管理</li></ul>
実行管理	ロボット管理	<ul style="list-style-type: none"><li>ロボットの登録</li><li>ロボットのステータス・起動ログ管理</li></ul>
	ロボットグループ管理	<ul style="list-style-type: none"><li>Robot のグループ分け</li></ul>
	ジョブステータス管理	<ul style="list-style-type: none"><li>各ジョブの実行状況、結果を一覧で表示</li><li>必要に応じてアラートメールの発報</li></ul>
	実行ログ管理	<ul style="list-style-type: none"><li>ジョブの実行ログの一元管理</li></ul>
	トリガー管理	<ul style="list-style-type: none"><li>時間トリガーの管理</li><li>キュートリガーの管理</li></ul>
	アセット管理	<ul style="list-style-type: none"><li>変数の共有、及び、一元管理</li></ul>
	キュー管理	<ul style="list-style-type: none"><li>大量のトランザクションの分散実行</li></ul>

# 4. Orchestrator主要機能一覧

## 4-1.Orchestratorの主要機能一覧(2/2)

カテゴリ	機能	概要
権限管理	ユーザー管理	• Orchestrator に接続するユーザーの管理
	ユーザーロール管理	• Orchestrator に接続するユーザーの権限管理
	テナント管理	• 全オブジェクト（パッケージ、Robot 等）をテナントごとに別管理
	フォルダー管理	• 単一テナント内で、Robot、プロセスなどのオブジェクトをフォルダーごとに管理
その他	ライセンス管理	• Orchestrator に接続する Studio や各 Robotのライセンスを集中管理
	監査	• Orchestrator のすべてのエンティティによって実行された操作の監査証跡を表示
	ログ分析（Kibana）	• 実行ログの高度な分析
	REST API	• 外部システムからのジョブの実行、キューの追加等、管理者が実行できる操作全て
	Web Hook	• Orchestratorで発生したイベントをWeb APIに対して通知

# 第2章

## 5. Orchestrator工 ンティティ説明

1. **UiPath Platform 概要説明**
  - 1-1. 本資料で使用する用語
  - 1-2. UiPath Platform の構成
2. **Orchestrator 導入のメリット**
  - 2-1. Orchestrator 導入のメリット一覧
  - 2-2. 効率的なパッケージのデプロイ
  - 2-3. 効率的なライブラリの配布
  - 2-4. 効率的・セキュアなアセットの管理
  - 2-5. 効率的なロボット、マシン、パッケージの管理
  - 2-6. 効率的なライセンス配分の運用
  - 2-7. スケジューリング実行
  - 2-8. 分散処理実行
  - 2-9. 外部システムからの実行
  - 2-10. 実行ログの統合管理
  - 2-11. サマリー
3. **Elasticsearch & Kibana による実行ログの高度分析**
  - 3-1. Elasticsearch & Kibana とは
  - 3-2. Elasticsearch & Kibana での実行ログ集約分析
  - 3-3. ユースケース
  - 3-4. 『ログフィールドを追加』アクティビティによるログ拡張
4. **Orchestrator 主要機能一覧**
  - 4-1. Orchestrator の主要機能一覧
5. **Orchestrator エンティティ説明**
  - 5-1. Orchestrator の主要エンティティ一覧
  - 5-2. マシン
  - 5-3. ロボット
  - 5-4. ロボットグループ
  - 5-5. パッケージ
  - 5-6. プロセス
  - 5-7. ジョブ
  - 5-8. 時間トリガー
  - 5-9. キュートリガー
  - 5-10. キュー
  - 5-11. アセット
  - 5-12. ライブラリ
6. **Orchestrator 構成について**
  - 6-1. Orchestrator の基本構成要素
  - 6-2. 冗長構成のイメージ
  - 6-3. Orchestrator クラウド (AWS) 構成とそのメリット
  - 6-4. AWS 構成時のネットワークについて



# 5. Orchestratorエンティティ説明

## 5-1.Orchestratorの主要エンティティ一覧

- Orchestratorで管理できる主要なエンティティは以下の通り
- マシン
- ロボット
- ロボットグループ
- パッケージ
- ライブラリ
- プロセス
- ジョブ
- キュー
- トリガー（キュー）
- トリガー（スケジュール）
- アセット
- ライブラリ

# 5. Orchestratorエンティティ説明

## 5-2.マシン

『マシン』とはStudio / Robotを実行できる状態にするためのPC端末をOrchstrator上に登録したエンティティ。  
『マシン』には標準、テンプレート の2種類がある。



### Ui Orchestrator

NAME ^	DESCRIPTION	TYPE ^	INSTALLED VERSION
CAPETRINA	This is Petrina's machine.	Standard	No Robots
PEGABRIEL	And this is mine.	Standard	Unknown

#### 標準

ロボットを定義するマシンの名前が常に同じである場合に使用。

#### テンプレート

非永続的な仮想デスクトップインフラストラクチャ (VDI) の場合など、Robot を定義するマシンの名前がその都度変更する場合に使用。  
このエンティティを設定すると、Active Directory のユーザー名を使用して、Attended Floating Robot を台数の制限なく接続できる。  
(この場合のRobot は、マシンキーとユーザー名の組み合わせのみを使用して Orchestrator に接続する。)

# 5. Orchestratorエンティティ説明

## 5-3.ロボット

『ロボット』とは『マシン』とWindowsユーザーの組み合わせで定義される「プロセスを実行するホスト」を表すエンティティ。



### Ui Orchestrator

The screenshot shows the 'Robots' tab in the UiPath Orchestrator interface. The table lists the following robots:

NAME	MACHINE	USERNAME	TYPE	ENVIRONMENTS	STATUS
Gustave	LAPTOP-IN7198N3	uipath\petrina.calota	Studio		Available
Jeff	PEGABRIEL	uipath\gabriel.pene	Studio	DOC	Available
Alexxx	voloacaalex	uipath\alexandru.voloaca	Studio	DENV	Disconnected

#### Attended Robot

人間ユーザーと同じワークステーションで動作し、日常的タスクの実行でユーザーを支援する。通常、ユーザーイベントによってトリガーされる。この種類のロボットの Orchestrator からプロセスを開始することはできない。また、これらのロボットは、ロックされた画面では実行できない。ロボットトレイまたはコマンドプロンプトからのみ開始できる。Attended ロボットは、人間の監視下でのみ実行必要有。

#### Unattended Robot

無人で作動し、多数のプロセスを自動で行うことができる。Attended ロボットの機能に加え、リモートでの実行や監視、スケジューリング、ワークキューのサポートも行うことができる。

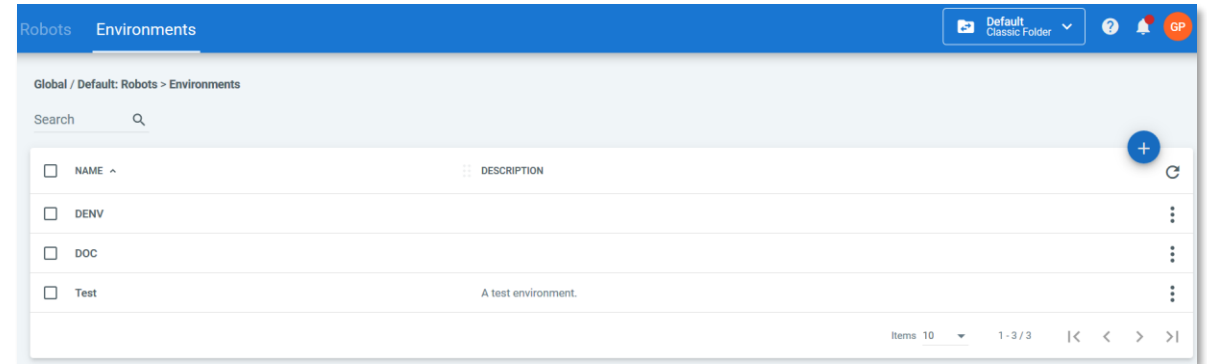
# 5. Orchestratorエンティティ説明

## 5-4.ロボットグループ

『ロボットグループ』とは、Orchestrator上で定義できるロボットの集合体。  
『プロセス』のデプロイに使用される。



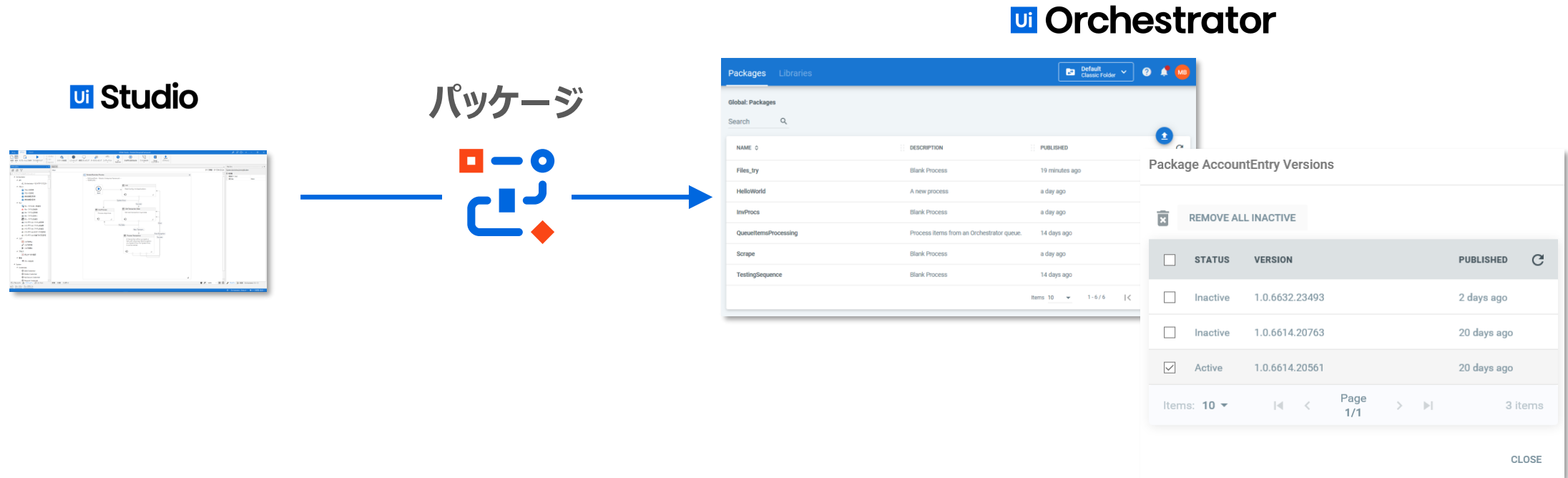
### Ui Orchestrator



# 5. Orchestratorエンティティ説明

## 5-5.パッケージ

『パッケージ』とは、UiPath Studio で開発したプロセスプロジェクトを、ロボットが実行できる .NuGetの形でフォーマットされたエンティティ。

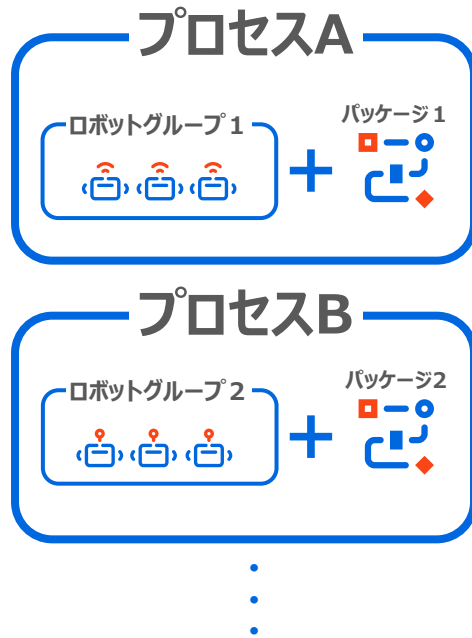


- StudioからOrchestratorに直接、または手動でアップロードすることでOrchestratorに『パッケージ』をパブリッシュすることができる。
- Orchestrator上で過去から現在までにパブリッシュされた各バージョンの『パッケージ』をダウンロードすることができる。
- Orchestrator上で現在デプロイされている（アクティブな）パッケージバージョンを確認することができる

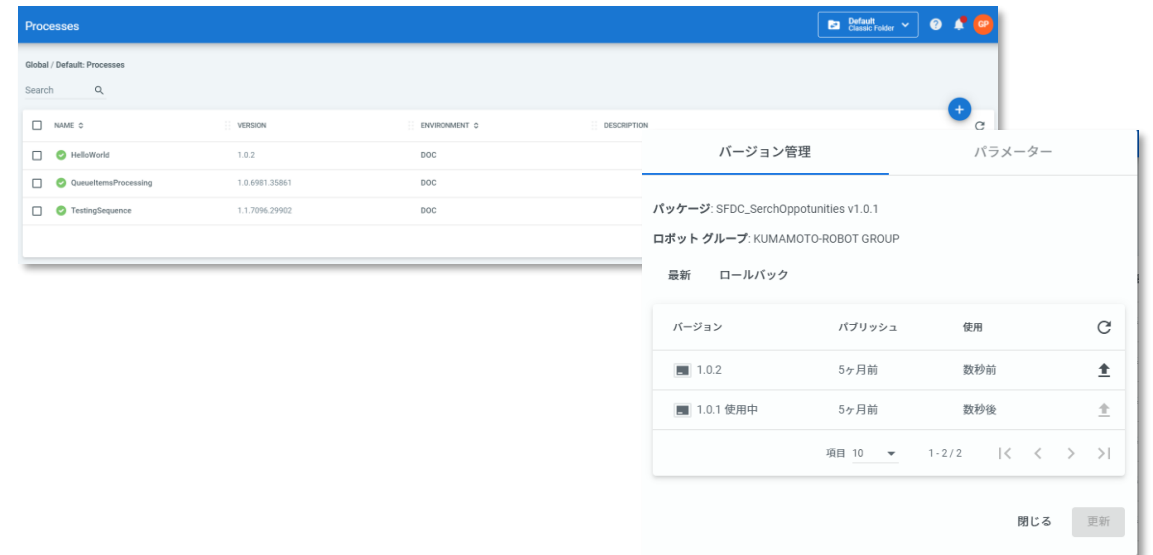
# 5. Orchestratorエンティティ説明

## 5-6.プロセス

『プロセス』とは、指定の『ロボットグループ』によって指定の『パッケージ』を実行可能な状態を定義するエンティティ。



### Ui Orchestrator



- デployされた『プロセス』に紐づいたパッケージのバージョンを変更指定することで、AR・URに関わらず次からロボットが該当『プロセス』を実行する際は指定バージョンの『パッケージ』に基づいたワークフローを実行する => ワークフローの改修・ロールバック後のデプロイが簡単に！

# 5. Orchestratorエンティティ説明

## 5-7. ジョブ

『ジョブ』とは、1つのロボットによる『プロセス』の実行を表すエンティティ。



## Ui Orchestrator

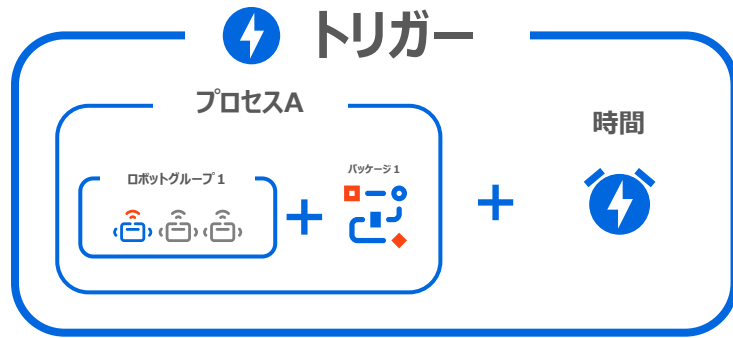
PROCESS	ROBOT	ENVIRONMENT	STATE	STARTED	ENDED	SOURCE
<input type="checkbox"/> HelloWorld	Bob	DOC	Successful	a few seconds ago	a few seconds ago	Manual
<input type="checkbox"/> HelloWorld	Bob	DOC	Successful	a minute ago	a minute ago	Manual
<input type="checkbox"/> TestingSequence	Bob	DOC	Faulted	9 minutes ago	9 minutes ago	Test
<input type="checkbox"/> QueueItemsProcessing	Bob	DOC	Faulted	10 minutes ago	10 minutes ago	Manual
<input type="checkbox"/> TestingSequence	Bob	DOC	Faulted	10 minutes ago	10 minutes ago	Test

- 『ジョブ』は以下のタイミングでOrchestratorから『ジョブ』が生成される
  - Orchestratorのジョブページにて『ジョブ』を作成した時
  - Orchestratorのトリガーページにて『トリガー』作成時に『プロセス』を指定して後、『トリガー』の実行条件のイベントが発生した時
  - 外部システムからAPIを通して『プロセス』の指定、『ジョブ』の実行命令をOrchestratorが受け取った時

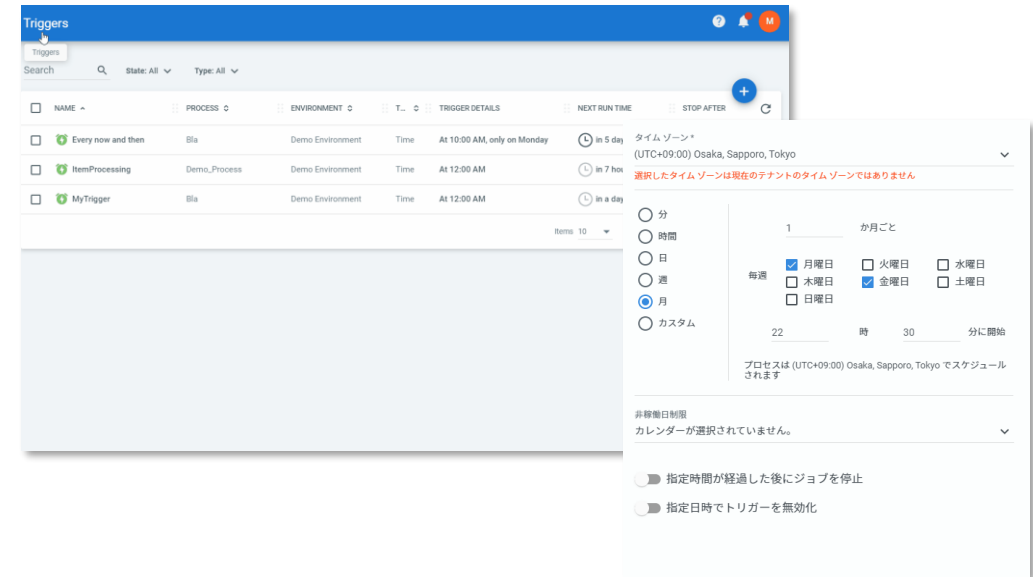
# 5. Orchestratorエンティティ説明

## 5-8.時間トリガー

『時間トリガー』とは、指定された時間になると『ジョブ』を生成するエンティティ



### Ui Orchestrator



- トリガーにより『ジョブ』生成時に実行可能なロボットがなかった場合、『ジョブ』は保留中となり実行可能なロボットが空くまで待ち続ける
- 非稼働日カレンダーを設定してある日付に、トリガーの『ジョブ生成』が行われる

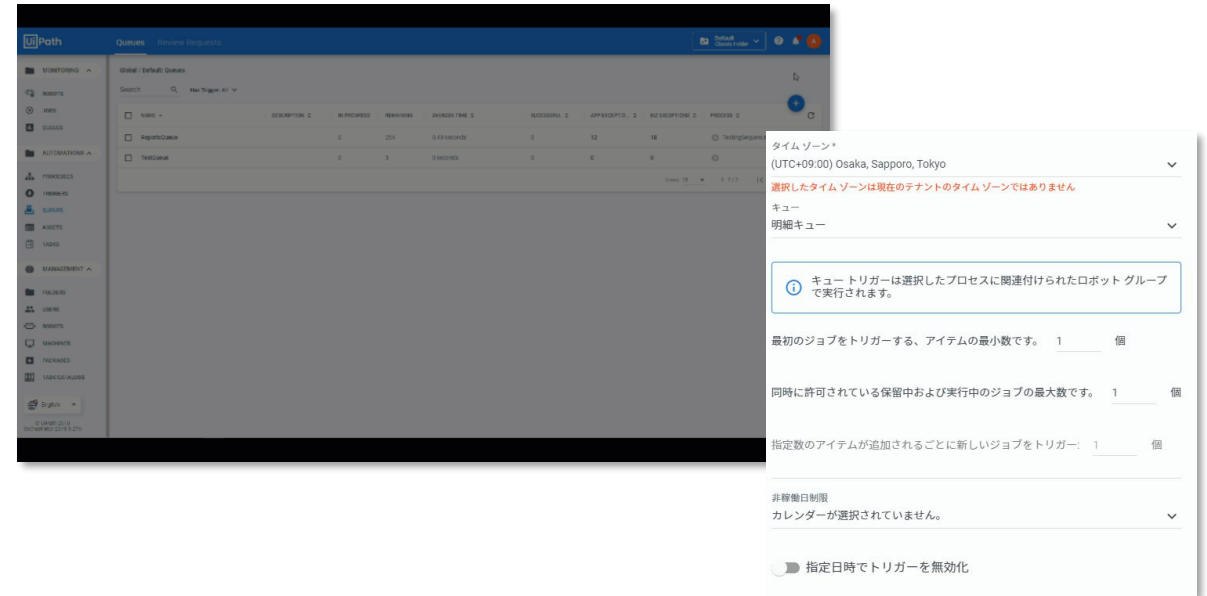
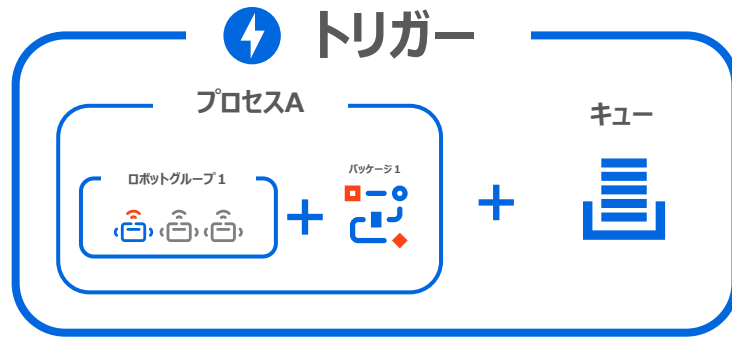


# 5. Orchestratorエンティティ説明

## 5-9.キュートリガー

『キュートリガー』とは、指定された『キュー』にキューアイテムが追加されると『ジョブ』を生成するエンティティ

### Ui Orchestrator



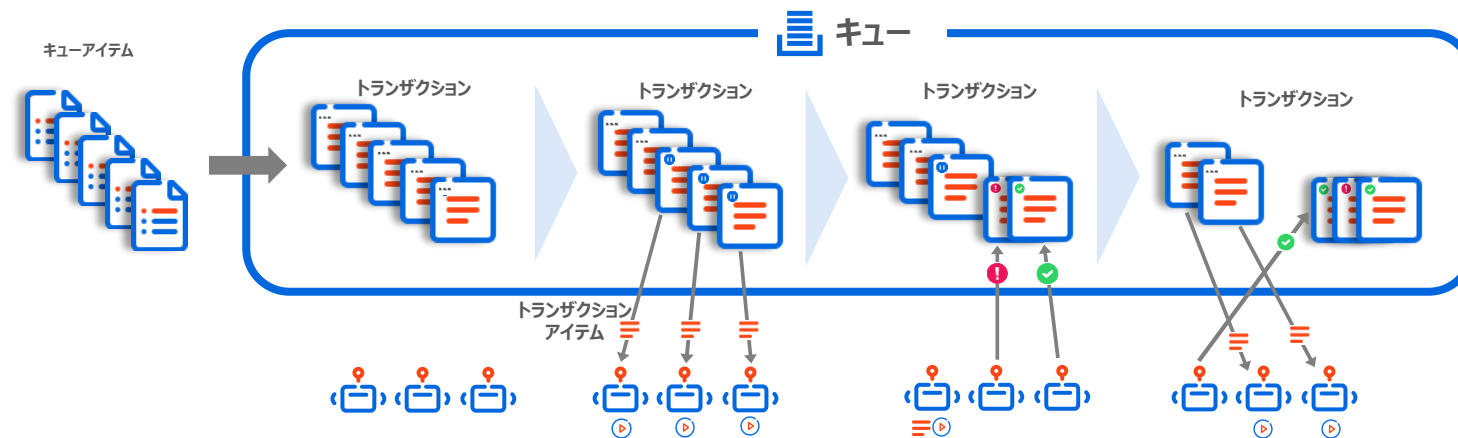
- トリガーにより『ジョブ』生成時に実行可能なロボットがなかった場合、『ジョブ』は保留中となり実行可能なロボットが空くまで待ち続ける
- 非稼働日カレンダーを設定してある日付に、トリガーの『ジョブ生成』が行われる

# 5. Orchestratorエンティティ説明

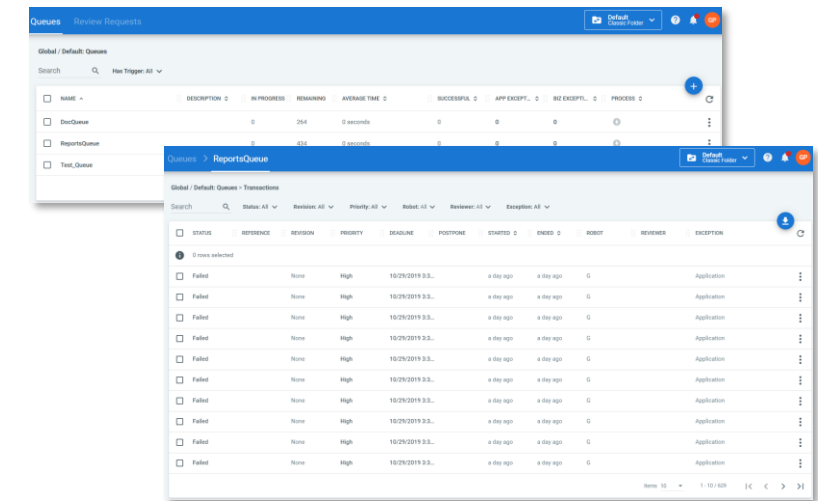
## 5-10.キュー(1/2)

『キュー』とは、『キューアイテム』と呼ばれる「様々な種類のデータ内の1レコード毎のエンティティ（顧客リストの内の1顧客データなど）」をアップロードすることによって生成される『トランザクション』を無制限に保存可能なエンティティ。

複数のロボットを利用した分散処理を可能にする。



### Ui Orchestrator

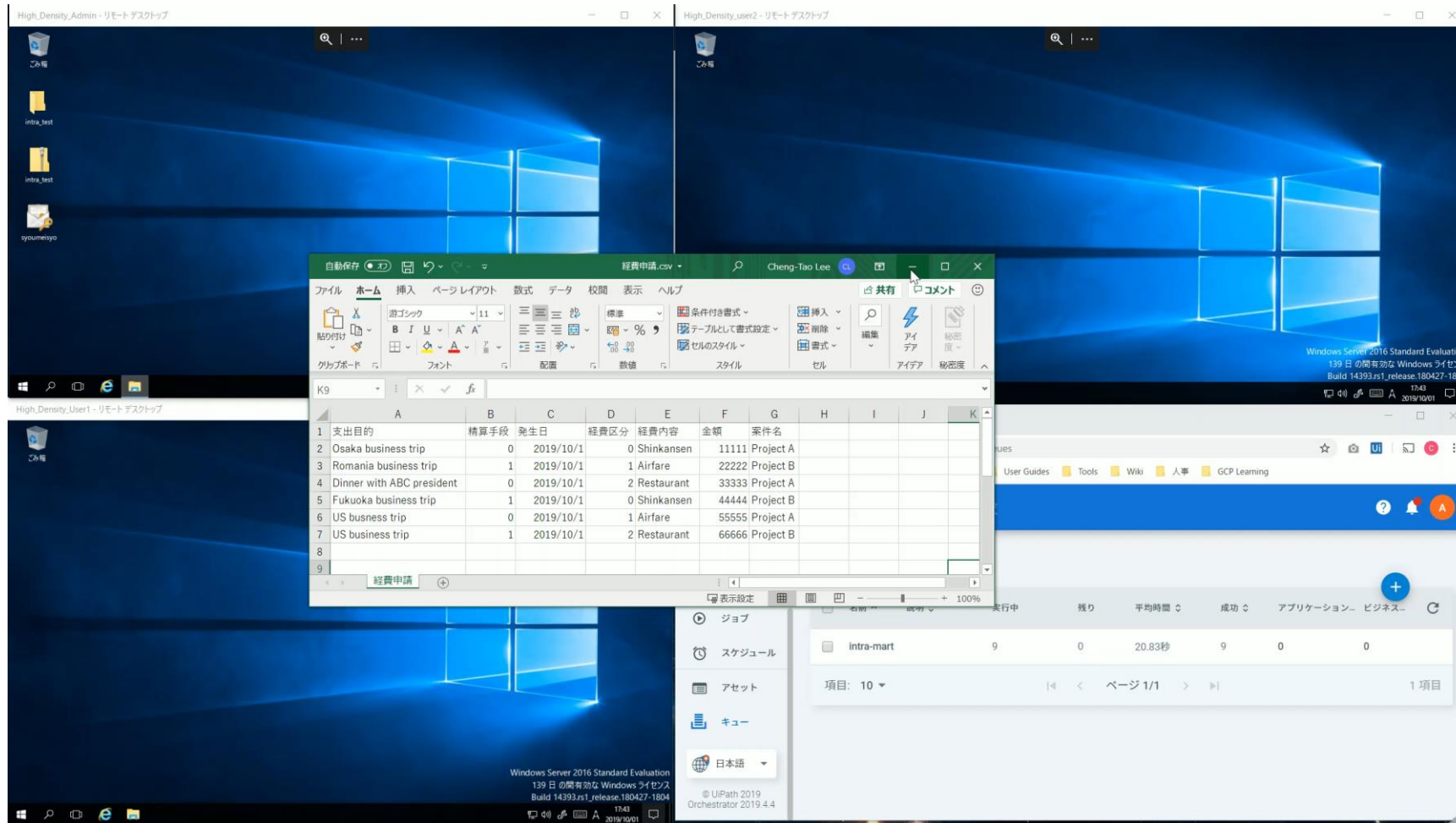


- キューアイテムのアップロード方法は以下
  - OrchestratorのキューページからCSVをアップロード
  - ロボットから「キューアイテムを追加 / 一括追加」アクティビティを利用してアップロード
- トランザクションアイテムはロボットから「トランザクションアイテムを取得」アクティビティで取得可能

# 5. Orchestratorエンティティ説明

## 5-10.キュー(2/2)

『キュー』機能を使った複数ロボットによる分散実行デモ



# 5. Orchestratorエンティティ説明

## 5-11.アセット

『アセット』とは、異なる自動化プロジェクトで使用可能な共有変数または認証情報をOrchestrator上に保存するエンティティ。アセットに特定の情報を格納することで、ロボットは容易にその情報にアクセスすることができます。

**アセットA**

- Text
- Bool
- Integer
- Credential

ユーザー名  
taro.tanaka@uipath.com

パスワード  
.....

**アセットB**

- Text
- Bool
- Integer
- Credential

値  
https://docs.uipath.com

### Ui Orchestrator

NAME	DESCRIPTION	TYPE	VALUE
CompanyName	A text asset	Text	UiPath
Demo_Credential		Credential	[In credential store]
Doc_test	A Boolean Asset	Bool	True
Queue_Asset	An integer asset	Integer	26
UIDemo_Credential	Demo	Credential	[In credential store]

- アセットに格納できる値は「Text」「Bool」「Integer」「Credential（ユーザー名とパスワードの組み合わせ）」の4種類
- 「Credential」アセットはAES 256 アルゴリズムで暗号化されるため、高いレベルのセキュリティ保証される。開発者もこの情報を呼び出すことができるが、値はセキュアストリングで保存されているため非表示となる。
- 各アセットにロボット毎に異なる値を設定することも可能。
- 各アセットの値は「アセットを取得」「資格情報を取得」アクティビティでロボットから取得可能

# 5. Orchestratorエンティティ説明

## 5-12.ライブラリ

『ライブラリ』とは、複数の再利用可能なコンポーネントを含む.NuGetの形でフォーマットされたエンティティ。



- UiPath Studio で開発したライブラリプロジェクトを、ロボットがアクティビティとして扱う事の出来るファイルもライブラリに含まれる。
- StudioからOrchestratorに直接、または手動でアップロードすることでOrchestratorにライブラリをパブリッシュすることができる。
- OrchestratorにパブリッシュされたライブラリはStudioのパッケージ管理からダウンロード可能。

# 第2章

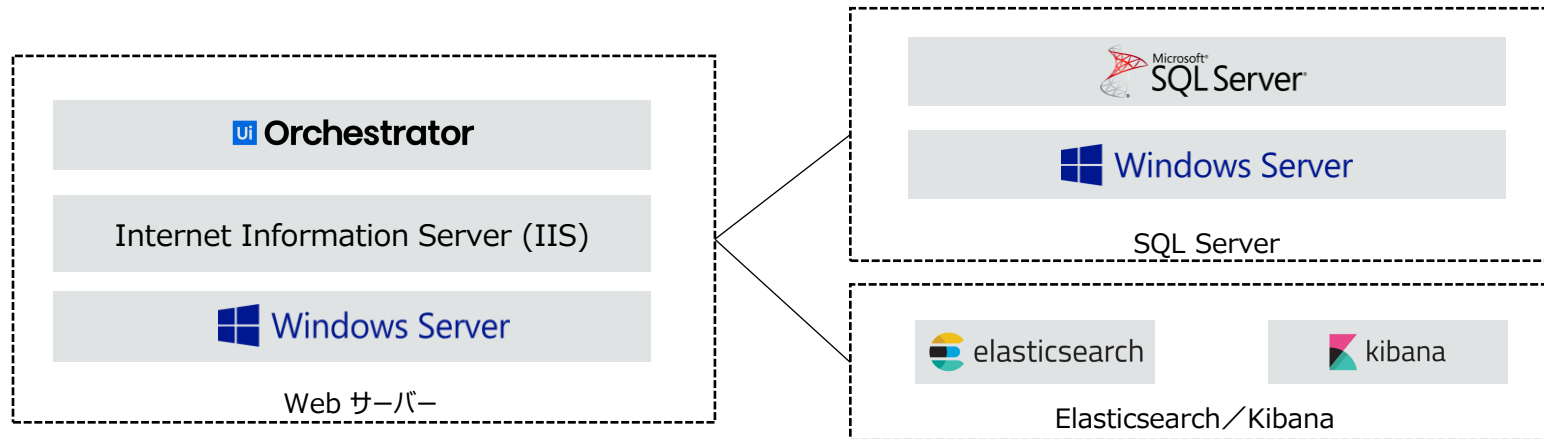
## 6. Orchestrator構成について

1. **UiPath Platform 概要説明**
  - 1-1. 本資料で使用する用語
  - 1-2. UiPath Platform の構成
2. **Orchestrator 導入のメリット**
  - 2-1. Orchestrator 導入のメリット一覧
  - 2-2. 効率的なパッケージのデプロイ
  - 2-3. 効率的なライブラリの配布
  - 2-4. 効率的・セキュアなアセットの管理
  - 2-5. 効率的なロボット、マシン、パッケージの管理
  - 2-6. 効率的なライセンス配分の運用
  - 2-7. スケジューリング実行
  - 2-8. 分散処理実行
  - 2-9. 外部システムからの実行
  - 2-10. 実行ログの統合管理
  - 2-11. サマリー
3. **Elasticsearch & Kibana による実行ログの高度分析**
  - 3-1. Elasticsearch & Kibana とは
  - 3-2. Elasticsearch & Kibana での実行ログ集約分析
  - 3-3. ユースケース
  - 3-4. 『ログフィールドを追加』アクティビティによるログ拡張
4. **Orchestrator 主要機能一覧**
  - 4-1. Orchestrator の主要機能一覧
5. **Orchestrator エンティティ説明**
  - 5-1. Orchestrator の主要エンティティ一覧
  - 5-2. マシン
  - 5-3. ロボット
  - 5-4. ロボットグループ
  - 5-5. パッケージ
  - 5-6. プロセス
  - 5-7. ジョブ
  - 5-8. 時間トリガー
  - 5-9. キュートリガー
  - 5-10. キュー
  - 5-11. アセット
  - 5-12. ライブラリ
6. **Orchestrator 構成について**
  - 6-1. Orchestrator の基本構成要素
  - 6-2. 冗長構成のイメージ
  - 6-3. Orchestrator クラウド (AWS) 構成とそのメリット
  - 6-4. AWS 構成時のネットワークについて

# 6. Orchestrator構成について

## 6-1.Orchestratorの基本構成要素

- 基本構成要素 : Web サーバー, SQL server, Elasticsearch/Kibana



- すべてのノードは物理マシンでも仮想マシンでも構成が可能
- オンプレミスとクラウドのどちらにも設置が可能

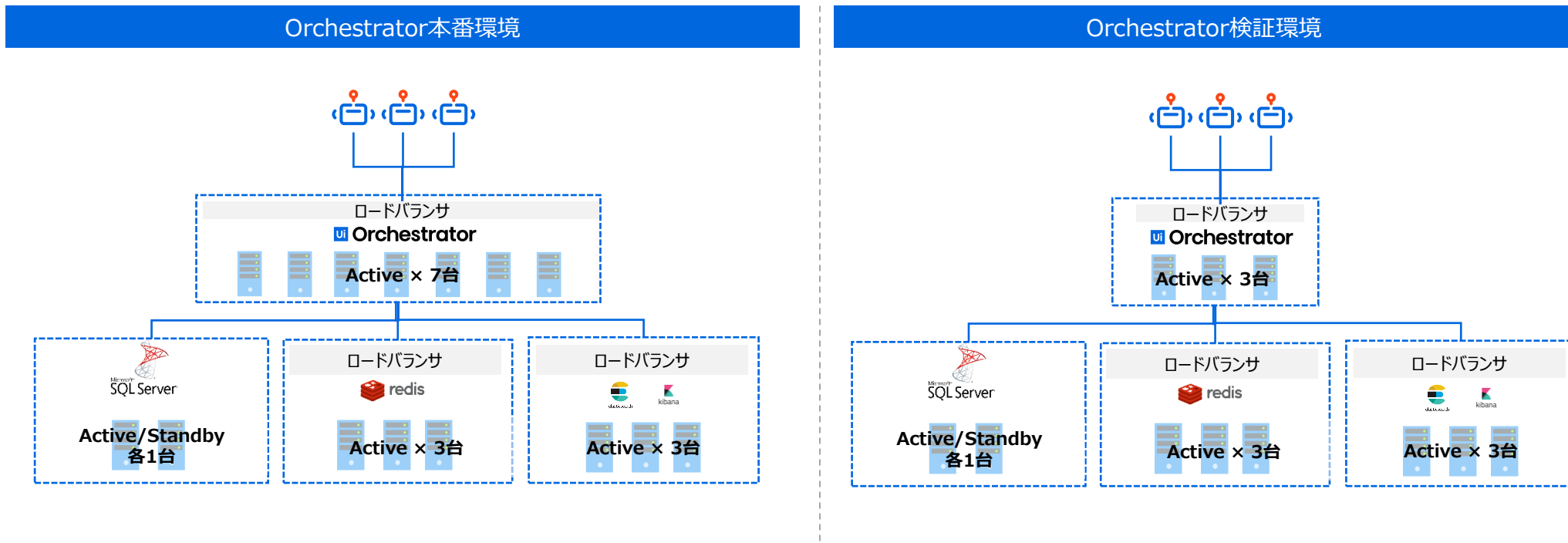


# 6. Orchestrator構成について

## 6-2.冗長構成のイメージ

- ユーザーやロボのアクセスを一定期間保持するためのRedisなどの“セッション管理サーバ”が必要となる
- Orchestratorのバージョンアップ対応や開発プロセスの検証のために、本番環境と同様の冗長構成で且つサイズダウンした検証環境も構築することも検討。

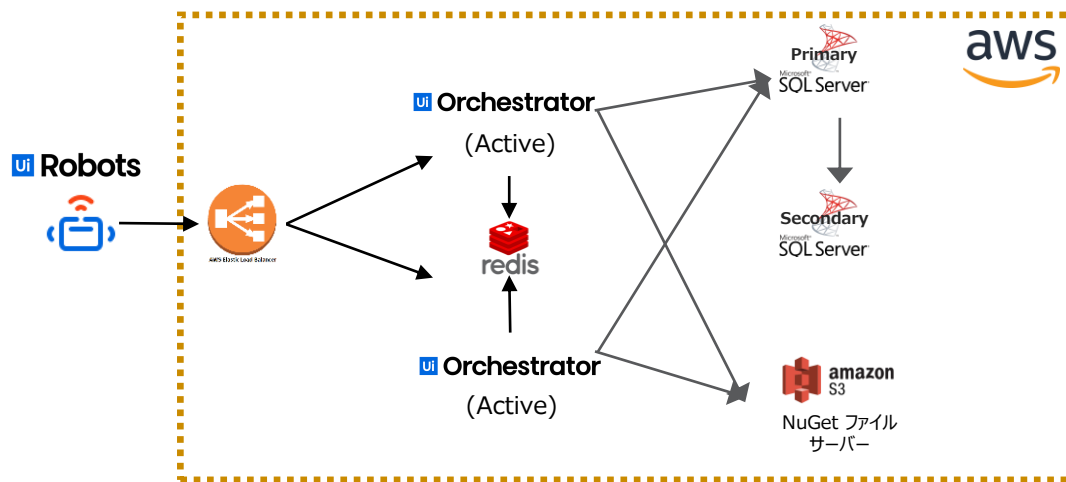
### 大規模UiPath導入プロジェクトの構成例



# 6. Orchestrator構成について

## 6-3.Orchestratorクラウド（AWS）構成とそのメリット

- 冗長構成に求められる各種サーバをオンプレで構築する場合、構成が複雑化し、導入・運用負担が大きい
- AWS上に構築する場合、マネージメントサービスを活用することで比較的シンプルな構成で環境構築が可能となる。また、運用規模に合わせて柔軟にシステムリソースを調整することが可能。



### シンプルな構成かつ短期間で環境構築が可能

AWSのクラウドサービス、特にロードバランサ、DBサービス、Elasticサービス活用することで冗長構成が非常に用意に構築できるため、事前の設計を丁寧に行っておけば短期（数週間）で環境構築が可能

### AWSマネージドサービス活用による運用負担軽減

2つのAvailability Zone（AWSのデータセンター）間の同期および各サービス障害時の自動復旧サービスが提供されるため、簡易なシステム障害時は自動復旧、切替が行われるため基盤環境の運用負担は軽減される（※自社環境特有の問題発生時支援のために有償のAWSテクニカルサービスは契約を推奨）

### 運用規模に合わせて柔軟にシステムリソースを調整することが可能

AWSのマネージメントコンソールから環境の停止、縮小／拡大が容易であるため、システムリソース費用のムダを最小化可能（一部サービスは縮小できない）

# 6. Orchestrator構成について

## 6-4.AWS構成時のネットワークについて

ネットワークの構成案は以下の3案

### ① 閉域網接続

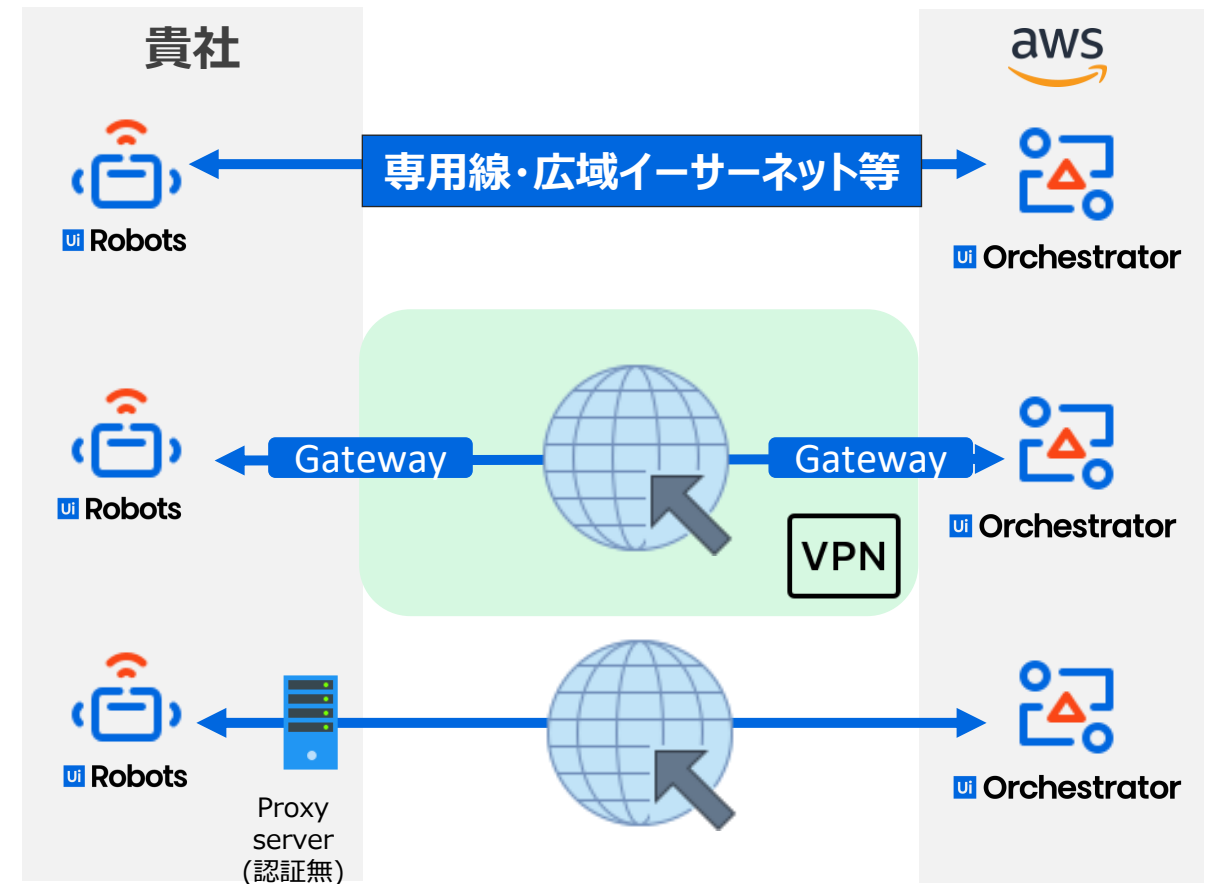
- 貴社LANとAWSを専用線・広域イーサネット等で接続

### ② VPN接続

- 貴社LANとAWSにGatewayを用意しVPN接続

### ③ Proxyサーバー（認証無）経由接続

- 貴社LAN内のProxyサーバーを経由してAWS内のOrchestratorに直接接続
- Proxyサーバー（認証有）はUiPath未対応



## 第3章

# 1. Orchestratorに関する役立つリンク集

## 1. 参考リンク

- 1-1. UiPath Orchestrator 公式ガイド
- 1-2. インストールガイド
- 1-3. 運用、バージョンアップガイド
- 1-4. UiPathディベロッパーブログ
- 1-5. UiPathフォーラム
- 1-6. UiPathアカデミー
- 1-7. UiPath無料評価版
- 1-8. RPAガバナンスガイドライン関連資料
- 1-9. パートナートレーニング

# 1. 参考リンク

## 1-1. UiPath Orchestrator 公式ガイド

UiPath Orchestrator 公式ガイド : <https://docs.uipath.com/orchestrator/lang-ja>

- UiPath 社が提供するオフィシャル・ガイドであり、不明な点等があった場合、一番最初に参照すべきガイド。



# 1. 参考リンク

## 1-2. インストールガイド

- UiPath Global の Orchestrator ガイドにおけるインストールの章

- <https://orchestrator.uipath.com/lang-ja/docs/about-installation>
- 表示言語、Orchestrator バージョンの選択が可能
- インストールマニュアルの位置づけ

- UiPath Japan 作成のインストールガイド

- ナレッジベース : 『Orchestrator 導入ステップバイステップガイド [2019 FT 対応版] 』

- <https://www.uipath.com/ja/resources/knowledge-base/orchestrator-installation-guide>
- Orchestrator と関連コンポーネントである SQL Server, Elasticsearch, Kibana について、インストールと構成の手順を、画面キャプチャと共に説明

- ナレッジベース : 『[2018.3対応版] Orchestrator 導入ステップバイステップガイド』

- <https://www.uipath.com/ja/resources/knowledge-base/orchestrator-installation-guide-2018.3>
- 上記ガイドの 2018.3 対応版

- ナレッジベース : 『Azure環境 UiPath Orchestrator 構築手順書』

- <https://www.uipath.com/ja/resources/knowledge-base/stack-uipath-on-azure>

- UiPath Go! : 『AWS環境 UiPath Orchestrator シングル構成 構築手順書』

- <https://go.uipath.com/component/stack-uipath-on-aws-simple-plan>

- UiPath Go! : 『AWS環境 UiPath Orchestrator 冗長構成 構築手順書』

- <https://go.uipath.com/component/stack-uipath-on-aws-full-plan>



# 1. 参考リンク

## 1-3.運用、バージョンアップガイド

- **UiPath Orchestrator システムの基盤設計・運用ガイド**

<https://www.uipath.com/ja/resources/knowledge-base/uipath-orchestrator-design-operation-guide>

UiPath Orchestratorを導入するためのシステム基盤の設計、構築、運用設計に関する留意事項やノウハウについて解説

- **Orchestrator 管理者のためのミドルウェア運用設定ガイド**

<https://www.uipath.com/ja/resources/knowledge-base/middleware-config-maintenance-guide>

UiPath Orchestratorの運用・保守に従事する管理者として、Orchestrator稼働に必須の構成要素（ソフトウェア）であるミドルウェア（Microsoft Internet Information Service およびMicrosoft SQL Server ）に対して推奨される設定項目、監視項目などのベストプラクティスを解説

- **UiPath Orchestrator Version Up Guide(2019LTS)**

<https://www.uipath.com/ja/resources/knowledge-base/uipath-orchestrator-version-up-guide>

UiPath Orchestratorのバージョンアップを計画、実施するための前提知識や留意事項、環境別例に基づいた基本手順について解説



# 1. 参考リンク

## 1-4. UiPathディベロッパーブログ

UiPath ディベロッパーブログ: <https://www.uipath.com/ja/blog/developer>

- UiPath 社員によるブログ.

UiPath RPA 製品について RPAソリューション リソース 企業情報 パートナー ニュース

トライアルの開始

### Featured Articles

#### UIPATH ディベロッパーブログ

**RPA TIPS**  
VMware Horizon Cloud on Microsoft Azure環境でのUiPath製品の利用について  
2019年8月26日

**RPA TIPS**  
UiPath製品のCitrix環境への対応について  
2019年8月19日

**RPA TIPS**  
UiPath 組み込みの OCR アクティビティを利用するドキュメント処理プラットフォーム  
2019年8月12日

**RPA**  
クリックアクティビティ徹底解説  
2019年7月29日

**RPA TIPS**  
Windows 標準 UI 自動化のヒント  
2019年7月29日

**RPA TIPS**  
プロセス爆速化のためのTIPS  
2019年7月29日

#### BLOG CATEGORIES

- Artificial Intelligence 0
- Industry Solutions 0
- Product Release 8
- RPA Tips 6

#### POPULAR ARTICLES

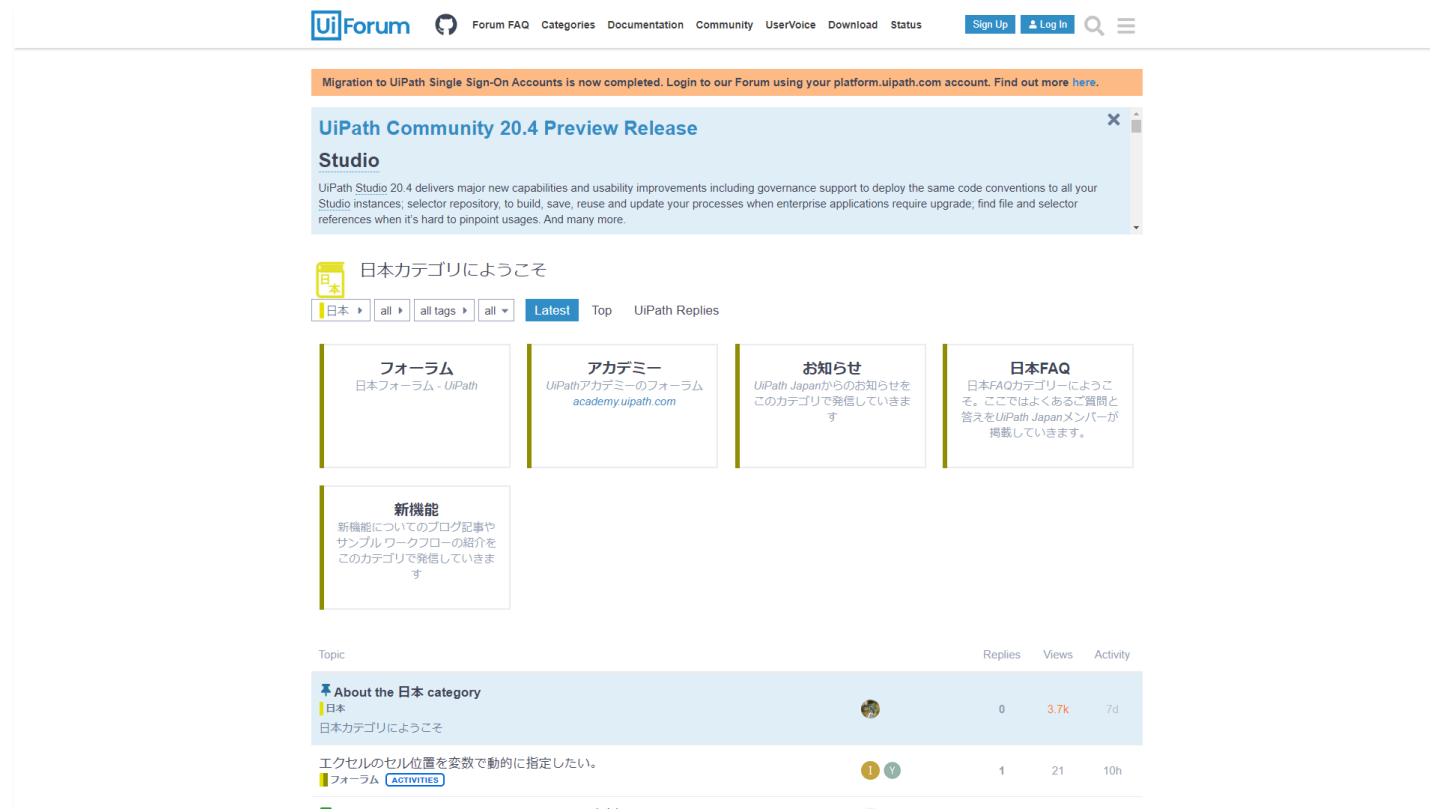
- 最新バージョン「UiPath 2018.3」リリース  
10/31/2018
- UiPath製品の新しい製品リリース方法を紹介  
3/6/2019
- 2019 ファストトラックリリース: オートメーションファースト時代、もっと先へ  
5/16/2019

# 1. 参考リンク

## 1-5. UiPathフォーラム

**UiPath Community Forum** : <https://forum.uipath.com/c/japan>

- 誰もがUiPathに関する質問・回答を投稿できるサイト
- UiPathからの製品に関する公式アナウンス（新機能紹介など）もフォーラムで行われます



# 1. 参考リンク

## 1-6.UiPathアカデミー

UiPath 社が提供する公式のオンライン学習サービス : <https://www.uipath.com/ja/rpa/academy>

- RPA 開発や UiPath Orchestrator に関するコンテンツが充実

The screenshot displays the UiPath Academy website interface. At the top, there is a navigation bar with the 'UiPath Academy' logo and links for 'ホーム', 'コース', '学習プログラム', 'リソース', and 'お問い合わせ'. A 'Login/Sign up' button is visible in the top right corner. The main content area features two course cards. The first card, titled 'RPA スターター ~はじめての UiPath Studio~', describes a course for beginners, mentioning 'RPAの基礎知識・UiPath製品の特徴・RPAに適した業務の選び方・UiPath Studioの開発の基礎を学習します' and 'RPA 概論・開発入門の、2レッスンで構成されています'. It lists '1コース' and includes a '初心者向けコース' graphic with 'RPA概論' and '開発入門' circles, and the text '4時間で 基礎知識 を習得'. The second card, titled 'RPA デベロッパー 基礎プログラム', describes a course for developers, mentioning 'UiPath Studio・UiPath Orchestratorを使用してプロフェッショナルな開発方法を学習します' and 'UiPath Studio開発 初級&中級コース・UiPath Orchestratorコースの、3コースで構成されています'. It lists '3コース' and includes a screenshot of the UiPath Studio interface showing an error handling configuration window. Both cards have a '学習をはじめよう >' button.

# 1. 参考リンク

## 1-7.UiPath無料評価版

UiPath無料評価版 : <https://www.uipath.com/ja/start-trial>

	個人および小規模事業者、その他組織向け <b>Community Cloud</b> <a href="#">Try it</a> 個人のRPA開発者および小規模事業者向け。Enterprise Cloud版にアップグレード可能。	法人向け <b>Studio</b> <a href="#">Try it</a> UiPath Studioを試したい法人の個別開発者向け。	法人向け <b>Enterprise Cloud</b> <a href="#">Try it</a> クラウド版RPA Enterprise Platform全製品。現在はプレビュー版として提供。 ①	法人向け <b>Enterprise Server</b> <a href="#">Try it</a> オンプレミス版Enterprise Automation Platform全製品 ②
無料評価期間	ずっと無料	60日間	60日間	60日間
Studio ①	2ライセンス	1ライセンス	2ライセンス	10ライセンス
付属のRobots	Attended Robots2台; Unattended Robots1台 ②	Attended Robots1台	Attended Robots2台; Unattended Robots1台 ②	Attended Robots10台; Unattended Robots10台; Non-Production Robots10台
追加のRobots	— ①	—	無制限;必要なライセンス数だけ利用可能	無制限;必要なライセンス数だけ利用可能 ⑤
AI Computer Vision	30メガピクセル分	—	240メガピクセル分	お客様のご利用環境に依存
自動化の上限	Robots 3台	—	UiPath Enterprise Cloudに沿ったエンタープライズスケール	お客様のご利用環境に依存
可用性	99.5%+	—	UiPathのサポートと99.5%+以上	高可用性アーキテクチャを利用可能
ユーザー管理	—	—	一元化されたエンタープライズユーザー管理	一元化されたエンタープライズユーザー管理
ソフトウェアの更新	UiPathによる自動更新	お客様による任意の更新	UiPathによる自動更新とお客様による任意の更新 ③	お客様による任意の更新
オンライントレーニングと認定	—	—	—	—
オンサイトトレーニング	—	—	—	—
サポート	UiPathフォーラムの無償サポート ④	SLAに基づく、チケットによるサポート ④	SLAに基づく、チケットによるサポート ④	SLAに基づく、チケットによるサポート ④
アクティベーション	オンラインのみ	オンラインとオフライン	オンラインのみ	オンラインとオフライン

# 1. 参考リンク

## 1-8.RPAガバナンスガイドライン関連資料

2018年以降、RPAの導入/運用体制が監査対象となってきたことを受け  
PwCあたら監査法人 と UiPathが共著でガバナンス構築のためのガイドライン及びハンドブックをリリース  
Orchestrator導入とともに検討すべきガバナンスルールについて解説したハンドブック  
#本ガイドラインは**特定のRPA製品に依存しない**

<https://www.uipath.com/ja/solutions/whitepapers/rpa-governance-handbook>



# 1. 参考リンク

## 1-9. パートナートレーニング

- **UiPath Orchestratorトレーニング**

[https://uipath-event.info/Partners/19\\_0905/UiPath0905.html](https://uipath-event.info/Partners/19_0905/UiPath0905.html)

UiPathがパートナー向けに行ったOrchestratorのトレーニング

主要機能やOrchestratorの構成等、Orchestratorについての基本知識が分かる

※パートナーのみ閲覧可能

Thank you!

---