

業務部門における 外部トリガーを利用した Unattended Robotの活用方法

2021/3/3

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。

用語	略語	説明
Ui Orchestrator	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。
Ui Studio	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）。
Ui Robots	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。

Contents

0. はじめに

- 0-1. 対象者と目的
- 0-2. UiPathロボットの種別
- 0-3. UiPathロボットの特徴
- 0-4. Orchestratorによる中央管理型のUnattended Robot実行
- 0-5. 外部トリガーによるUnattended Robot実行
- 0-6. 外部トリガーによるUnattended Robot実行の社内展開イメージ
- 0-7. 外部トリガーによるUnattended Robot実行の適用業務範囲例

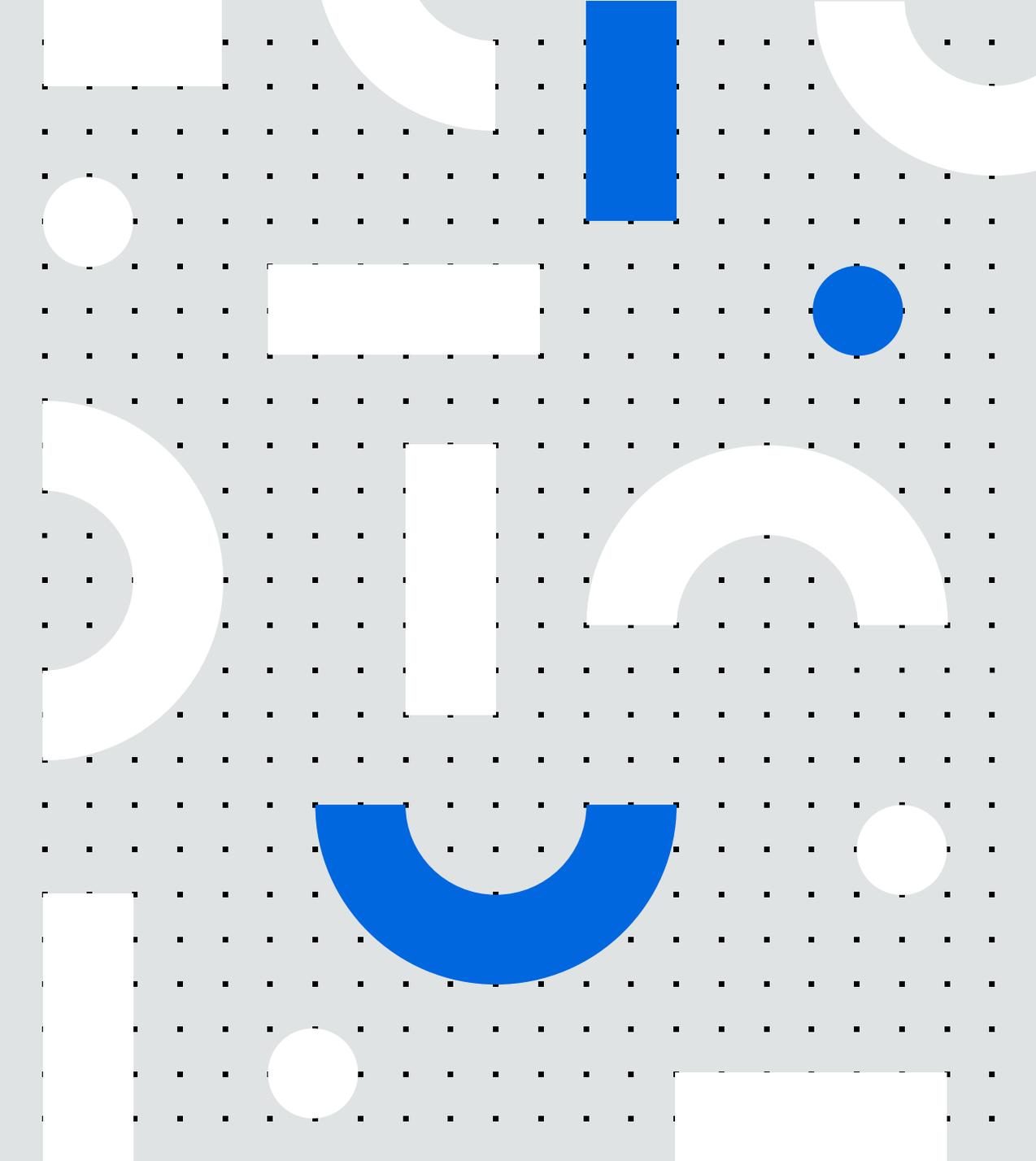
1. Unattended Robotの実行方法

- 1-1. 時間（タイム）トリガー
- 1-2. キュートリガー
- 1-3. イベント監視によるトリガー（メールトリガー）
- 1-4. イベント監視によるトリガー（フォルダ監視トリガー）
- 1-5. MicrosoftのPower Automateトリガー
- 1-6. PowerShellトリガー
- 1-7. Appsトリガー

2. 各種トリガーの特徴

3. 環境設定（参考）

0. はじめに



0. はじめに

0-1. 本資料の対象者と目的

対象者 Robotを活用して自動化領域の更なる拡大を期待されている、エンドユーザー様とパートナー様

目的 OrchestratorによるUnattended Robotの実行方法、特に業務ユーザーが**任意のタイミングで実行できるようになる外部トリガーについて理解する**

OCによる中央管理型のUR実行 (例：タイムトリガー)

Orchestratorによる中央管理のもと、
大量処理を無人で24時間いつでもスケジュール実行可能

外部トリガーによるUR実行

業務ユーザーも
任意のタイミングでRPAを間接的に実行可能

<得られる効果>

1. 業務ユーザーが任意のタイミングでRPAを間接的に実行可能
2. URライセンスの有効活用
3. 自動化領域の更なる拡大
4. RPAを知らなくてもRPAによる自動化を体験できる

0. はじめに

0-2. UiPathロボットの種別

Robotには人（業務ユーザー）の操作によってワークフローを起動し、ジョブを実行するタイプのロボット「**Attended Robot**」と主にOrchestratorのスケジューラー設定によってワークフローを起動し、ジョブを実行するタイプのロボット「**Unattended Robot**」の2種類

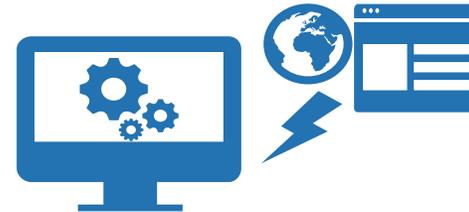


Attended Robot :

人の操作によってワークフローを起動し、ジョブを実行するタイプのロボット



PCタスクトレイからユーザーが任意のタイミングでワークフローを起動、実行



Unattended Robot :

Orchestratorのスケジューラー設定によってワークフローを起動し、ジョブを実行するタイプのロボット



Orchestrator上のスケジュール設定でワークフローを起動、実行

Attended RobotとUnattended Robotの比較

比較観点	 Attended Robot	 Unattended Robot
概念	人の操作によってワークフローを起動し、処理を実行するロボット	人の介在なし※に処理を実行できるロボット ※どうしても一時的に人の介在が必要な場合は Action Centerでカバーする
自動化対象業務	向いている業務： 個人の作業～部門の業務の範囲、デスクトップ上の作業自動化 使用が望ましいケース <ul style="list-style-type: none"> ・ 人間からの入力や意思決定が必要 ・ 実行中、人間による監視が必要 ・ 定期的に実行スケジュール設定できない 業務例： サービスデスク、ヘルプデスク、コールセンター	向いている業務： 部門の業務～全社の業務の範囲、業務プロセスなどの複雑な自動化 使用が望ましいケース <ul style="list-style-type: none"> ・ AR必須以外の場合はUR推奨 ・ 同時複数の処理実行 ・ トリガーによる処理実行 業務例： 各種バックオフィス業務
開発難易度	URと比較して、考慮すべき点が少ない	無人実行のため、以下の観点で考慮すべき <ul style="list-style-type: none"> ・ 入出力、エラーハンドリング、実行前初期化処理、実行後終了処理
導入効果	削減時間 人の監視が必要でかつ、URと比べて自動化範囲が狭いため、削減できる時間小さい	削減時間 無人実行できるかつ、業務に跨いだ自動化が可能のため、削減できる時間大きい

Attended RobotとUnattended Robotの比較

比較観点	 Attended Robot	 Unattended Robot
開発・実行	<p>入出力</p> <ul style="list-style-type: none"> 対話型で人による入出力行うことが可能 <p>エラーハンドリング</p> <ul style="list-style-type: none"> 実行者が手動でエラー回避することができる。 エラーが発生した場合、内容を確認し、再時刻など処置することができる <p>実行前実行後の処理</p> <ul style="list-style-type: none"> 実行者が変わりに準備することが可能 	<p>入出力</p> <ul style="list-style-type: none"> 無人実行のため、ファイルやメールなど明確に入出力データを受け渡しする必要がある <p>エラーハンドリング</p> <ul style="list-style-type: none"> エラー発生した場合、通知やリカバリー含め詳細な設計、実装が必要 エラー解決のために、ログ、スクリーンショット取得が必要 <p>実行前実行後の処理</p> <ul style="list-style-type: none"> ファイル作成、削除、アプリケーション開始終了などが必要
運用	<p>実行方法</p> <ul style="list-style-type: none"> UiPath Assistantまたはコマンドラインから実行 スケジュール実行はライセンス違反 <p>※ただし、Windowsタスクスケジューラ実行のうち、直接ユーザの操作によるAR起動はライセンス違反ではない（サポート対象外）</p>	<p>実行方法</p> <ul style="list-style-type: none"> Orchestratorから実行 スケジュール実行が可能 <p>※OCの機能使った場合は技術サポート対象、Windowsタスクスケジューラ使った場合は技術サポート対象外</p> <p>・さまざまなトリガー実行が可能（本テーマ）</p>

0. はじめに

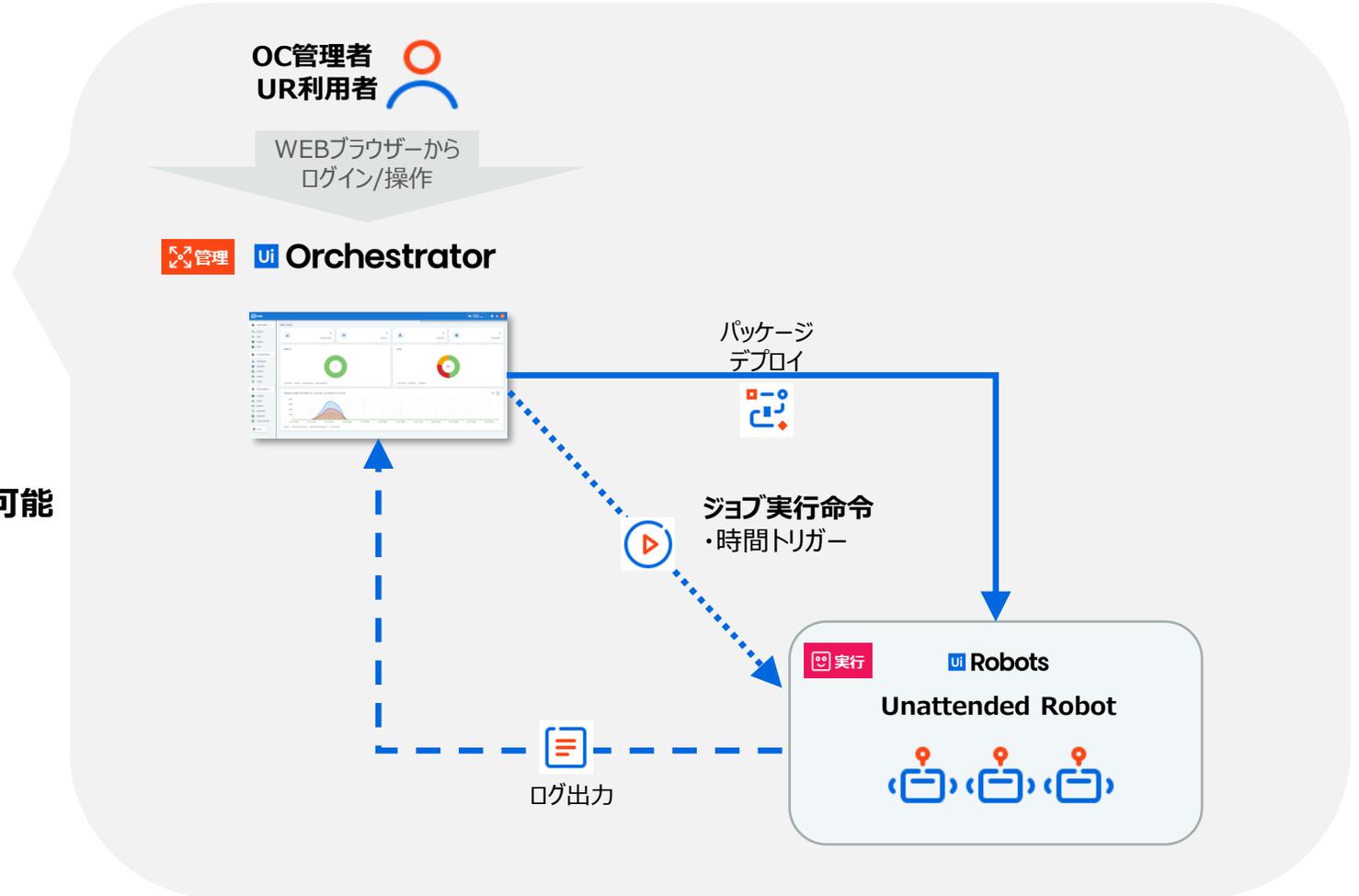
0-4. Orchestratorによる中央管理型のUnattended Robot実行

OrchestratorによるUnattended Robot実行の簡易イメージは下記のとおり

**OCによる中央管理型のUR実行
(例：タイムトリガー)**

Orchestratorによる中央管理のもと、
大量処理を無人で24時間いつでもスケジュール実行可能

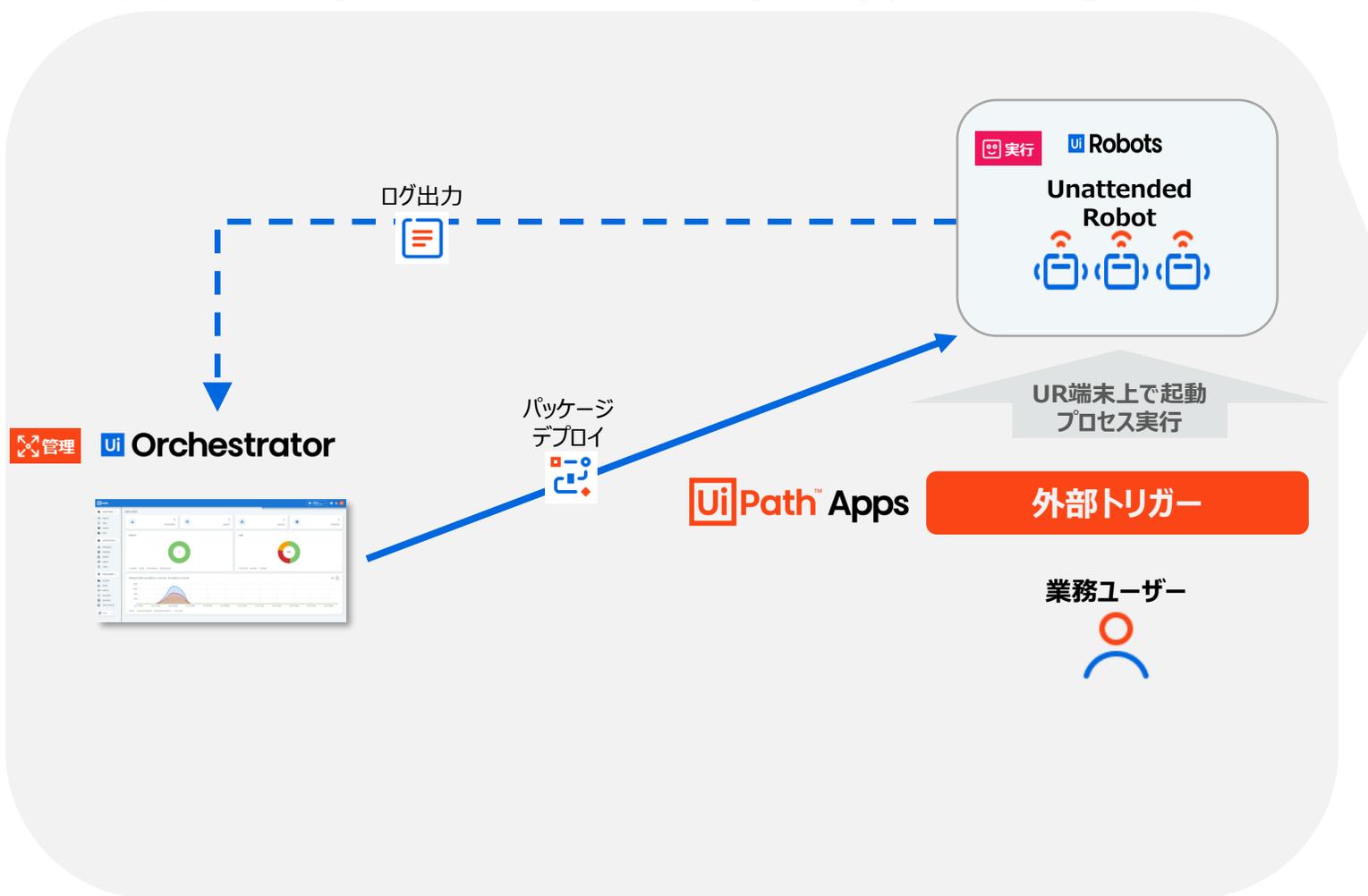
- 人の介入を必要とせず、24時間いつでもスケジュール実行可能
- 同時複数の大量処理実行
- 業務に跨いだ自動化が可能のため、削減できる時間大きい
- CoEの統制管理下のもと利用可能



0. はじめに

0-5. 外部トリガーによるUnattended Robot実行

外部トリガーによるUnattended Robot実行の簡易イメージは下記のとおり



外部トリガーによるUR実行

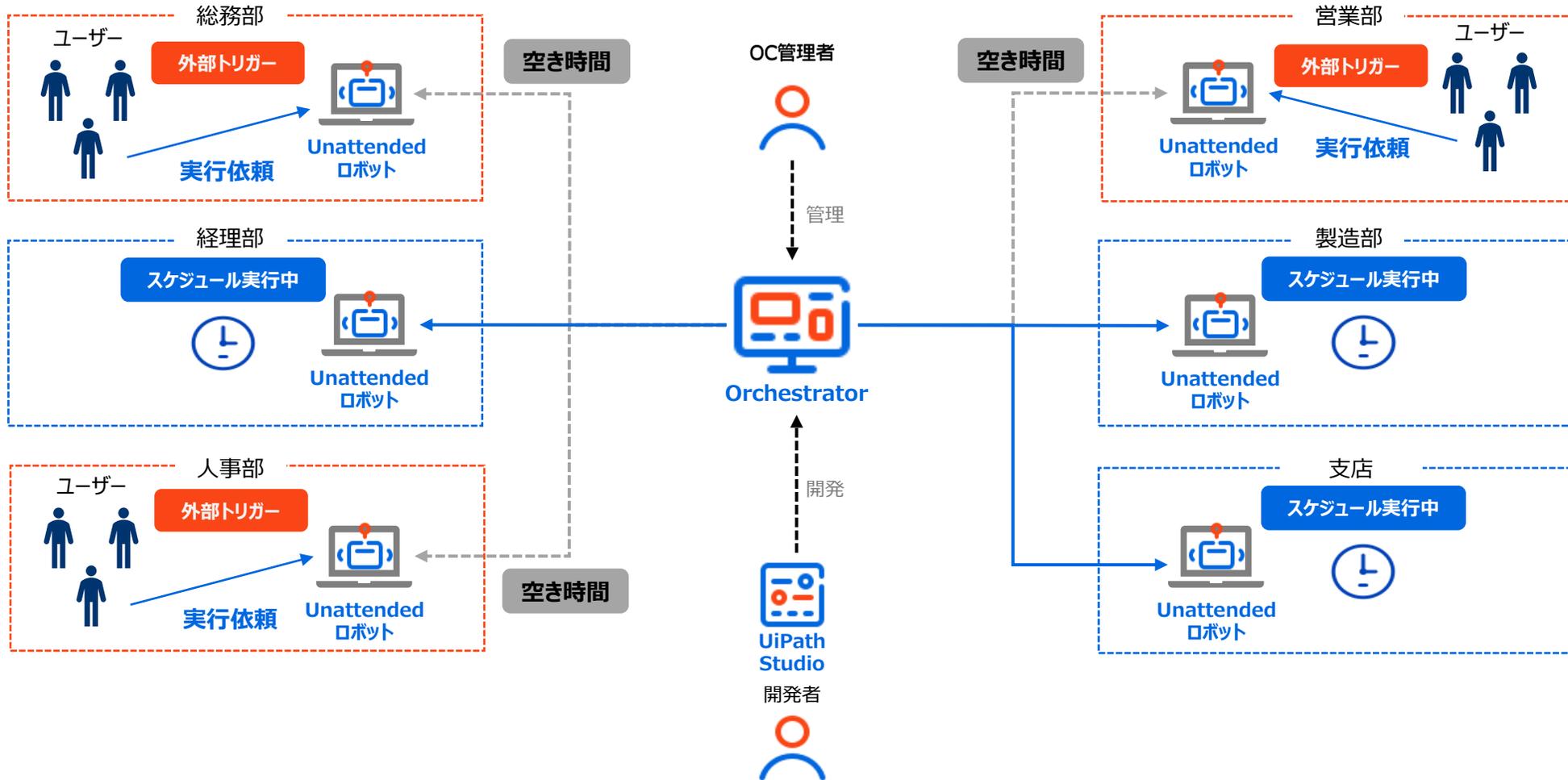
業務ユーザーも
任意のタイミングでRPAを間接的に実行可能

- 業務ユーザーが**任意のタイミング**でRPAを間接的に実行
- URライセンスの**有効活用**
- 自動化領域の更なる**拡大**
- RPAを知らなくてもRPAによる**自動化を体験**できる

0. はじめに

0-6. 外部トリガーによるUnattended Robot実行の社内展開イメージ

Unattended Robotを社内に広く展開し、ユーザーが外部トリガーを利用してUnattended Robotを実行することによって、ライセンスの有効活用が可能となり、**得られる費用対効果が更なる高くなる**



0. はじめに

0-7. 外部トリガーによるUnattended Robot実行の適用業務範囲例

外部トリガーによるUnattended Robot実行の適用業務範囲は多岐にわたります。

営業活動	顧客管理	見積管理	情報収集	資料作成
	商品管理	受注管理	請求管理	アフターフォロー
営業業務	レポートニング	経費申請	勤怠登録	その他事務
人事業務	人事異動	退職	給与・賞与	勤怠管理
	福利厚生	身上変更	安全衛生	労使関係
	表彰・懲戒	ヘルプ・デスク		
経理業務	売掛債権管理	買掛債務管理	在庫管理	固定資産管理
	原価管理	経費管理	管理会計業務	決算業務
	外部開示業務	税務計算・申告業務	税務調査対応	
財務業務	現金出納管理	手形管理	有価証券管理	債務保証管理
	貸付金管理	借入金管理	社債管理	デリバティブ取引管理
	外貨建取引管理	資金管理	資産流動化業務	

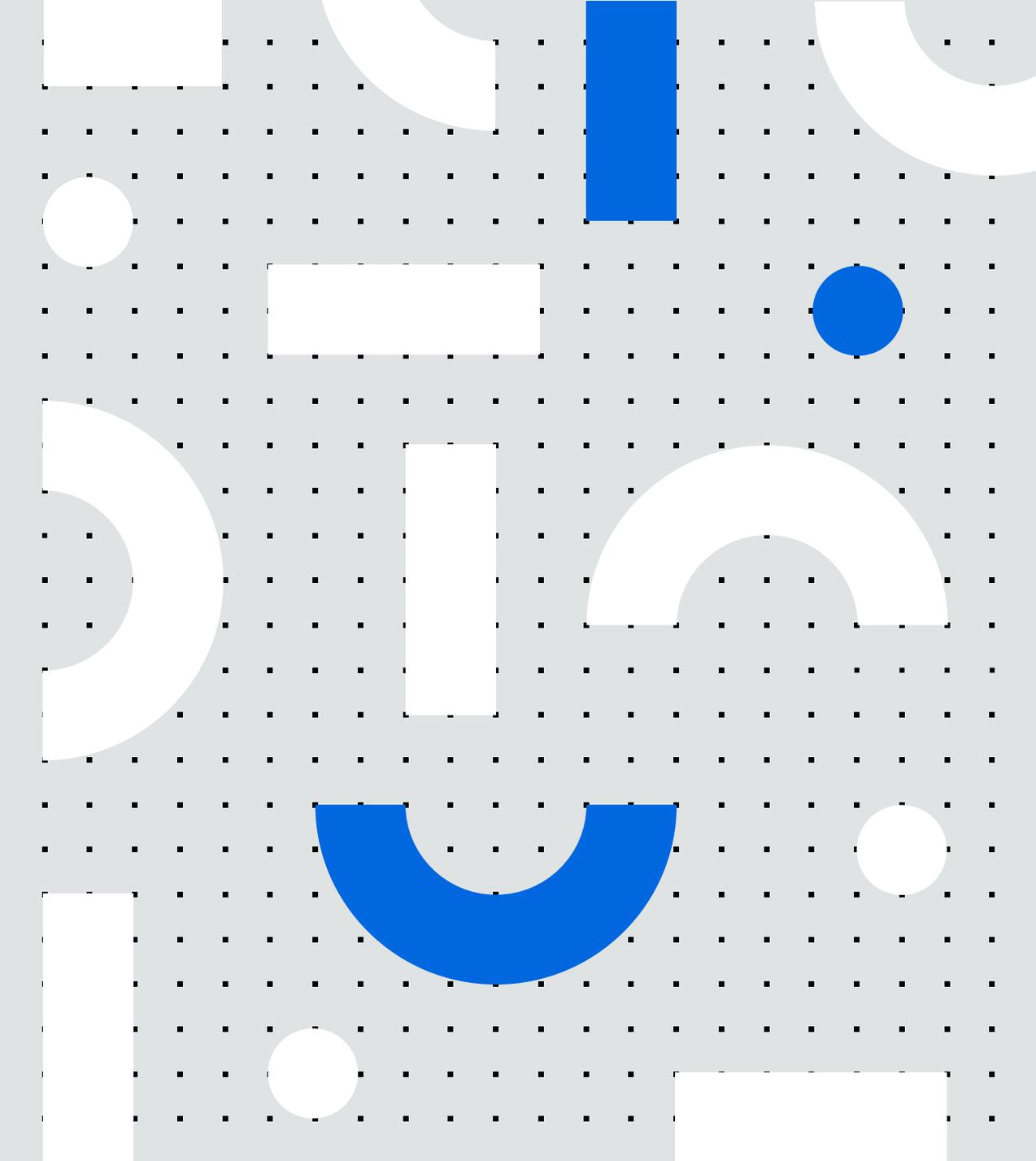
凡例：RPA適用度合い

High

Middle

Low

1. Unattended Robotの実行方法



1. Unattended Robotの実行方法

■ OCによる中央管理型のUR実行

1-1. 時間（タイム）トリガー

■ 外部トリガーによるUR実行

1-2. キュートリガー

1-3. イベント監視によるトリガー（メールトリガー）

1-4. イベント監視によるトリガー（フォルダ監視トリガー）

1-5. MicrosoftのPower Automateトリガー

1-6. PowerShellトリガー

1-7. Appsトリガー



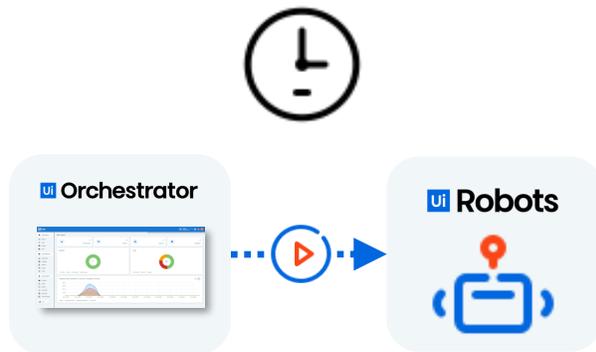
1. Unattended Robotの実行方法

1-1. 時間（タイム）トリガー

Orchestratorからスケジュールを設定することで、**設定した時刻に自動的にURが実行される中央管理された実行方法**

メリット

- 大量処理を無人で24時間いつでもスケジュール実行可能
- 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



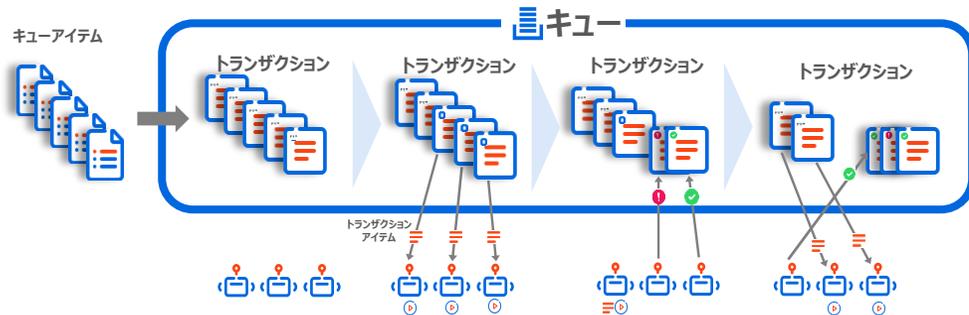
1. Unattended Robotの実行方法

1-2. キュートリガー（外部トリガー）

Orchestratorに中央管理され、『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できる

メリット

- 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
- 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



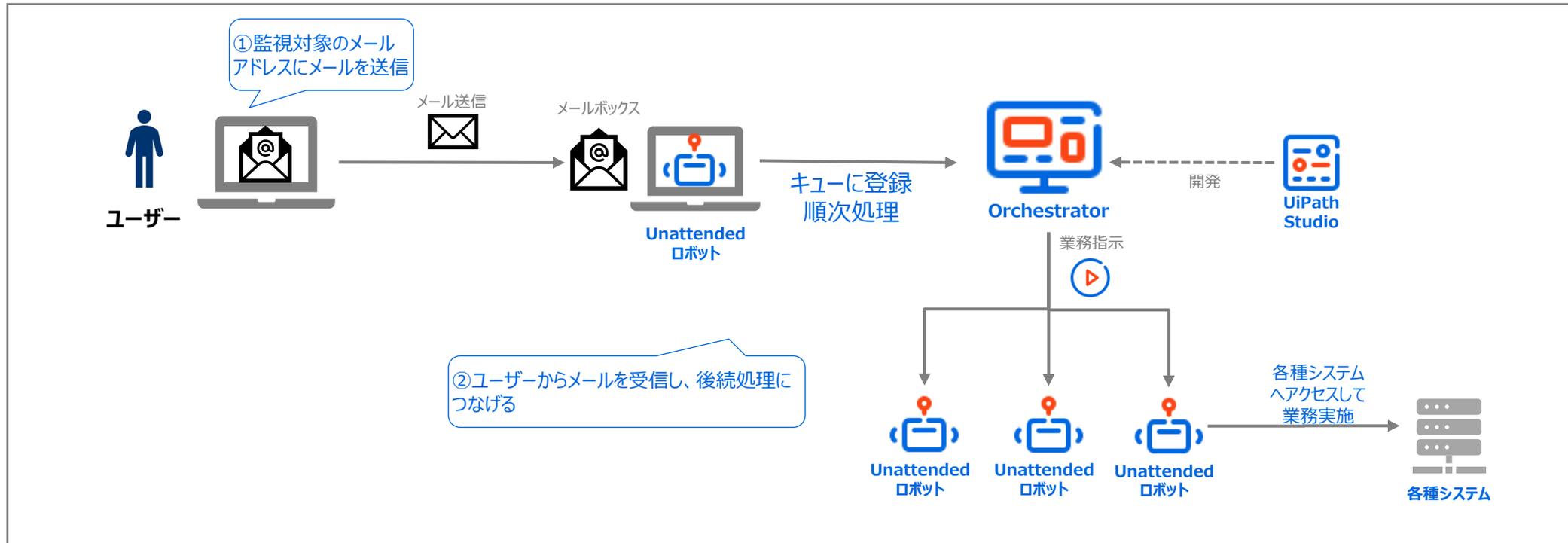
1. Unattended Robotの実行方法

1-3. イベント監視によるメールトリガー（外部トリガー）

Unattended Robotにて、メールボックスを監視させ、現場の業務ユーザーからのメールをトリガーにして、後続処理につなげる

メリット

- 現場の業務ユーザーが任意のタイミングでメールを送信することでURに実行命令することが可能



Get Outlook Mail MessagesやSave Attachmentsアクティビティ
<https://docs.uipath.com/activities/lang-ja/docs/get-outlook-mail-messages>
<https://docs.uipath.com/activities/lang-ja/docs/save-mail-attachments>

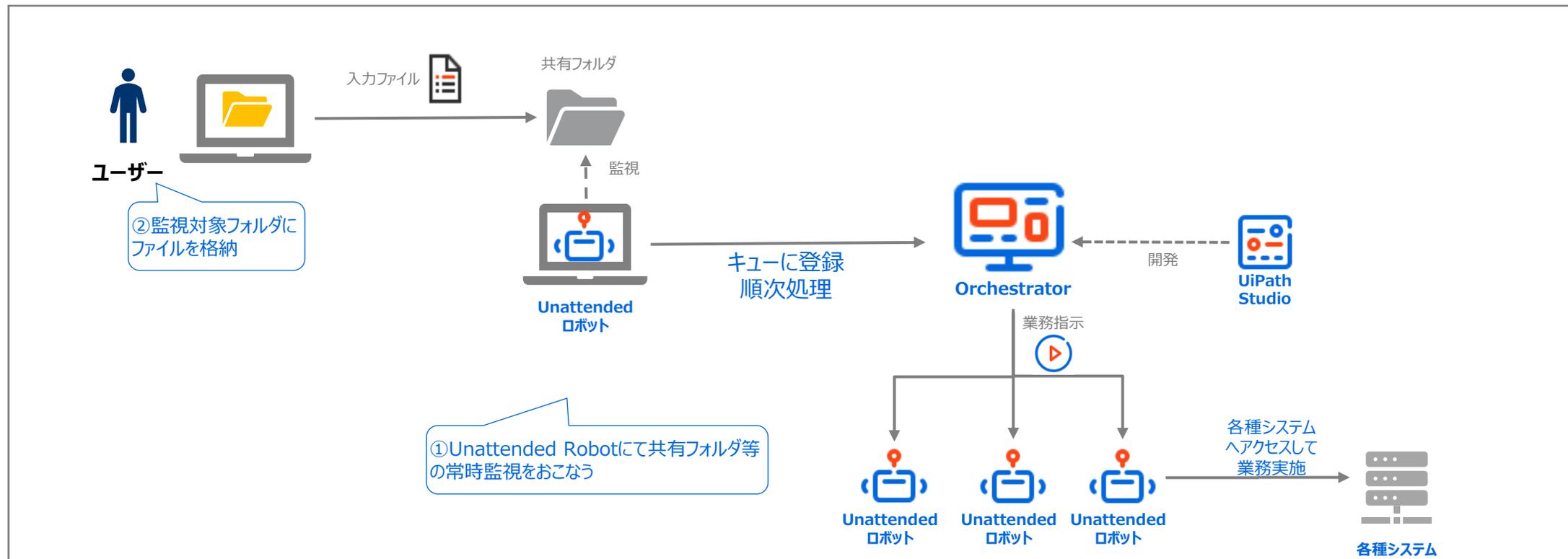
1. Unattended Robotの実行方法

1-4. イベント監視によるファイル変更トリガー（外部トリガー）

Unattended Robot にて、共有フォルダ等を監視させ、現場の業務ユーザーからのファイル格納などの処理をトリガーにして、後続処理につなげる

メリット

- 現場の業務ユーザーが任意のタイミングでフォルダにファイルを格納することでURに実行命令することが可能



ファイル変更トリガー (File Change Trigger)

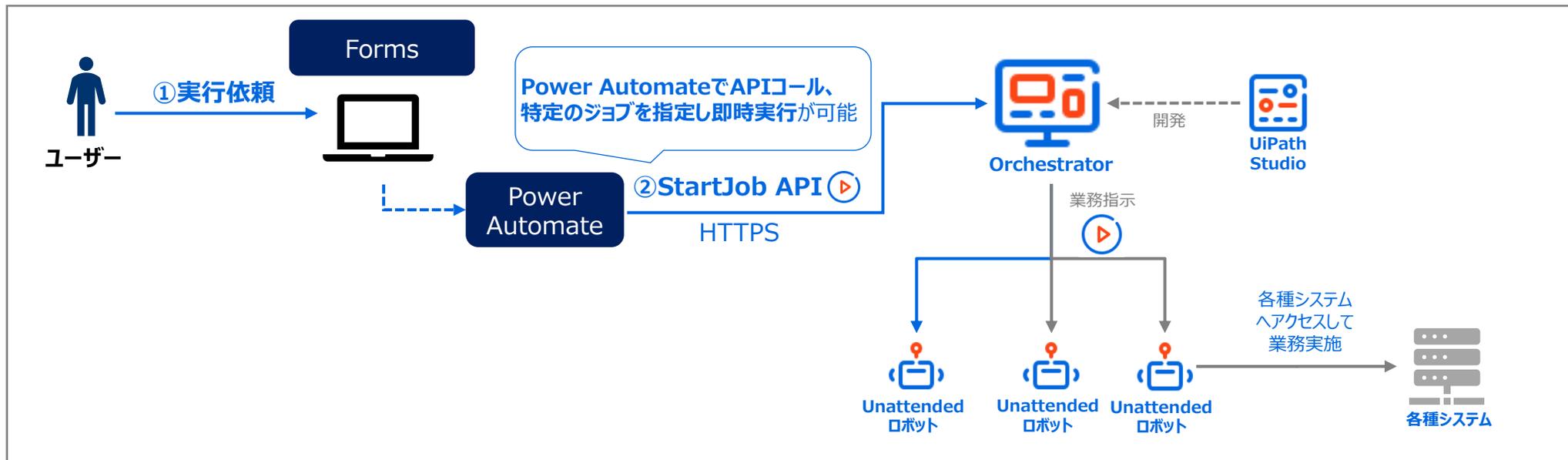
<https://docs.uipath.com/activities/lang-ja/docs/file-change-trigger-v-2>

1. Unattended Robotの実行方法

1-5. MicrosoftのPower Automateトリガー（外部トリガー）

外部システム（Power Automate）からAPI経由でOrchestratorにプロセス実行命令（StartJob API）をおこなう
メリット

- 現場の業務ユーザーが任意のタイミングでPower Automateを利用してURに実行命令することが可能
- MicrosoftのForms（フォーム）を利用時に本人認証による実行ユーザーの特定が可能
- Power AutomateからOrchestrator APIを呼び出し、ジョブの即時実行が可能
- フォーム提出時に、Power Automateを利用し本人確認メールを自動送信する仕組みを簡単に実装可能



Microsoft Flow と UiPath Orchestrator の連携方法

<https://www.uipath.com/ja/blog/developer/microsoftflow-uipathorchestrator-connection>

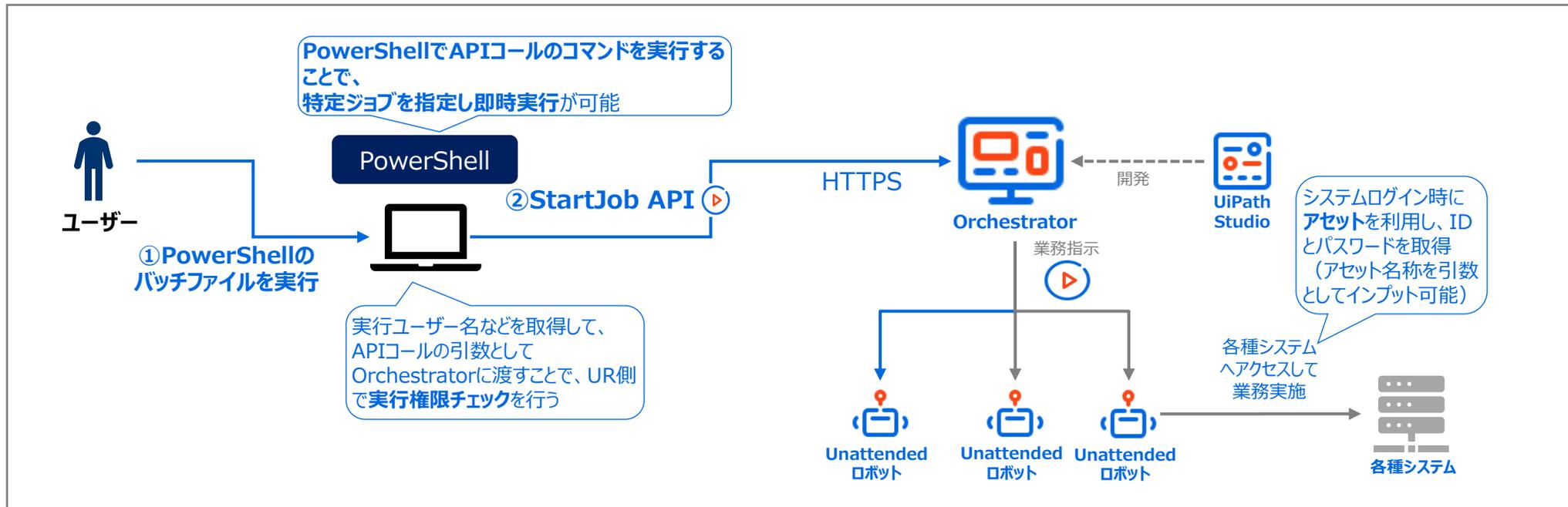
1. Unattended Robotの実行方法

1-6. PowerShellトリガー（外部トリガー）

外部アプリケーション（PowerShell）からAPI経由でOrchestratorにプロセス実行命令（StartJob API）をおこなう

メリット

- 現場の業務ユーザーが任意のタイミングでPowerShellを利用してURに実行命令することが可能
- PowerShellを利用時に実行したドメインユーザーの特定が可能
- PowerShellを実行することでOrchestrator APIを呼び出し、ジョブの即時実行が可能
- 引数の受渡しも可能



Orchestrator API PowerShell サンプルスクリプト

<https://www.uipath.com/ja/resources/knowledge-base/sample-script-orchestrator-api-powershell>

1. Unattended Robotの実行方法

1-7. Appsトリガー（外部トリガー）

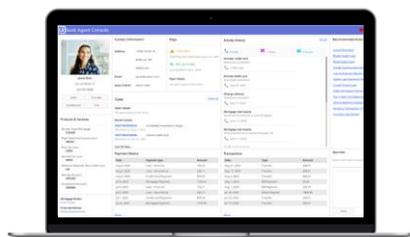
新製品であるAppsを利用して、Orchestratorを経由してURに実行命令をおこなう（21年2月時点ではCloud Orchestratorのみ対応）

メリット

- 現場の業務ユーザーが任意のタイミングでAppsを利用してURに実行命令することが可能
- ウェブインターフェース、ドラッグ & ドロップコントロールを使用して、視覚的に魅力的なプロフェッショナルアプリをデザイン可能
- アプリ開発のコストと複雑さを軽減
- RPAでデータに接続するアプリ、クラウド、オンプレミス、レガシー、APIなしで利用可能
- エンタープライズグレードのアプリケーションから、時間を節約するためのシンプルなフォームまで、すべてを構築することが可能
- すぐに稼働し、1クリックでビジネスユーザーと簡単にアプリを共有可能
- 引数の受渡などもスムーズに実装可能

UiPath Apps

業務ごとに最適なインターフェイスを手軽に開発、誰でも自動化を使いこなすことを可能に



Unattended
ロボット

UiPath Robots

既存のビジネスシステム



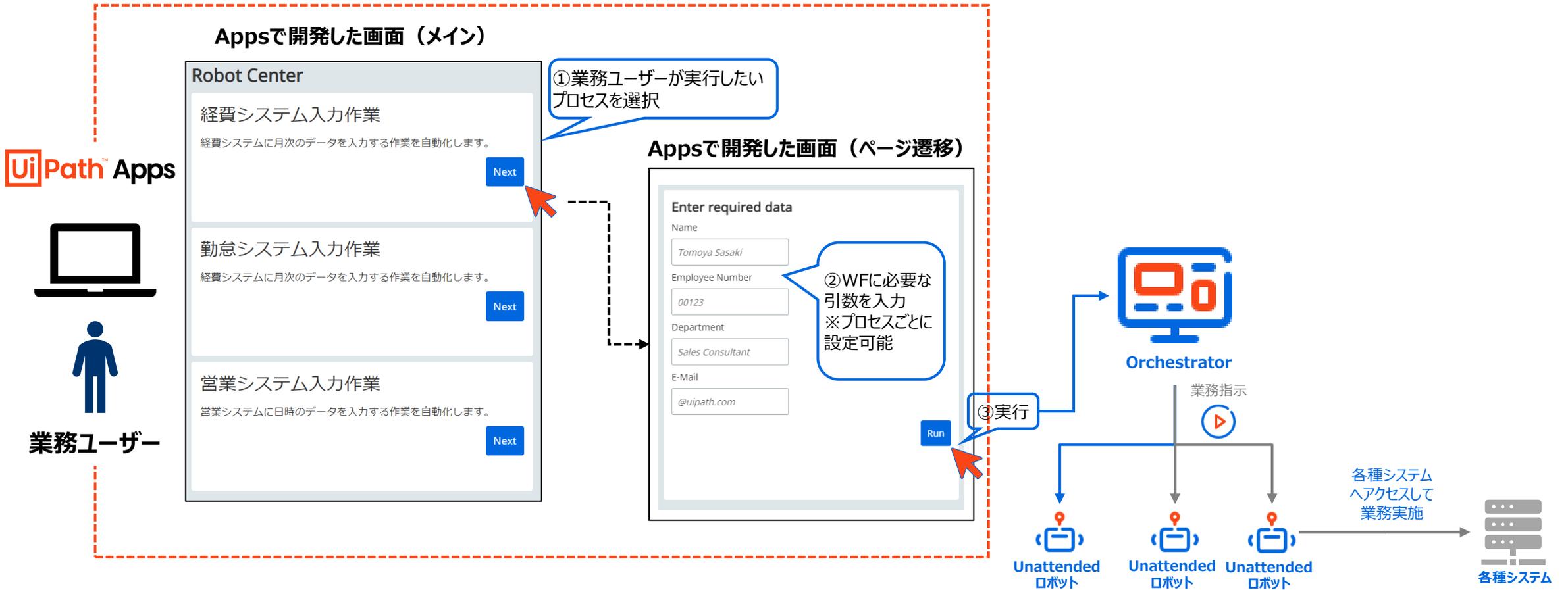
各種システム
ユーザーデータベース、財務会計データ
ベース、文書データベースなど分散した
システムと接続可能

簡単にシームレスな操作でRPAを使いこなし、現場業務ユーザーの自動化領域の拡大に

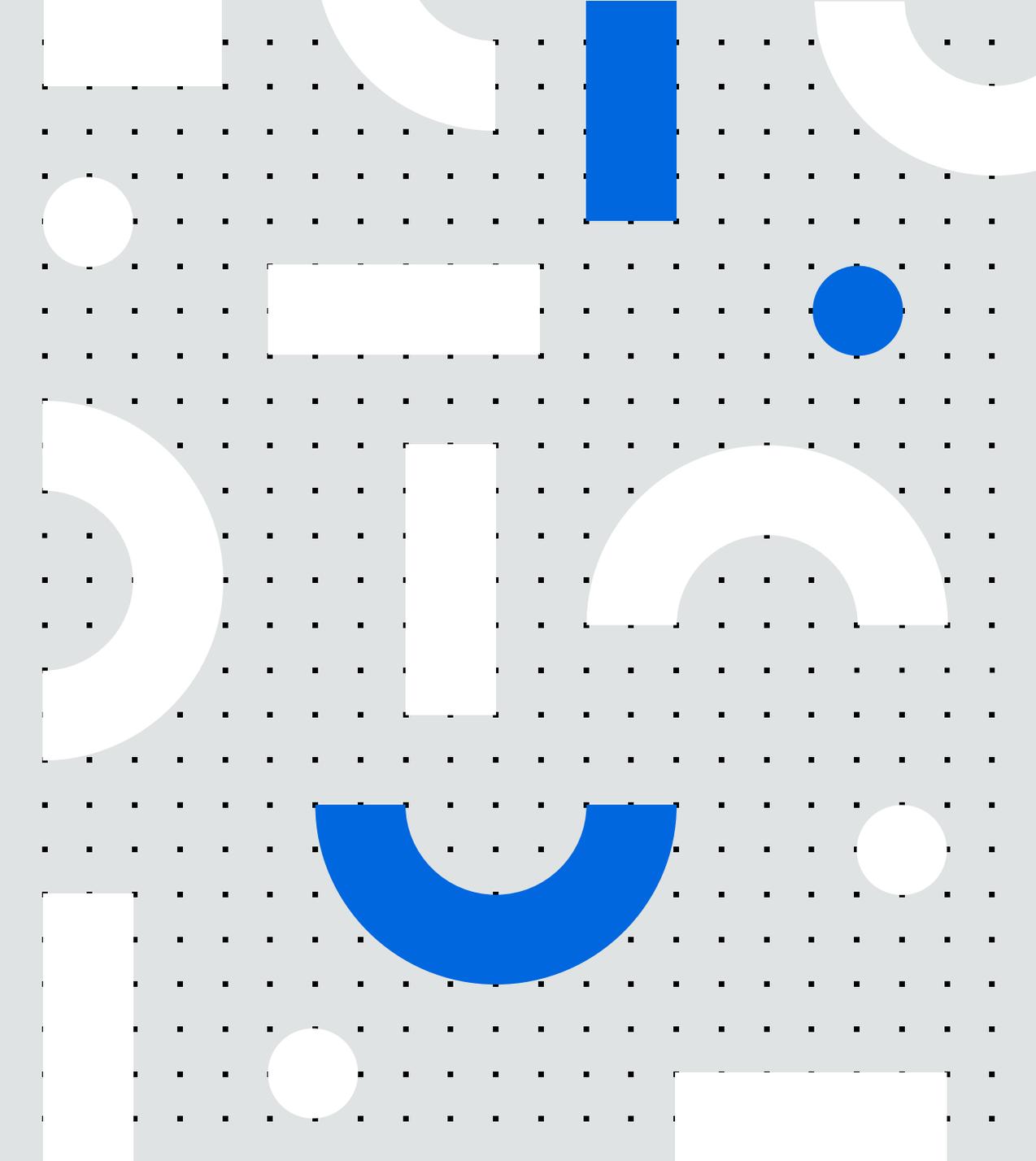
1. Unattended Robotの実行方法

1-7. Appstrigger (外部トリガー)

現場業務ユーザーが任意のタイミングでプロセスを選択し、Orchestratorを経由してURに実行命令をおこなう



2. 各種トリガーの特徴



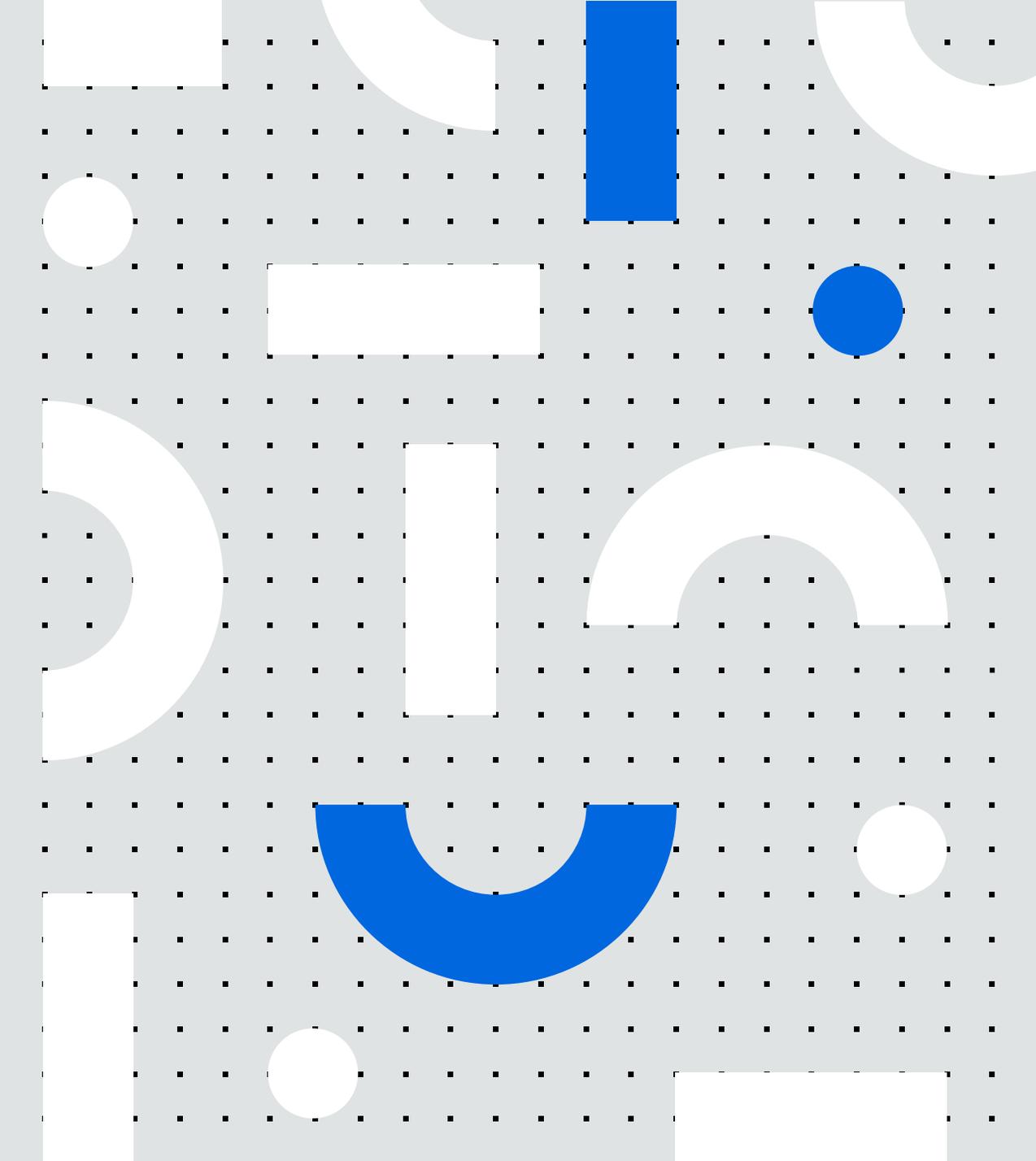
2. 各種トリガーの特徴（参考）

本資料でご紹介したUnattended Robotをトリガー実行した場合のそれぞれの特徴は下記のとおり

	実装の容易さ	ユーザーによる 任意のタイミン グでの利用	ユーザーの 使い勝手	Orchestrator	導入・運用 の時間	ファイルの受渡	引数の受渡
1-1. タイムトリガー	◎	-	-	オンプレ /クラウド	◎	共有フォルダ等	-
1-2. キュートリガー	○	-	-	オンプレ /クラウド	◎	共有フォルダ等	-
1-3. メールトリガー	○	◎	○	オンプレ /クラウド	○	添付 ファイルとして 取扱い可能	○
1-4. フォルダ監視トリガー	○	◎	○	オンプレ /クラウド	○	入力ファイルとして 取扱い可能	○
1-5. Power Automate トリガー	△	◎	△	クラウド推奨	○	共有フォルダ等	○
1-6. PowerShell トリガー	△	◎	△	オンプレ /クラウド	△	共有フォルダ等	○
1-7. Appsトリガー	◎	◎	◎	クラウド ※21年中に オンプレ対応予定	◎	共有フォルダ等	◎
ユーザーによる、 Orchestrator 即時実行	◎	◎	△	オンプレ /クラウド	△	共有フォルダ エンティティ等	○

Appendix.

3. 環境設定 (参考)



Appendix.

3. 環境設定（参考）

3-1. システム構成例

3-2. Unattended Robotライセンスについて

3-3. Orchestratorの設定について ~モダンフォルダの設定~

3-4. Orchestratorの設定について ~引数の設定~

3-5. Orchestratorの設定について ~アセットを活用した実行権限のチェック~

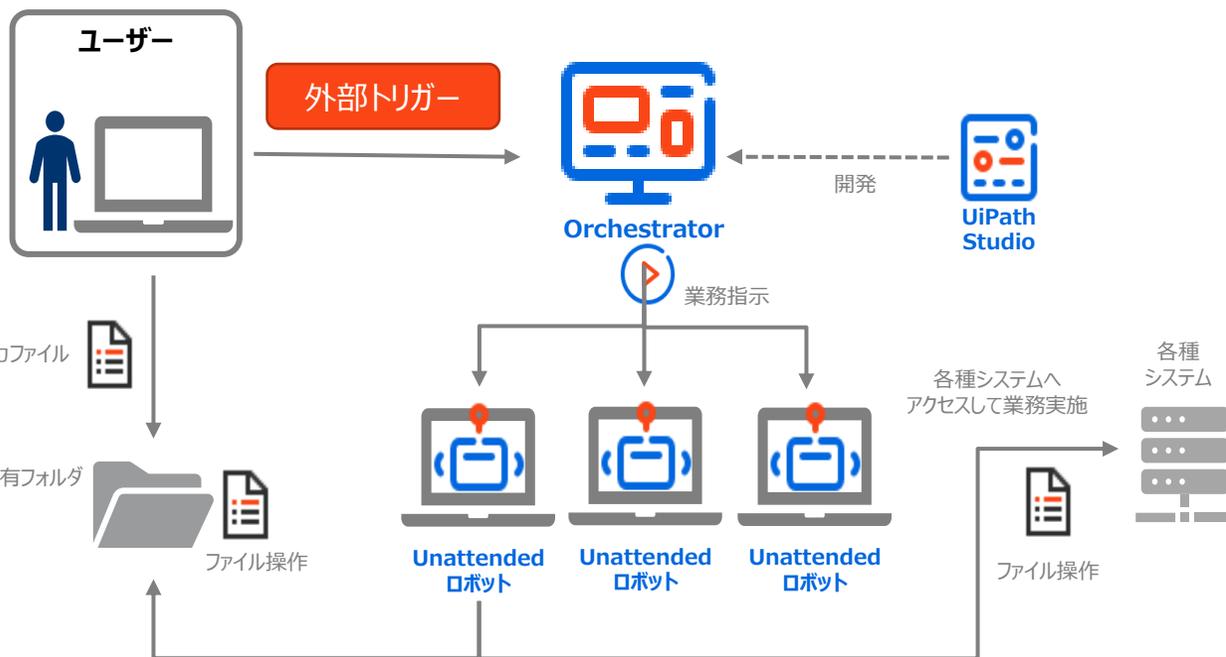
3-6. Cloud OrchestratorにおけるAPIリクエスト

3-7. Power Automateにおけるワークフロー例

3. 環境設定 (参考)

3-1. システム構成例

外部トリガーにてUnattended Robotを実行する際のシステム構成例は下記のとおり



システム構成例

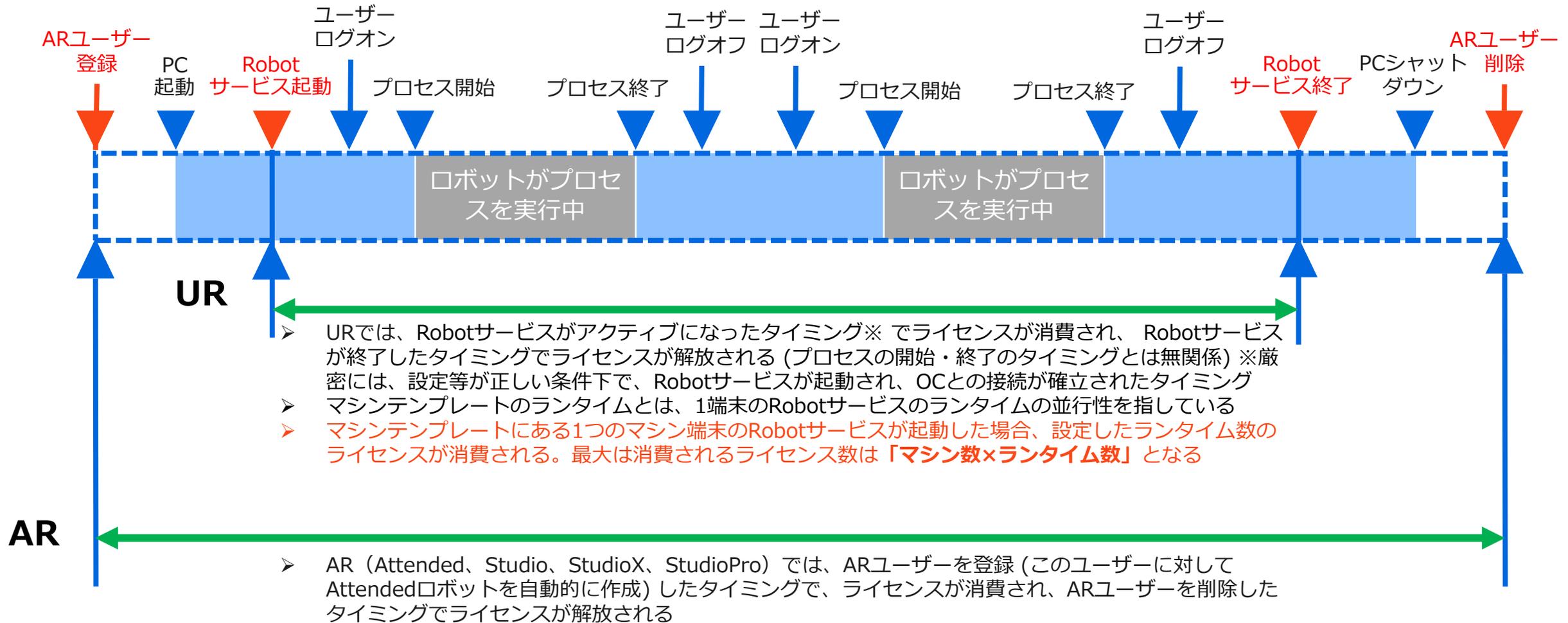
- Orchestrator (オンプレミス/クラウド)
- Unattended Robot
※同時実行数分
- Studio
- 共有フォルダ (ユーザーとURがアクセスできるファイルサーバー等)
- 外部トリガー (Power Automate, Office365等)

各種ハードウェア要件、ソフトウェア要件はドキュメントポータルよりご確認ください。
<https://docs.uipath.com/lang-ja>

3. 環境設定 (参考)

3-2. Unattended Robotライセンスについて

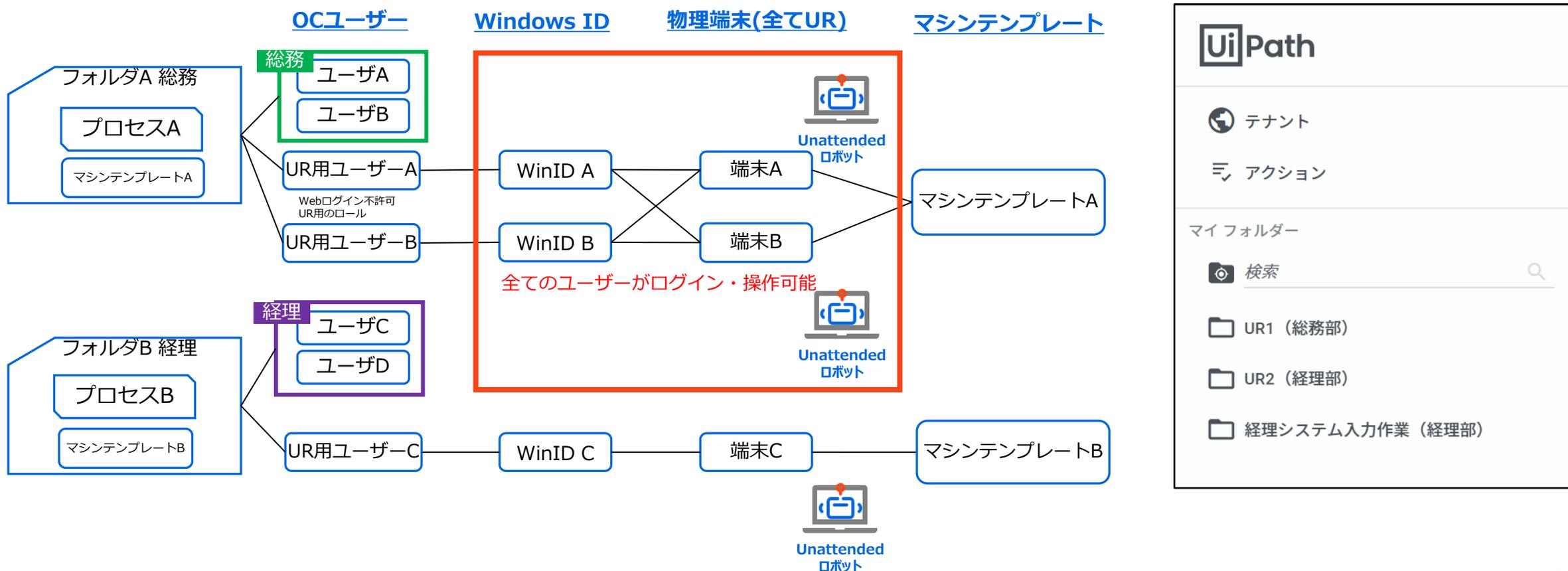
AR、URライセンス利用数カウント(消費と解放)の考え方は以下のとおり



3. 環境設定 (参考)

3-3. Orchestratorの設定について ~モダンフォルダの設定①~

外部トリガー実行の場合、利用する業務ユーザーが増え、自動化対象システムにおけるアセット管理（資格情報）が複雑になることに加えて、URに対するプロセスの同時実行数を適切にするため、Orchestrator（モダンフォルダ）利用では、フォルダを細かく分けて作成することを推奨



3. 環境設定（参考）

3-3. Orchestratorの設定について ~モダンフォルダの設定②~

外部トリガー実行におけるOrchestrator（モダンフォルダ）の推奨設定について、利用する業務ユーザーが増え、対象システムにおけるアセット管理等が複雑になることが予想されるため、フォルダを細かく分けて作成することを推奨

部署単位、URユーザー単位でフォルダを作成

UR1 (総務部)

UR2 (経理部)

経理システム入力作業 (経理部)

ホーム オートメーション 監視 キュー アセット ストレージ バケット テスト アクション カタログ

アクセス許可 Machines

検索

ユーザー名	ロボットの種類	ロール
Unattended Robot 1	Unattended	Robot

Unattended Robot用のロールを付与

URライセンスを付与

項目 10 1-1/1

3. 環境設定 (参考)

3-4. Orchestratorの設定について ~引数の設定~

外部トリガーにて引数を受け渡す際のイメージと手順は下記のとおり



■ WF内での引数設定例

引数 (名前)	方向	引数の型
Arguments	入力	String

様々な外部トリガー

UiPath Apps



PowerShell

etc...



Orchestrator



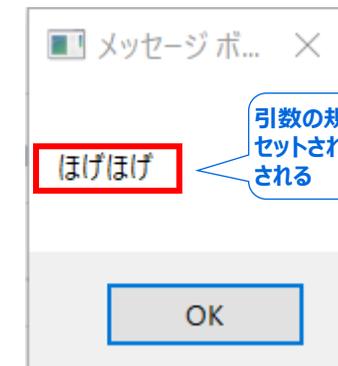
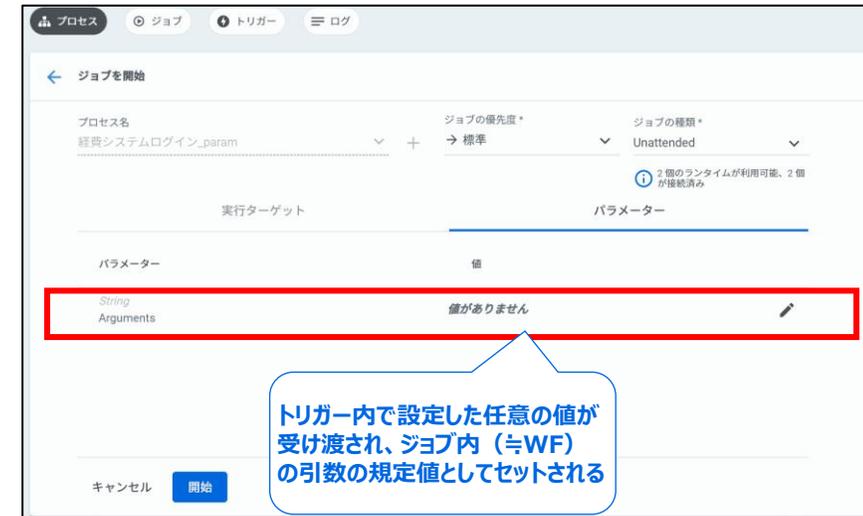
Unattended
ロボット



WF

APIコールの引数をOrchestratorに渡す
引数名 : Arguments
値 : トリガー内で任意の値を入力
(例 : "ほげほげ"と入力)

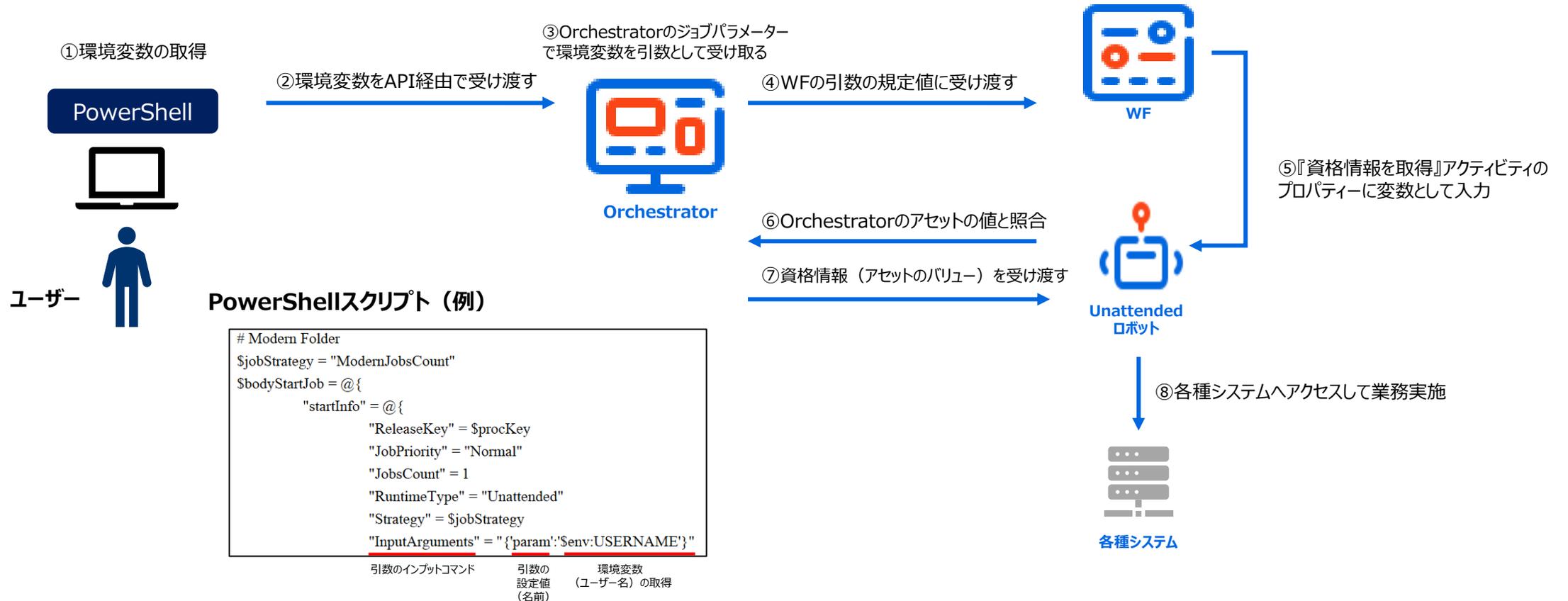
■ Orchestrator内でのJobパラメーター画面



3. 環境設定（参考）

3-5. Orchestratorの設定について ~アセットを活用した実行権限のチェック①~

外部トリガー実行時に、利用ユーザーの特定方法とアセットの活用イメージは下記のとおり



3. 環境設定（参考）

3-5. Orchestratorの設定について ~アセットを活用した実行権限のチェック②~

外部トリガー実行時に、利用ユーザーの特定方法とアセットの活用イメージは下記のとおり



名前	説明	種類	値
ユーザー-A		Credential	[資格情報ストア内]
ユーザー-B		Credential	[資格情報ストア内]

ユーザー（外部トリガーにてURを実行するユーザー）毎にシステム等にログインする際の資格情報を登録する。
その際、PowerShellなどで環境変数を取得したときの値と一致させる必要がある。
（例：ユーザー名、メールアドレス、社員番号など）

プロパティ: UiPath.Core.Activities.GetRobotCredential

- その他
 - Orchestrator フォルダ パス: "UR1 (総務部)"
 - プライベート:
- 入力
 - アセット名: param
- 共通
 - タイムアウト (ミリ秒): アクティビティの実行
 - 表示名: 資格情報を取得
- 出力
 - パスワード: Password

引数：param
Orchestratorのジョブパラメーターで環境変数（ユーザー名など）を引数として受け取ったのち、資格情報を取得アクティビティのアセット名として受け渡す。

3. 環境設定 (参考)

3-6. Cloud OrchestratorにおけるAPIリクエスト

Cloud OrchestratorにおけるAPI アクセス情報の取得方法は下記のとおり

※Cloud Orchestrator利用想定



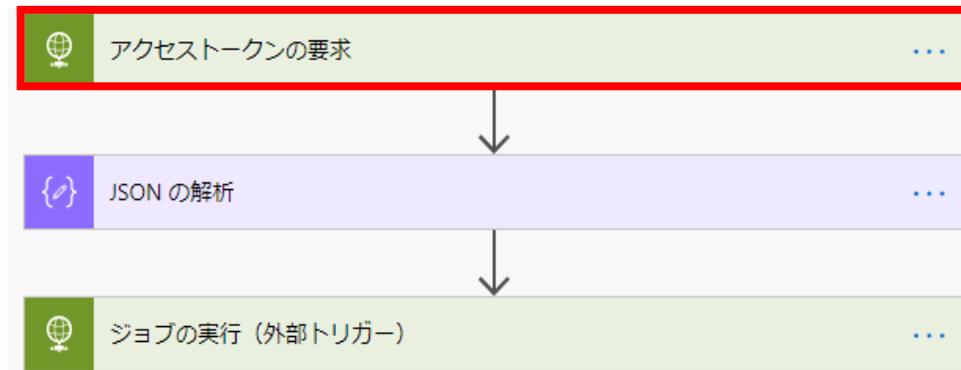
Cloud OrchestratorのAPIについて
<https://docs.uipath.com/orchestrator/lang-ja/v0/reference/consuming-cloud-api>

3. 環境設定 (参考)

3-7. Power Automateにおけるワークフロー例①

Power Automateにおけるワークフロー例 (StartJob APIのみ) 下記のとおり

※Cloud Orchestrator利用想定



アクセストークンの要求

*方法: POST

*URI: https://account.uipath.com/oauth/token

ヘッダー

Content-Type	application/json	×
X-UIPATH-TenantName	テナント名	×
Enter キー	値の入力	

クエリ

Enter キー	値の入力	
----------	------	--

本文

```
{
  "grant_type": "refresh_token",
  "client_id": "クライアントID",
  "refresh_token": "ユーザーキー"
}
```

Cookie: HTTP クッキーを入力してください

[詳細オプションを表示する](#)

3. 環境設定 (参考)

3-7. Power Automateにおけるワークフロー例②

Power Automateにおけるワークフロー例 (StartJob APIのみ) 下記のとおり

※Cloud Orchestrator利用想定

The image shows a workflow diagram on the left and a detailed view of the 'JSONの解析' (JSON Parsing) step on the right. The workflow consists of three steps: 'アクセストークンの要求' (Request Access Token), 'JSONの解析' (JSON Parsing), and 'ジョブの実行 (外部トリガー)' (Run Job (External Trigger)). The 'JSONの解析' step is highlighted with a red border. The detailed view of this step shows the 'JSONの解析' configuration. Under 'コンテンツ' (Content), '本文' (Body) is selected. Under 'スキーマ' (Schema), a JSON schema is defined with the following structure:

```
{
  "type": "object",
  "properties": {
    "access_token": {
      "type": "string"
    },
    "id_token": {
      "type": "string"
    },
    "scope": {
```

A 'サンプル JSON バイロードの挿入' (Insert Sample JSON Payload) dialog box is open, showing a sample JSON payload:

```
{
  "access_token": "test",
  "id_token": "test",
  "scope": "test",
  "expires_in": 86400,
  "token_type": "Bearer"
}
```

The '完了' (Done) button is visible at the bottom right of the dialog. A red box highlights the 'サンプルから生成' (Generate from Sample) button, and a blue callout box contains the instruction: 「サンプルから生成」を押下し、左記のスキーマをサンプルとして入力 (Press 'Generate from Sample' and enter the schema as a sample).

3. 環境設定 (参考)

3-7. Power Automateにおけるワークフロー例③

Power Automateにおけるワークフロー例 (StartJob APIのみ) 下記のとおり

※Cloud Orchestrator利用想定



ジョブの実行 (外部トリガー)

*方法: POST

*URI: `https://cloud.uipath.com/アカウント論理名/テナント名/odata/Jobs/UiPath.Server.Configuration.OData.StartJobs`

ヘッダー

Content-Type	application/json	×
X-UIPATH-TenantName	テナント名	×
Authorization	Bearer access_token ×	×
X-UIPATH-OrganizationUnitId	フォルダーID	×
Enter キー	値の入力	

クエリ

Enter キー	値の入力	適
----------	------	---

本文

```
{
  "startInfo": {
    "ReleaseKey": "JobのKey (APIで要求)",
    "Strategy": "ModernJobsCount",
    "RobotIds": [],
    "JobsCount": 1,
    "JobPriority": "Normal",
    "RuntimeType": "Unattended",
    "InputArguments": "{}"
  }
}
```

Cookie: HTTP クッキーを入力してください

詳細オプションを表示する

Thank you



Legal Copy: Es quate in preperu mquaspitatia nobitui menduntiam non nem facimen estrum faceaturenia voluptist, sum, voluptatiis et quodit aces del imi, quidell aborepello beat quatatus, sam im el et officiis estrum facepedio quae nonserum, ex et que doluptatius, que cuptureped quidi tentoresto moluptati odionecea nia quis ilit ma videbitesci ra poruntin cum con nost, sedis ex explab imporem. Hillore peria velis quae et mod quia sed ma volo temossitio que as sinci dem qui occulpa si ommolore susanda nihitis ellamet officiisit volupta temporumqui bero et quis undis eos elique pario. Et moluptae solore velicil itassi omnihiciis entempor re, voluptiunt occumquam, a verae pre debitat exerum voloriandi alit porehenihit, omnim eum cum venihit enihill enihillit, aut qui am dollam re duntio tene sunt.