



## 詳細解説！Hinemos ver.7.0の全体像

2021.11.10  
NTTデータ先端技術株式会社  
澤井 健

Hinemos ver.7.0

# Hinemos ver.7.0が実現する運用アナリティクス



運用アナリティクス

### 収集・蓄積

<b>パフォーマンス</b> <ul style="list-style-type: none"><li>リソース値</li><li>応答時間</li><li>ログ件数</li></ul>	<b>システム構成</b> <ul style="list-style-type: none"><li>プロセス</li><li>パッケージ</li><li>OS</li></ul>
<b>プラットフォームログ</b> <ul style="list-style-type: none"><li>システムログ</li><li>NWパケット</li><li>センサーデータ</li></ul>	<b>アプリ情報</b> <ul style="list-style-type: none"><li>アプリログ</li><li>プロセス</li><li>リソース</li></ul>

### 見える化・分析

<b>障害検知</b>	<b>将来予測</b>
<b>構成情報の見える化</b> <ul style="list-style-type: none"><li>プロセス</li><li>RPM</li><li>Linux</li></ul>	<b>インテリジェントなフィルタリング</b> <ul style="list-style-type: none"><li>プロセス</li><li>msi</li><li>Windows</li></ul>

### 自動化

<b>ジョブ管理・ワークフロー制御</b>	<b>リソース最適化</b>
<b>パッケージの最新化</b> <ul style="list-style-type: none"><li>RPM</li><li>msi</li><li>OS</li></ul>	<b>RPA 管理</b>

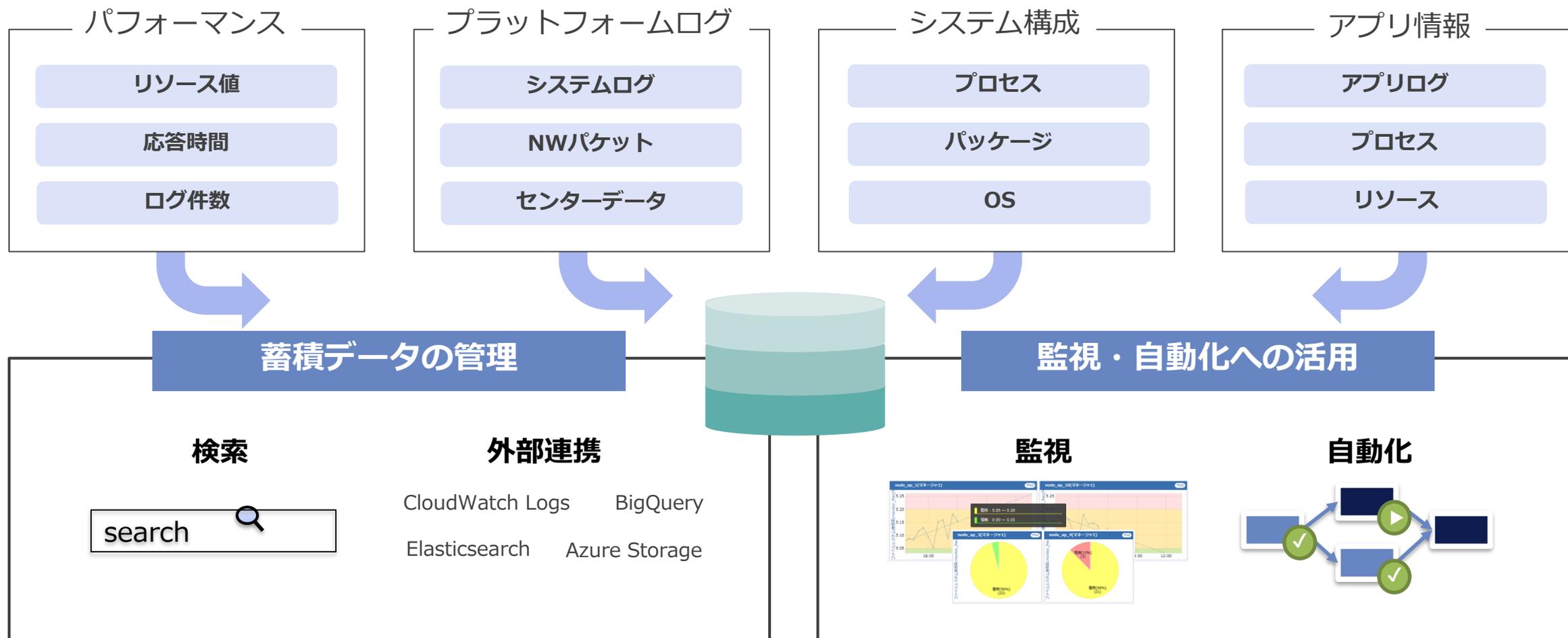
## DX

デジタルトランスフォーメーション

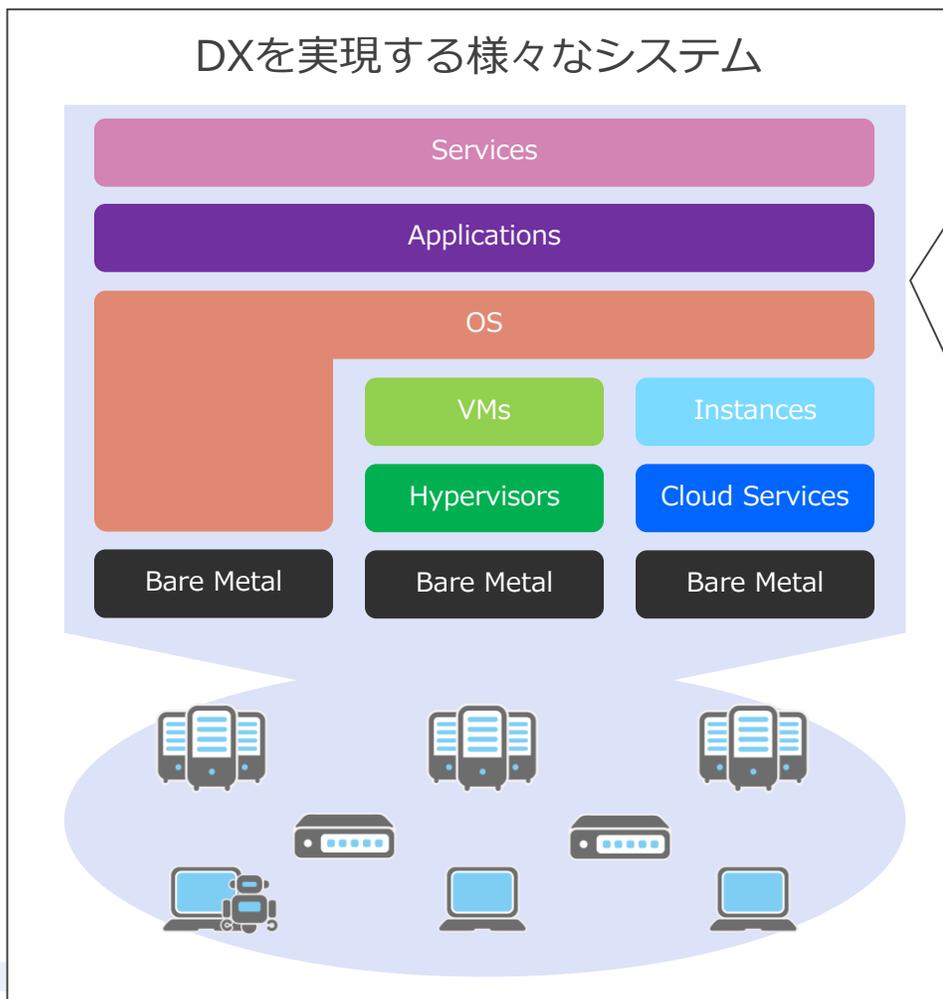
DX を推進する IT システムの方法論の監視・自動化ツールを統合



ITのありとあらゆるデータを収集・蓄積し、蓄積データの管理と監視・自動化への活用



## リアルタイムなシステム運用状況の監視・視覚化と、将来予測や変化量による異常を検知



### 監視

A screenshot of a monitoring dashboard displaying a map of Japan. A red traffic light icon is shown next to the map, indicating a critical status. The dashboard includes various data points and status indicators.

### 将来予測

Two line graphs illustrating future predictions and anomalies:

- The top graph shows CPU usage (CPU) over time. A blue circle highlights a period labeled "今は正常だが" (Currently normal, but...). A red circle highlights a point labeled "1ヶ月後に異常" (Abnormality in 1 month). A dashed line indicates a predicted upward trend.
- The bottom graph shows memory usage (MEM) over time. A red circle highlights a point labeled "予測値超えが多発し急降下" (Prediction exceeded frequently, sharp drop). A blue box highlights a point labeled "次の値を予測し 予想以上の変化を検知" (Predict the next value and detect changes beyond expectations).

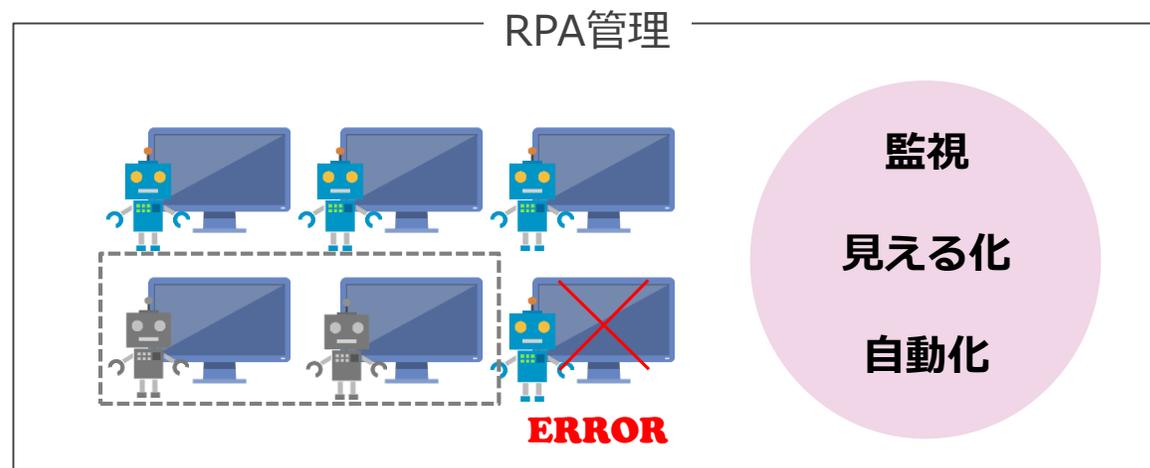
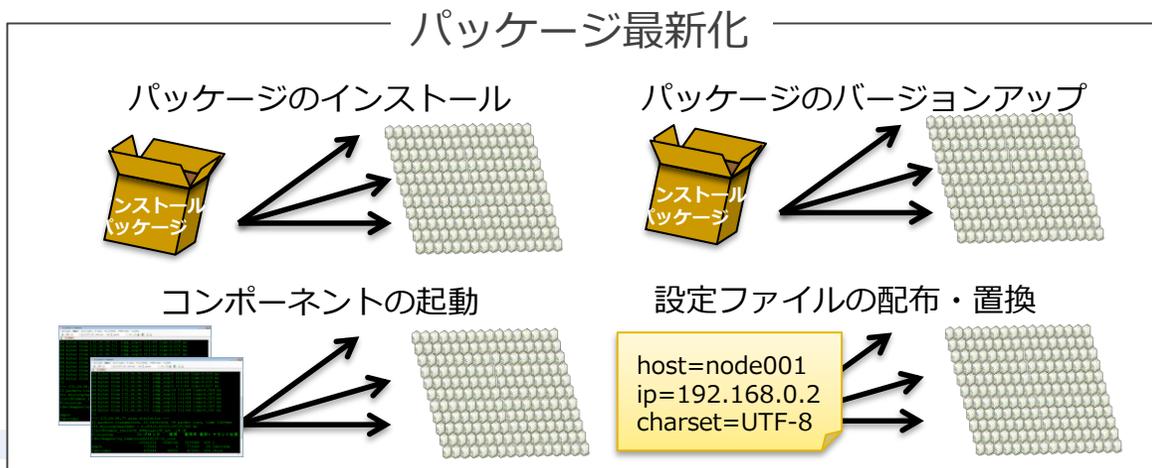
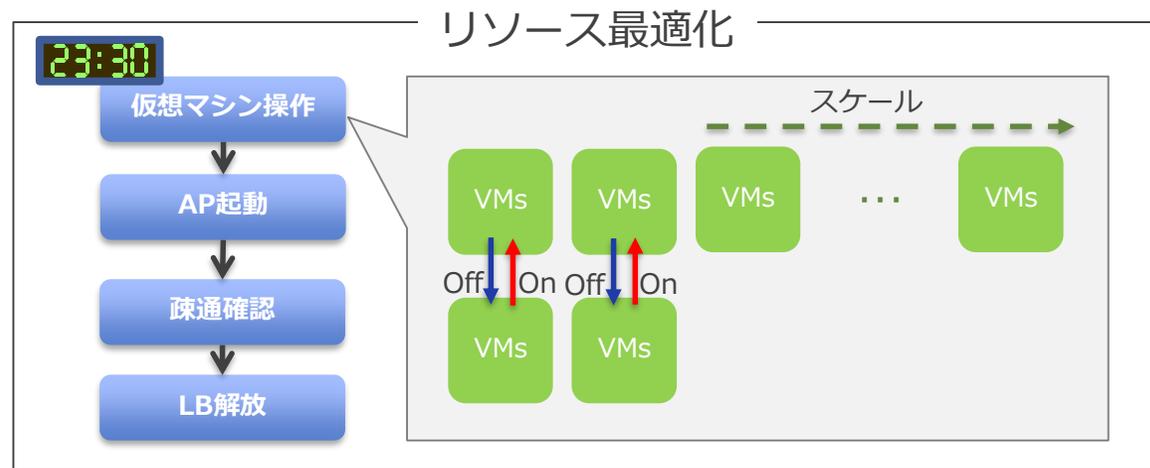
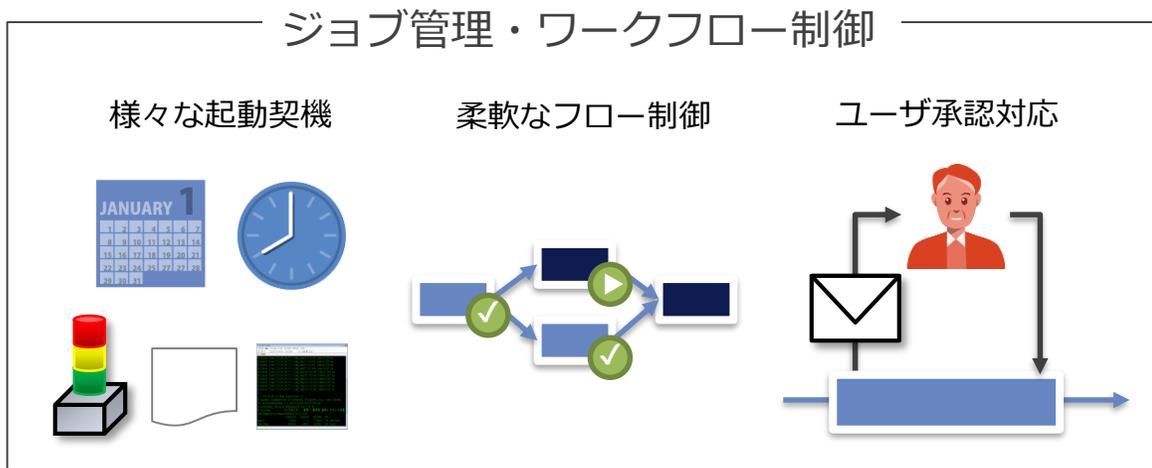
### 構成情報の見える化

A diagram showing the visualization of configuration information. It features a central cylinder labeled "CMDB" (Configuration Management Database). Below it, a sequence of packages is shown: "PKG Ver.1", "PKG Ver.1", "PKG Ver.2", and "PKG Ver.2". A yellow starburst labeled "変化" (Change) is positioned between the two "PKG Ver.1" items, indicating a change in configuration.

### インテリジェントなアラート

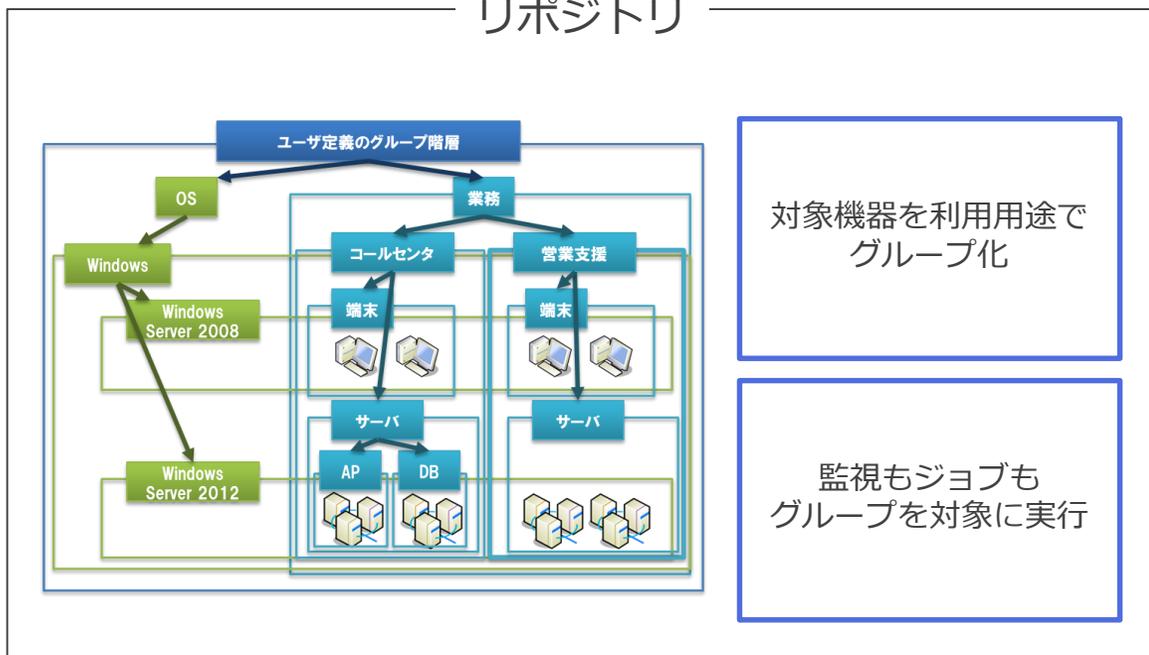
A diagram illustrating intelligent alert filtering. It shows a series of "msg" (message) boxes in various colors (red, yellow, green, blue). A magnifying glass icon labeled "フィルタールール" (Filter Rule) is shown filtering the messages, resulting in only two "msg" boxes (blue and grey) being displayed.

## ジョブ・ワークフロー、サーバリソース、パッケージ構成、RPAの様々な自動化を集約管理



## 機能間で共通利用する設定の集約と、シームレスな機能間連携を実現

### リポジトリ



### カレンダー



### アカウント



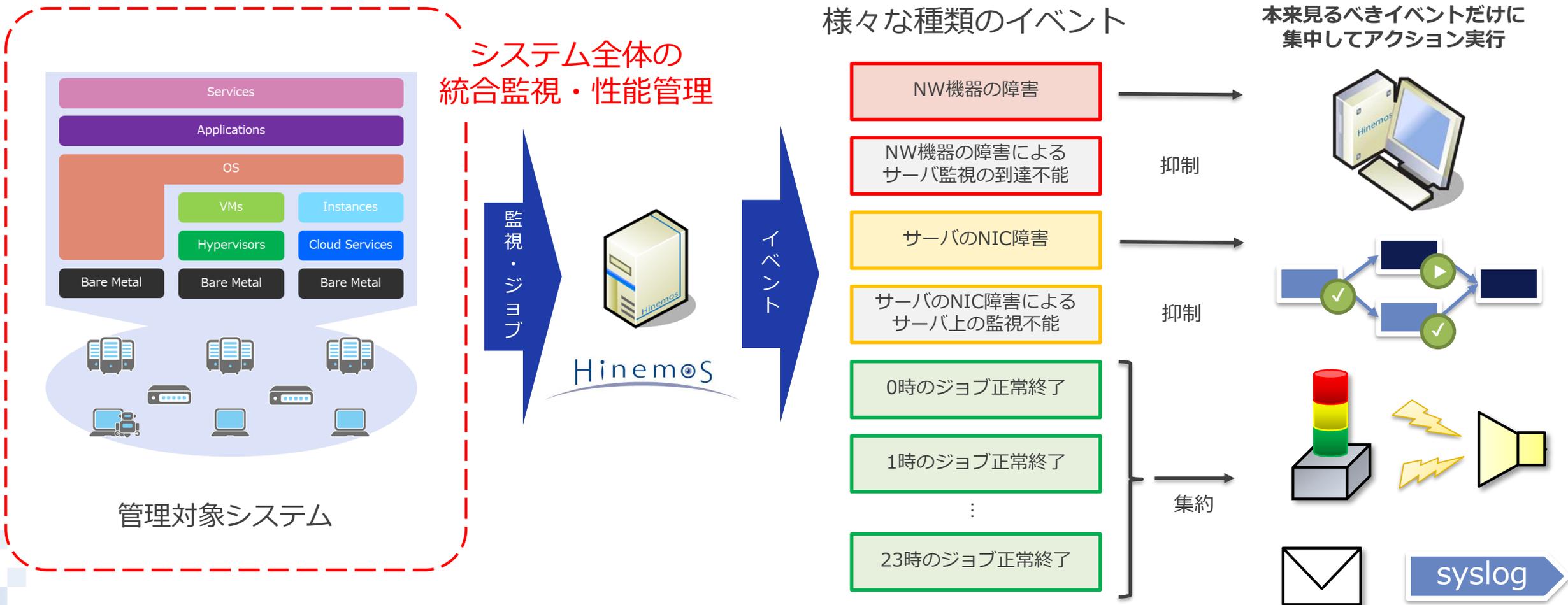
### 通知



# 新機能 メッセージフィルタ

# 背景 本質的なイベント

運用現場では様々な事象検知のために大量の「メッセージ」が発生し、「本質的なイベント」を発見する事が困難になっています



# メッセージフィルタとは

ルールエンジンを活用し、インテリジェントなアラートと自動化を実現



## メッセージフィルタの4つの特徴

インテリジェントなアラート

インテリジェントな自動化

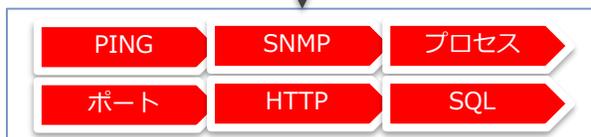
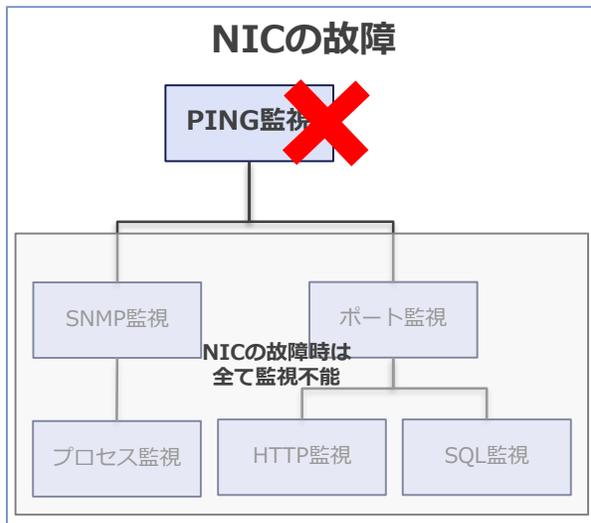
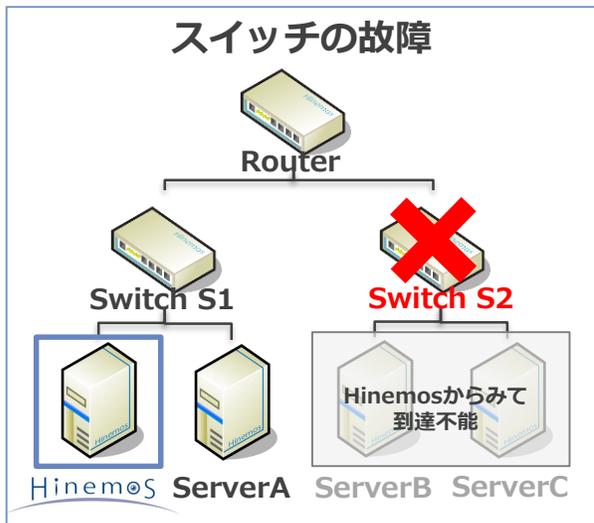
ルールベースの条件指定

Hinemosからのシームレスな導入

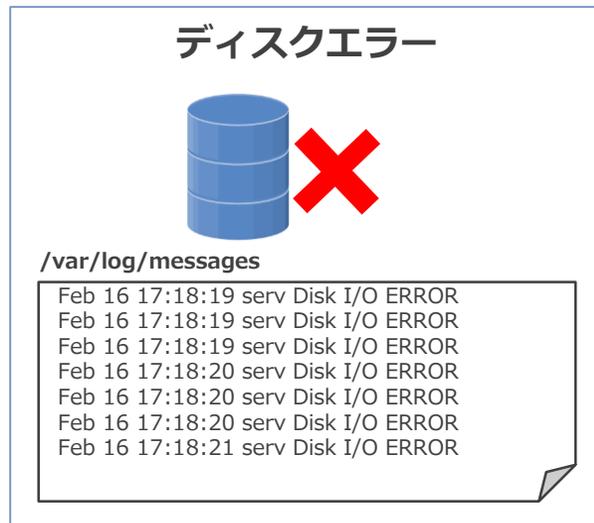
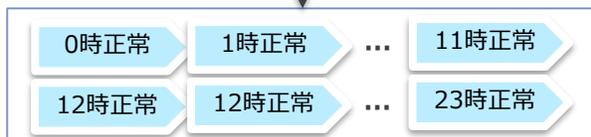
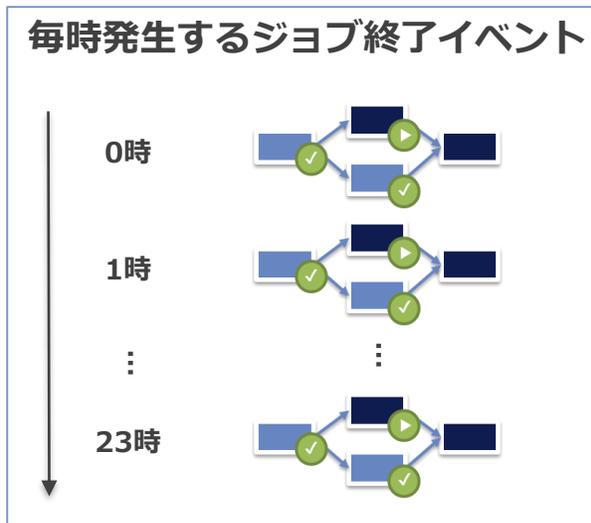
# インテリジェントなアラート

不要なメッセージの抑制と関連メッセージの集約により本質的なイベントの対処に注力できます

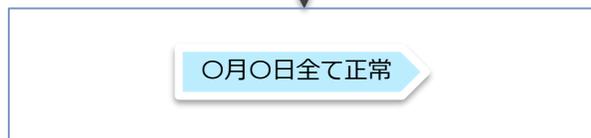
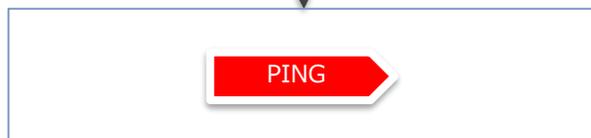
## 不要なメッセージの抑制



## 関連メッセージの集約

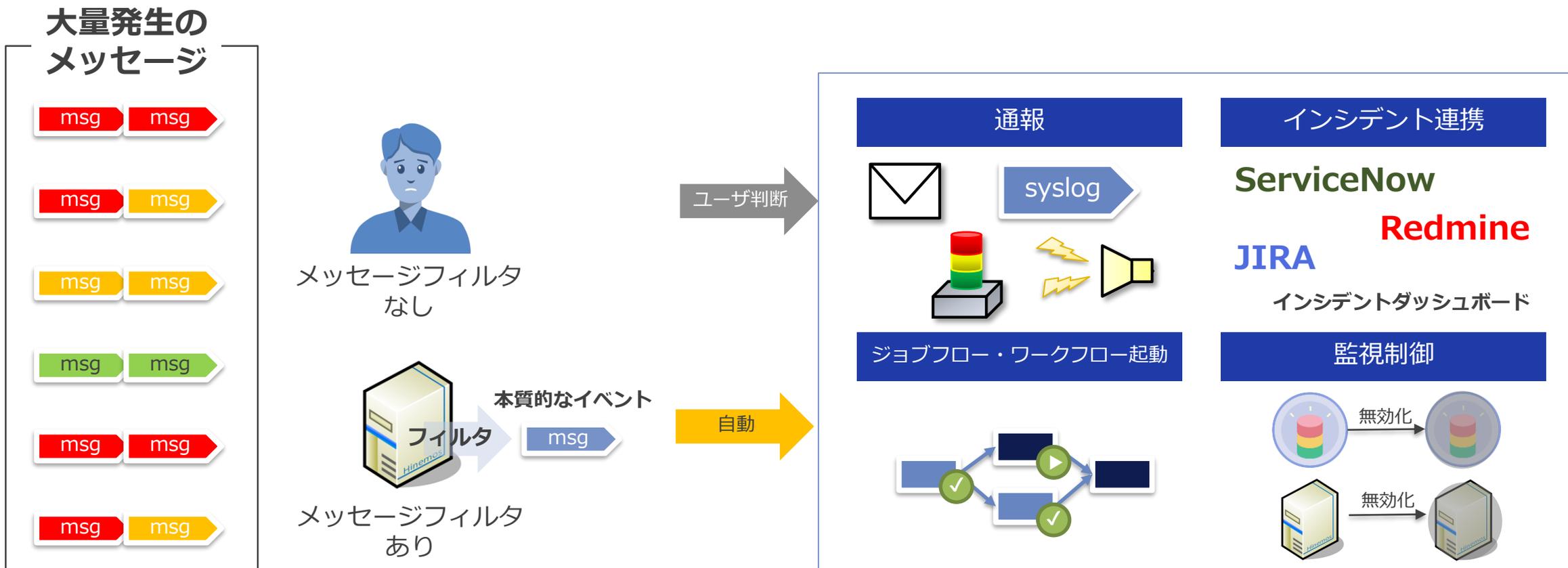


## 本質的なイベントへのフィルタリング (抑制・集約)



# インテリジェントな自動化

本質的なイベントメッセージから直ちに通報、インシデント連携、ジョブフロー・ワークフロー起動、監視制御といった運用業務に連動します

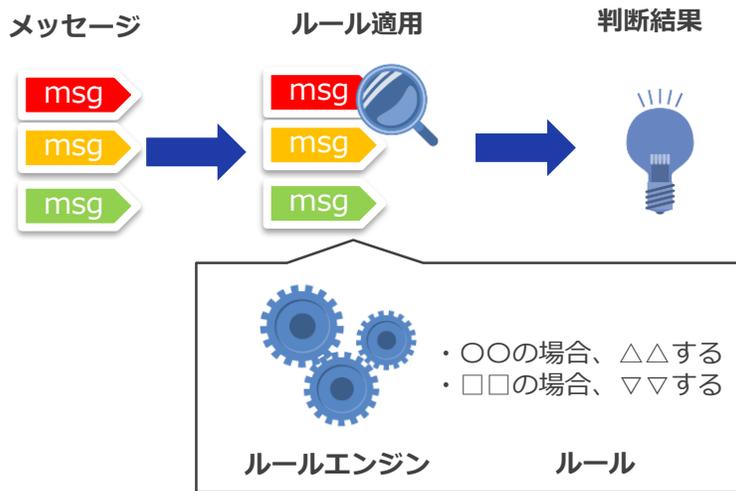


# ルールベースの条件指定

When/Thenで定義するシンプルなルールを指定するだけ。複合イベント処理（CEP）により、イベント間の関係性をルールに指定できます

## ルールエンジンを採用

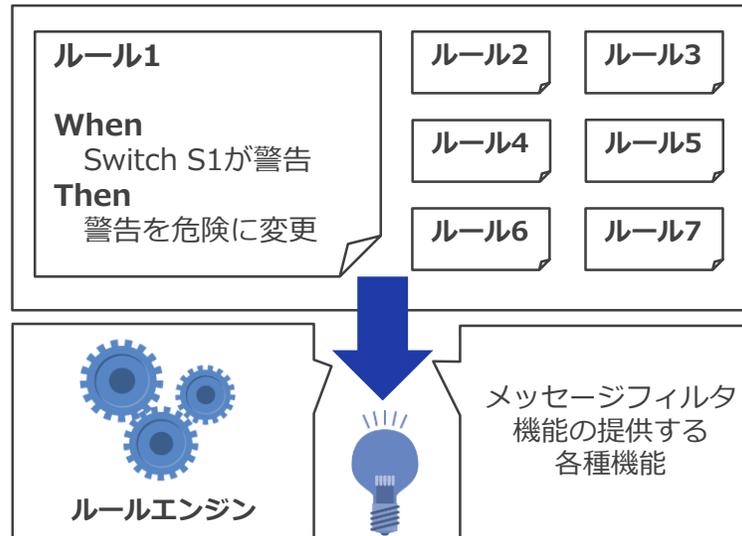
ルールエンジンとは、与えられたルールに従って判断を行う機能です



メッセージフィルタ機能では、Droolsを採用しています

## When/Thenのルールを記述

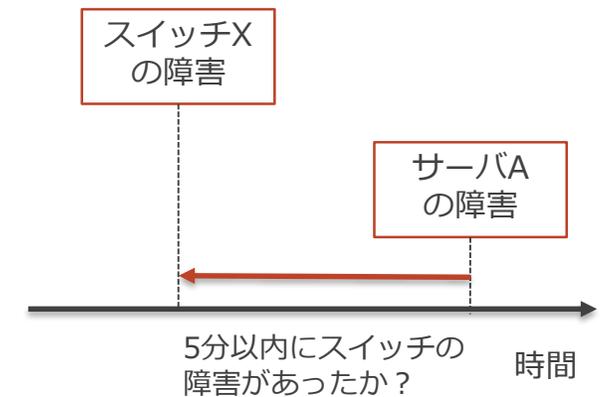
When/Thenからなるルールを記載すると、メッセージフィルタが適切に判断します



ルールの文法は、DRL (DROOLS RULE LANGUAGE) ルール言語に従います

## イベント間の関係性も条件化

複合イベント処理（CEP）によりイベント間の関係性をWhenの条件に記載できます



Droolsが提供するCEP機能をそのまま使用できます

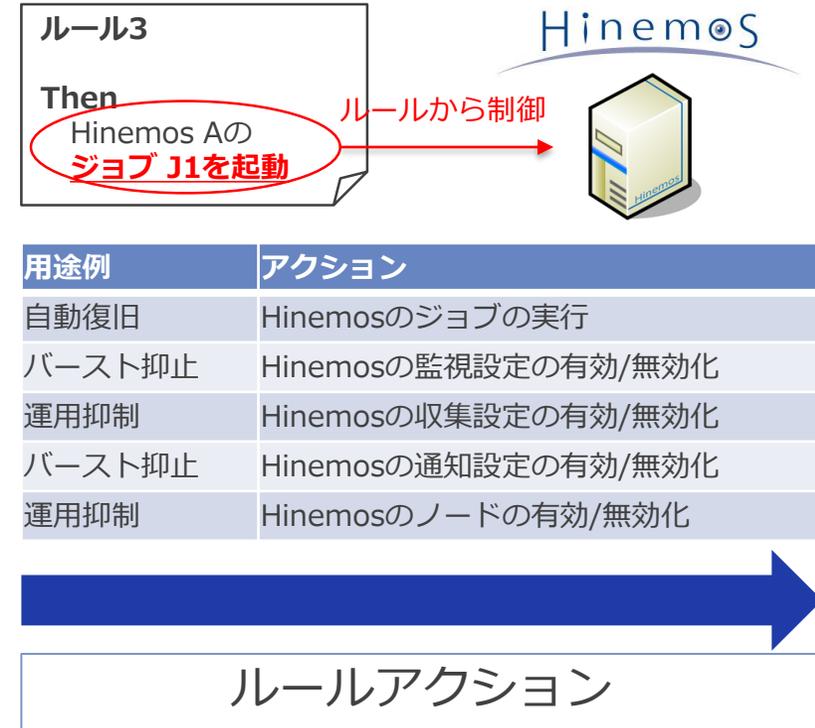
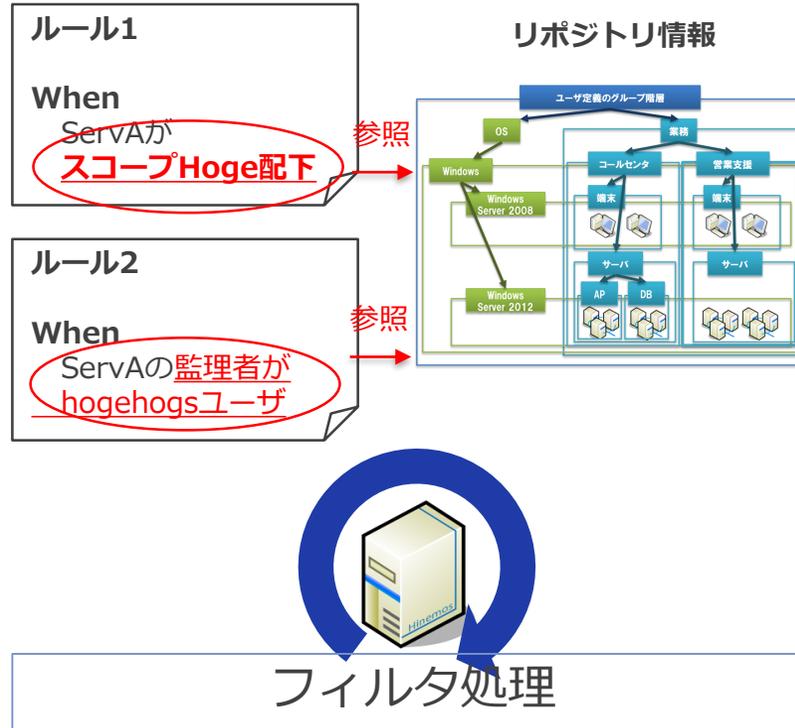
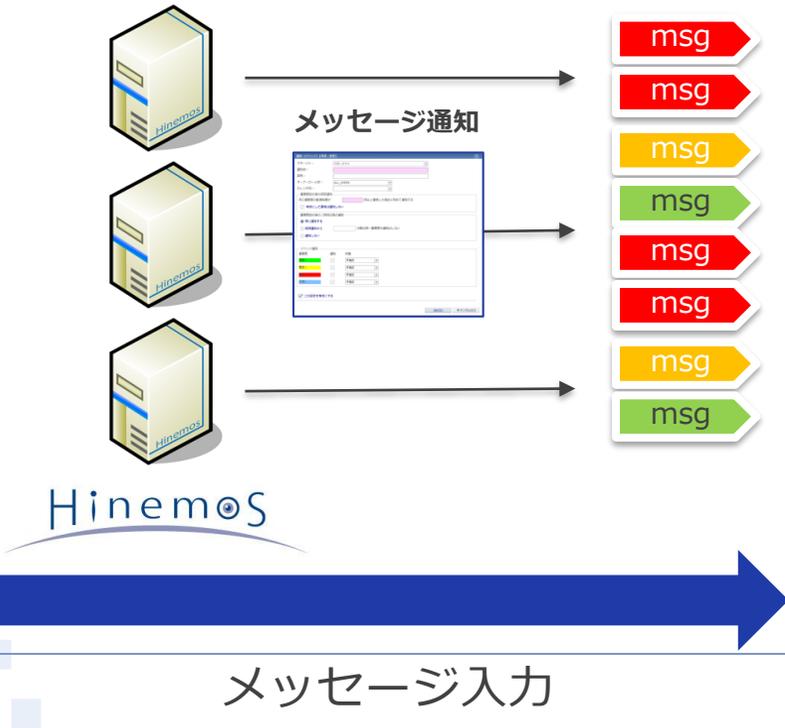
# Hinemosからのシームレスな導入

Hinemosメッセージを受信し、リポジトリ情報をルールの中で参照可、そしてルール判定後のアクションでもHinemosの直接操作も可能です

Hinemosの監視やジョブ実行結果をメッセージ通知を使ってメッセージフィルタに簡単に連携できます

Hinemosの保持するリポジトリ情報をルールの条件（When）で利用できます

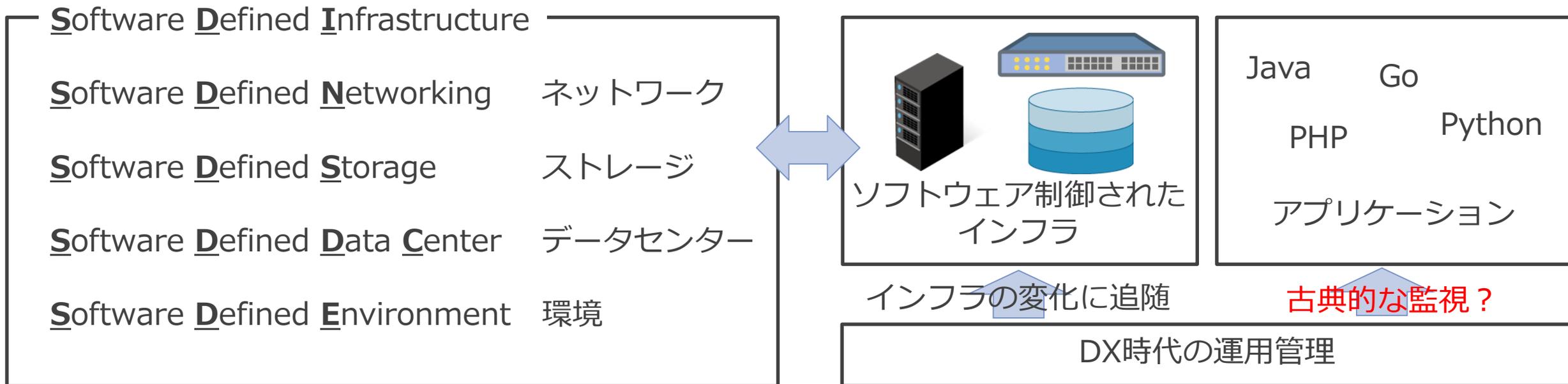
ルールの判定の結果のアクション（ルールアクション）でHinemosの各種操作が可能です



用途例	アクション
自動復旧	Hinemosのジョブの実行
バースト抑止	Hinemosの監視設定の有効/無効化
運用抑制	Hinemosの収集設定の有効/無効化
バースト抑止	Hinemosの通知設定の有効/無効化
運用抑制	Hinemosのノードの有効/無効化

# 新機能 SDML

ソフトウェアで定義・制御する世界は基盤（インフラストラクチャー）が中心



ユーザのニーズに合わせた柔軟なインフラの変更へ運用管理は追随  
次は、アプリケーションの運用の利便性向上へ

サービスレベルの維持に重要なアプリケーションの監視は古典的に運用

## アプリ開発後に運用設計

アプリケーション開発中は開発のみに注力して、運用については別工程や別チームが考える事として分割されている

## アプリ開発者以外が運用設計を

既に出来上がってしまったアプリに対して運用を考えるため、本質的な監視を導入しようとしても改修を伴い実現できないケースが多い

## 短いシステム開発のサイクル

クラウド・コンテナ・アジャイル開発といった要素により開発からリリースまでのサイクルが非常に短くイテレーティブ（反復的）な開発の流れに

## アプリ監視設計の期間がない

アプリ開発や試験から運用開始までに監視設計の期間が短くなり、基本的な監視を導入しただけで本質的な監視が行えていないケースが多い

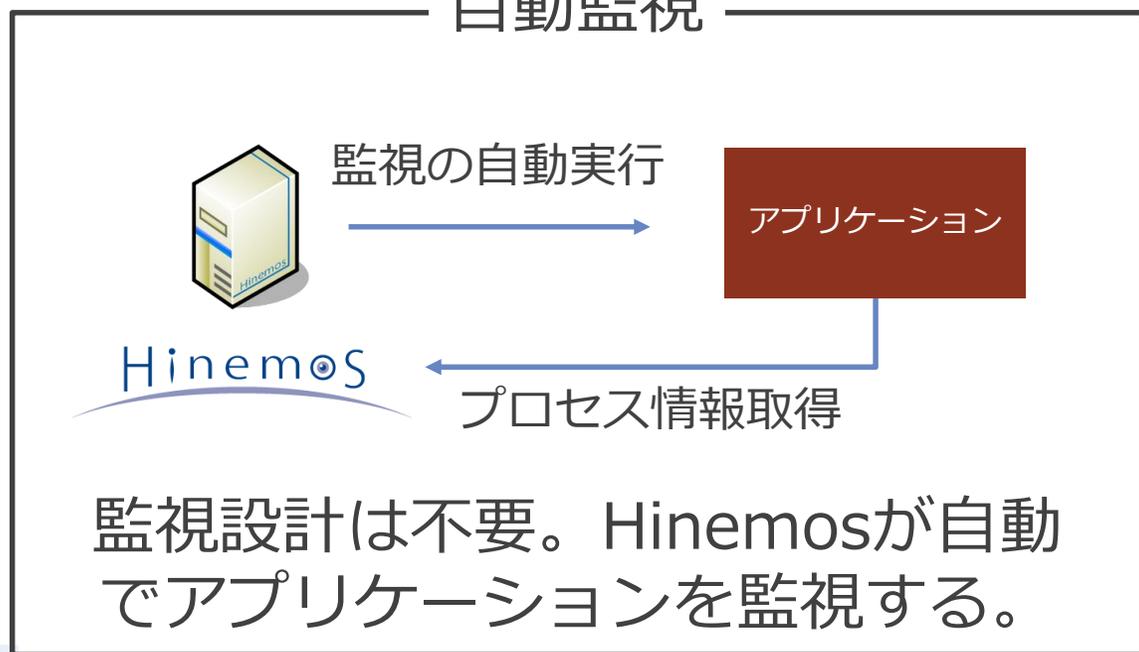
アプリケーションの開発やリリースのライフサイクルに合わせてながらより本質的な監視を実現できる仕組みが重要になる

Hinemosが提案する新たな運用

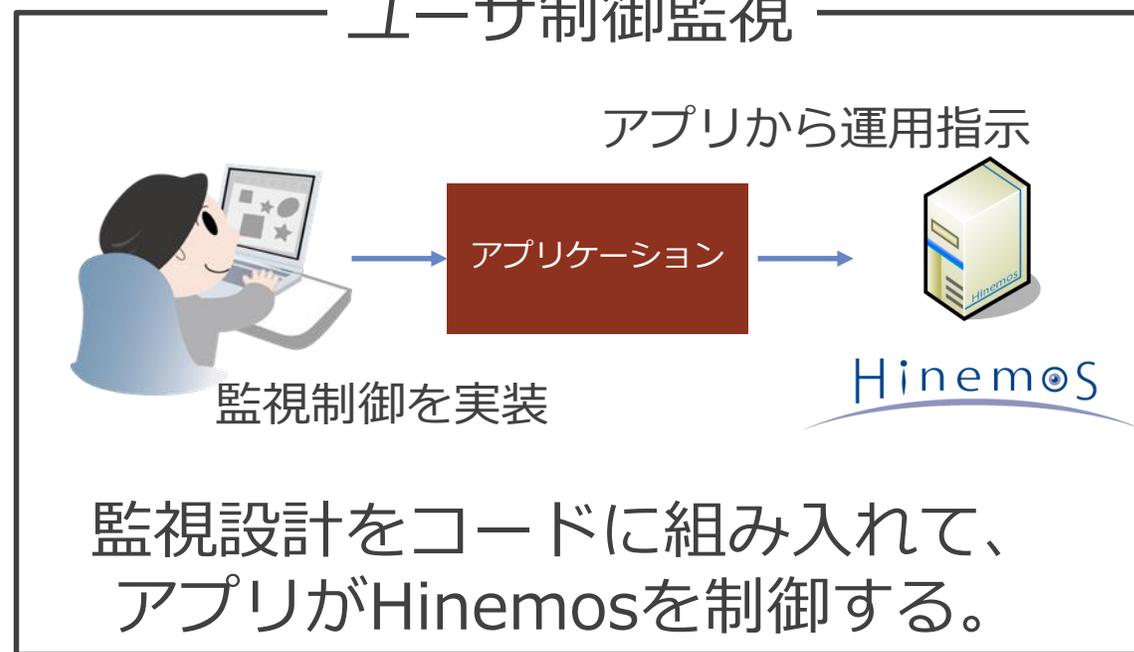
# Software Defined Monitoring and Logging

ソフトウェアでモニタリング（監視）とロギングを定義・制御

## 自動監視

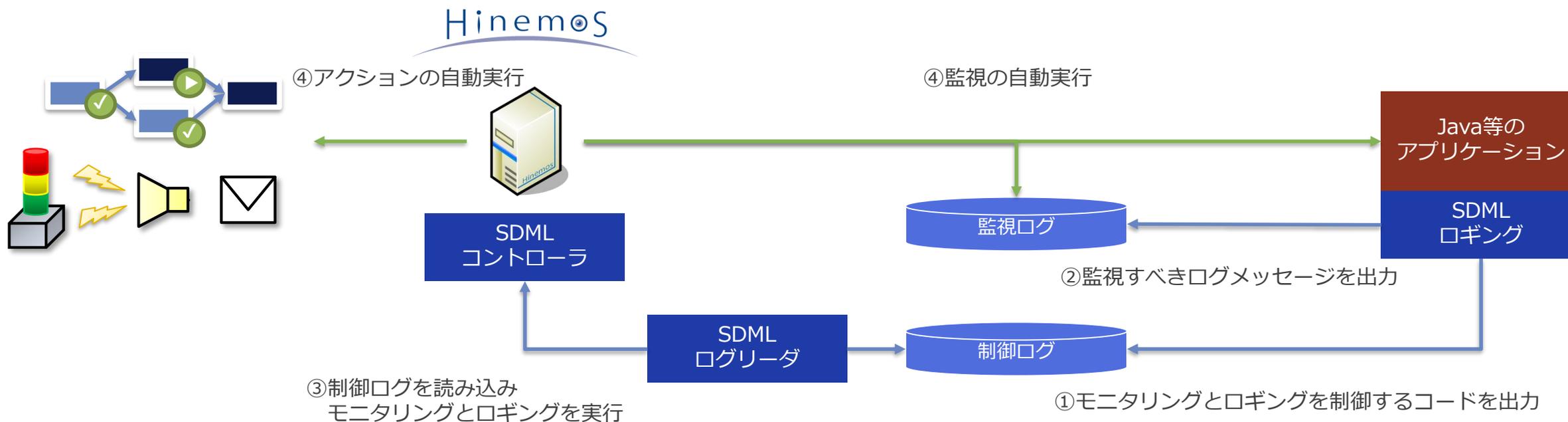


## ユーザ制御監視



特許出願中(特願2021-157931)

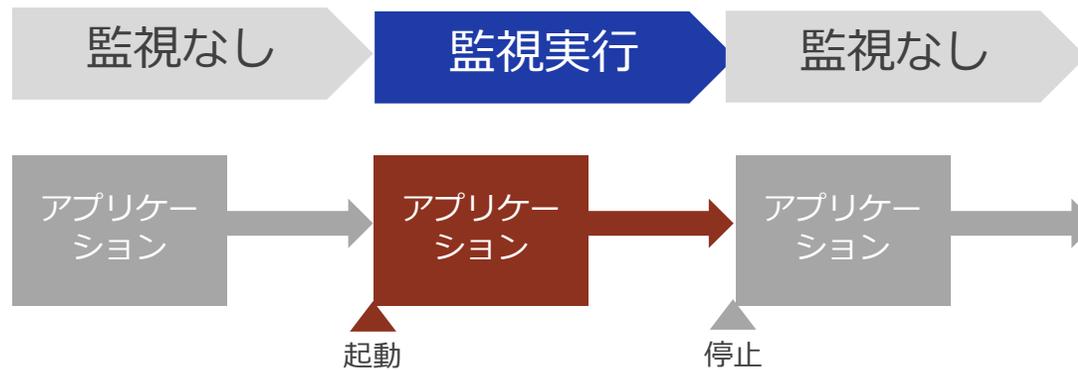
SDMLロギングをアプリケーションに組み込み、制御ログと監視ログを介在し  
モニタリング（監視）とロギングの定義と制御を実現



**自動監視**では、何もしなくてもアプリの監視が可能になる  
**ユーザ制御監視**では、アプリ開発に「監視制御」を含める事が可能になる

設計は不要！アプリケーションの起動・停止に合わせて自動的に監視を実行

## 自動的な監視実行

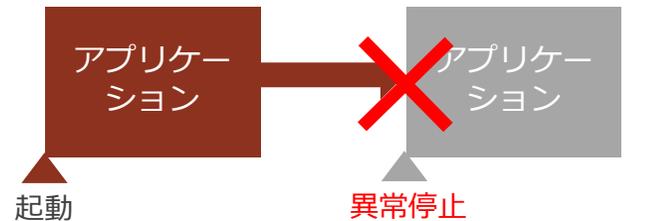


アプリケーション起動時に監視を自動で開始し、アプリケーション正常終了時に監視を停止します。

どんな監視が必要かの設計や、監視の設定が不要で自動でアプリケーションのプロセスの死活状態や内部状態を監視し、異常時にアラートを上げます。

## 様々な異常検知

### ① プロセスの死活状態



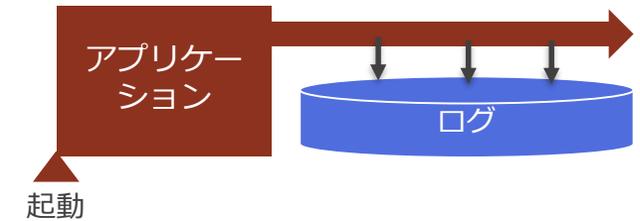
アプリケーション異常終了を検知してアラートを上げます

### ② プロセスの内部状態



プロセスの内部状態を定期的にウォッチし異常の予兆を検知します

### ③ プロセスのエラーログ



プロセスの出力するログを監視しエラーメッセージを検知します

## アプリケーション開発の中で監視制御を実装して最適な運用を実現

### 起動時に指定の監視を実行

```
class HogeAP {  
    public static void main(String[] args){  
        // 監視を起動  
        プロセス監視を起動；  
        ポート監視を起動；  
        ログファイル監視を実行；  
    }  
}
```

アプリケーションの実装の中で、アプリケーションに適した監視を指定します。

SDMLは制御コードを介して、この監視指示に従いこのアプリケーションの監視を行います。

### 異常時に指定の復旧ジョブを起動

```
try {  
    // 業務処理A  
} catch (HogeException e) {  
    スタックトレースを出力；  
    // 業務処理A専用の復旧ジョブXを起動  
    復旧ジョブXを起動；  
}
```

異常発生時にスタックトレース等のログを出力するだけでなく、復旧ジョブの起動まで制御します。

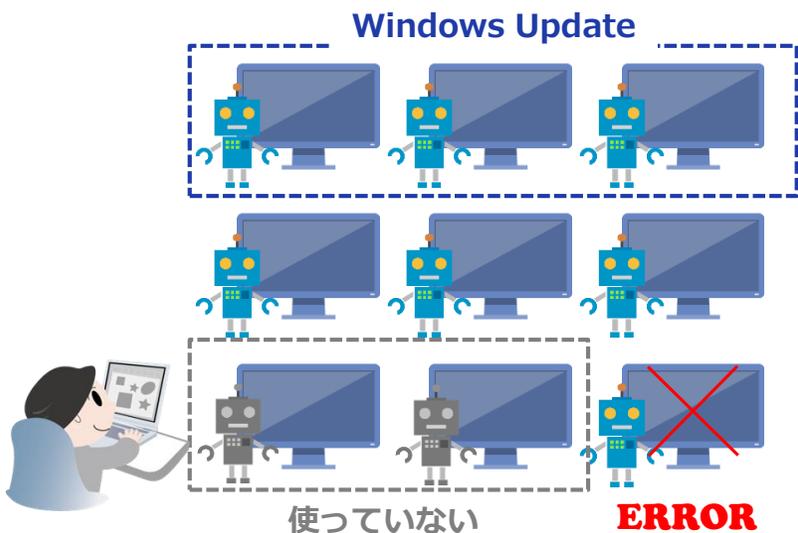
SDMLは制御コードを介して、このジョブ実行指示に従い、指定のジョブを起動します。

# 新機能 RPA管理

# WinActor導入の現状

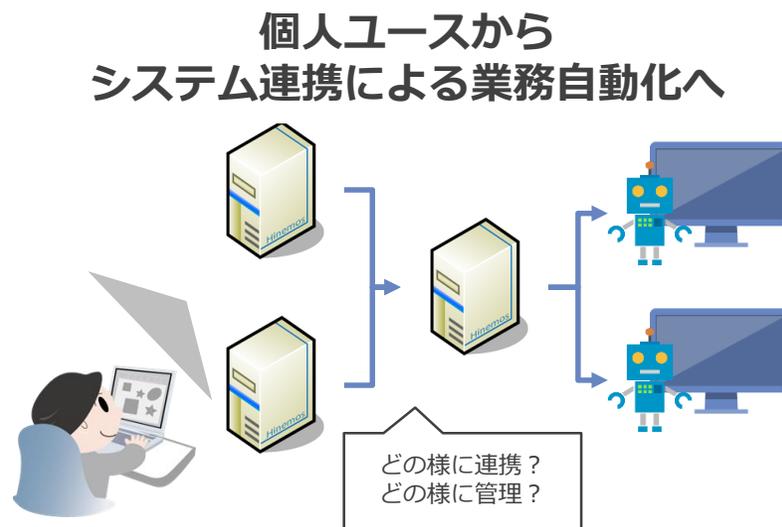
オフィス現場で爆発的に広まった事による次の悩みは**運用コスト**

## WinActorの導入端末数の増大による運用管理



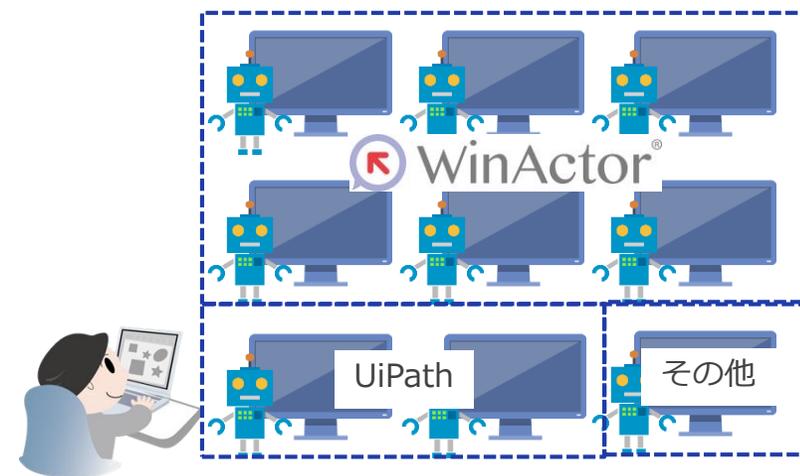
端末数が多くなるだけでシナリオの監視、PC端末の構成管理、適正なライセンス数把握が難しくなる

## WinActorの適用範囲の拡大による運用管理



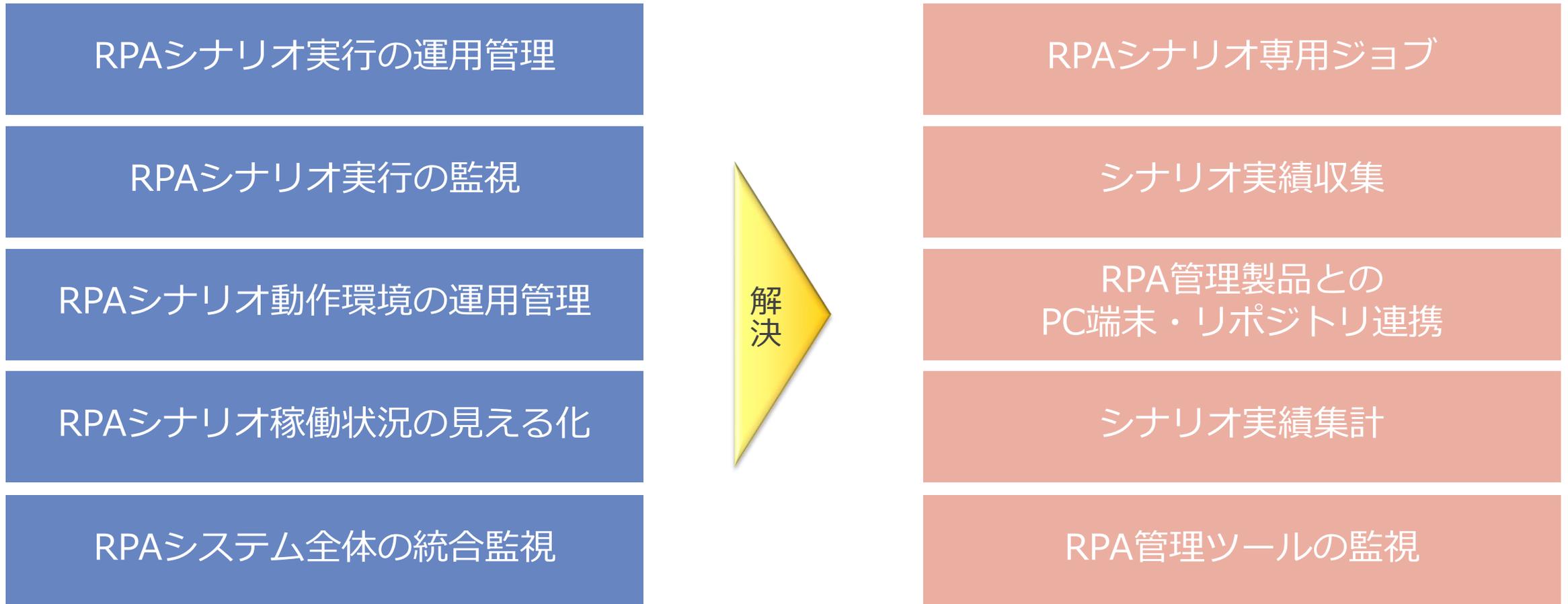
個々人の便利ツールから発展し業務全体を自動化・効率化するには、統括的な管理ツールが必要になる

## 他のRPAツール並行導入による運用管理



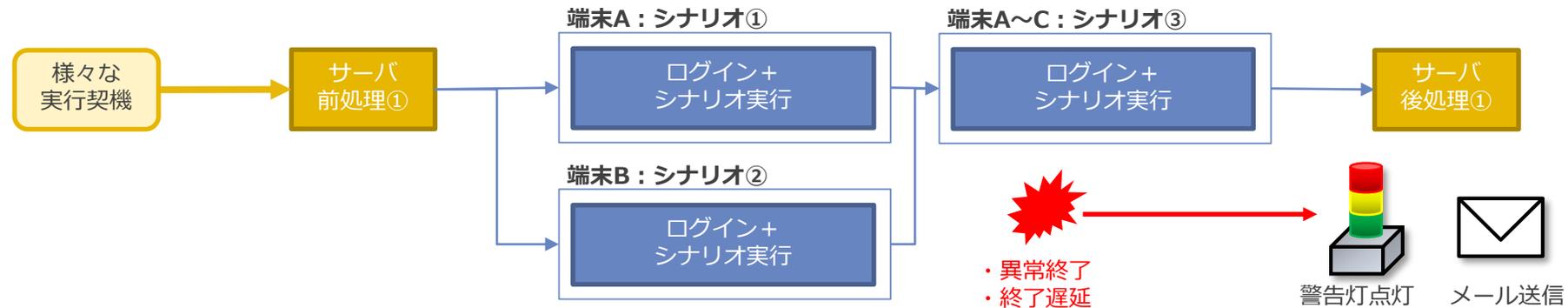
部署や用途の違いによりWinActor以外のRPAツールが混在すると環境を統括的に管理する事が難しくなる

WinActorの導入端末数の増大と適用範囲の拡大、他のRPAツール並行導入により肥大化した運用管理のコストをHinemosが大きく削減します



**各項目の具体的な課題とHinemosによる解決する機能をご紹介します**

## Hinemosによるジョブフロー管理にてRPAシナリオ実行の運用を効率的な管理を実現



### 様々な実行契機

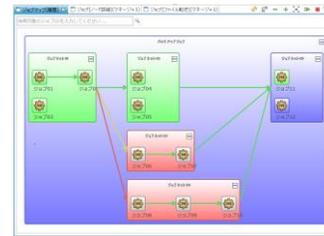
- 様々な起動契機を選択可能  
即時実行  
スケジュール実行  
監視連動  
ファイルチェック契機  
API/CLI契機
- 業務カレンダー連携が可能  
営業日・メンテナンス日切替

### 端末・サーバ間連動

- PC端末・サーバ間連動が可能  
PC端末へのログインも可能  
ITシステムとPC端末を跨ぎ  
業務フローを一括管理
- 複数シナリオの効率的実行  
複数PC端末での同時実行可能  
同一シナリオならスコープを  
使ってジョブ定義を集約可能

### 実行状況の確認

- 直感的な状況確認が可能  
フローをそのまま表示  
色による正常・異常判断

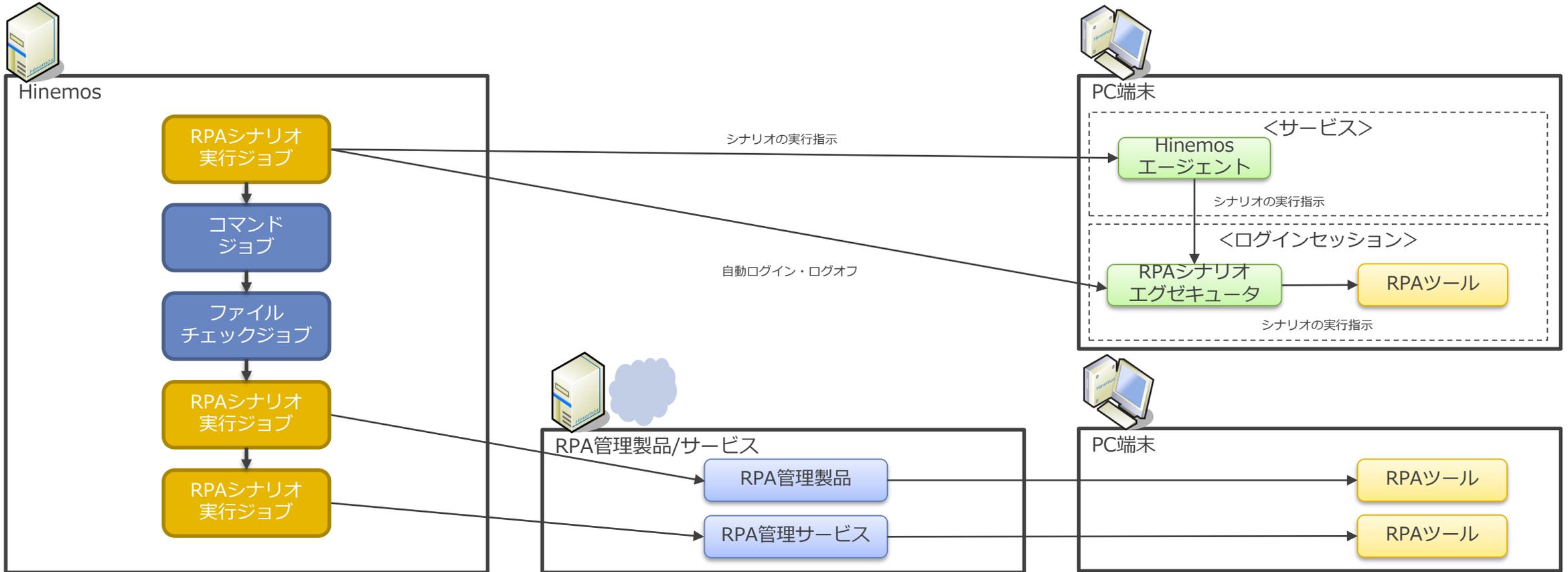


### 実行状況の監視

- 戻り値からの正常性管理  
シナリオの異常を検知
- 終わらないシナリオの監視  
遅延監視により長時間の  
シナリオを検出
- エラーメッセージの記録  
エラー内容はHinemosクライ  
アントから簡易に確認

## リモートからRPAシナリオを実行するRPA専用ジョブを追加

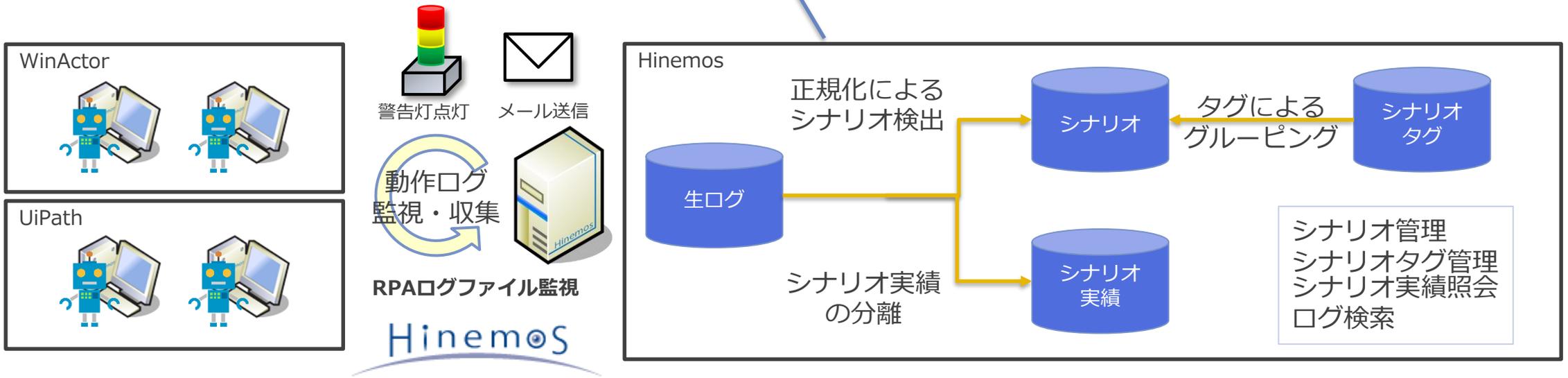
HinemoS



**複雑なセットアップは不要でGUIから実行するシナリオを指定するだけ**

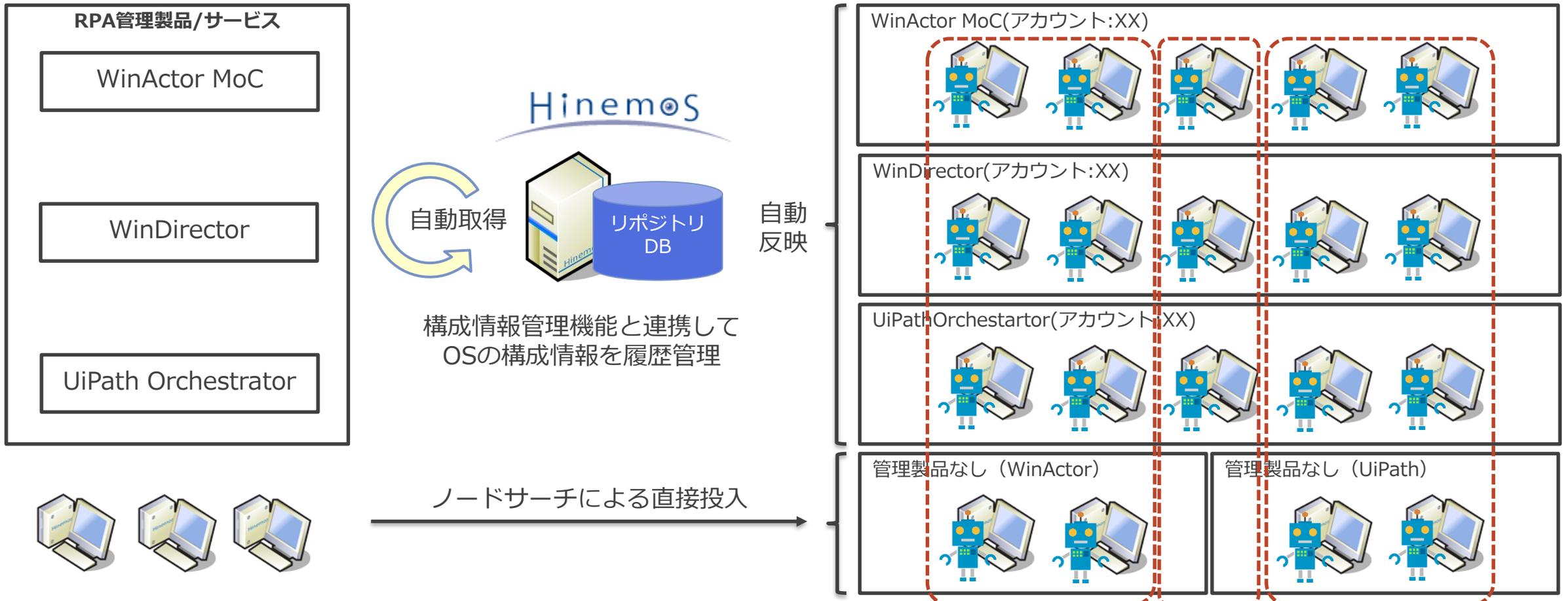
## 複数のRPA製品を跨って実行した監視とシナリオの検出・実績照会が可能に

マネージャ	RPA種別	シナリオID	シナリオ名	シナリオ識別子	説明	オーナー
マネージャ1	WinActor	CREATE-WinActor_0017	メール受注データ登録	RegistMailOrder.ums7		Role_V
マネージャ1	WinActor	CREATE-WinActor_0018	WEB-EDI受注データ振分け	SortWEBEDIOOrder.ums7		Role_V
マネージャ1	WinActor	CREATE-WinActor_0019	WEB-EDI受注データ登録	RegistWEBEDIOOrder.ums7		Role_V
マネージャ1	WinActor	CREATE-WinActor_0020	受注データメール送信	SendMailOrder.ums7		Role_V
マネージャ1	WinActor	CREATE-WinActor_0021	発注データアップロード			Role_V
マネージャ1	WinActor	CREATE-WinActor_0022	出荷データダウンロード			Role_V
マネージャ1	WinActor	CREATE-WinActor_0023	出荷指示メール送信			Role_V
マネージャ1	WinActor	CREATE-WinActor_0024	請求データダウンロード			Role_V



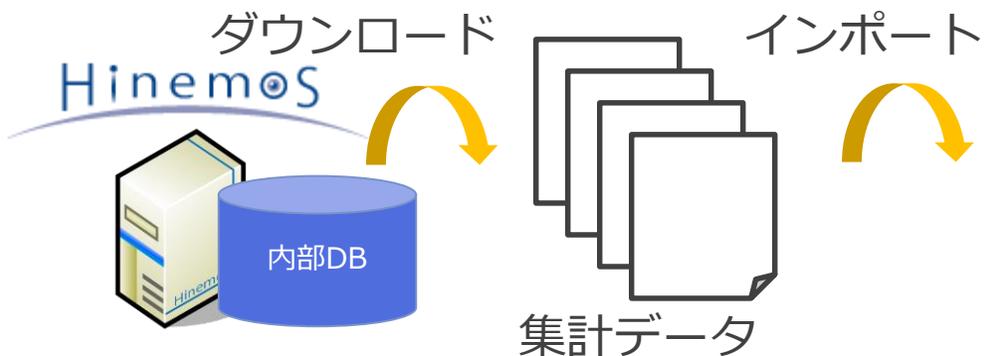
既に稼働している環境に導入して、現在の稼働状況を簡易に把握

## RPA管理製品からRPAの動作するPC端末情報を取得してリポジトリへ自動反映

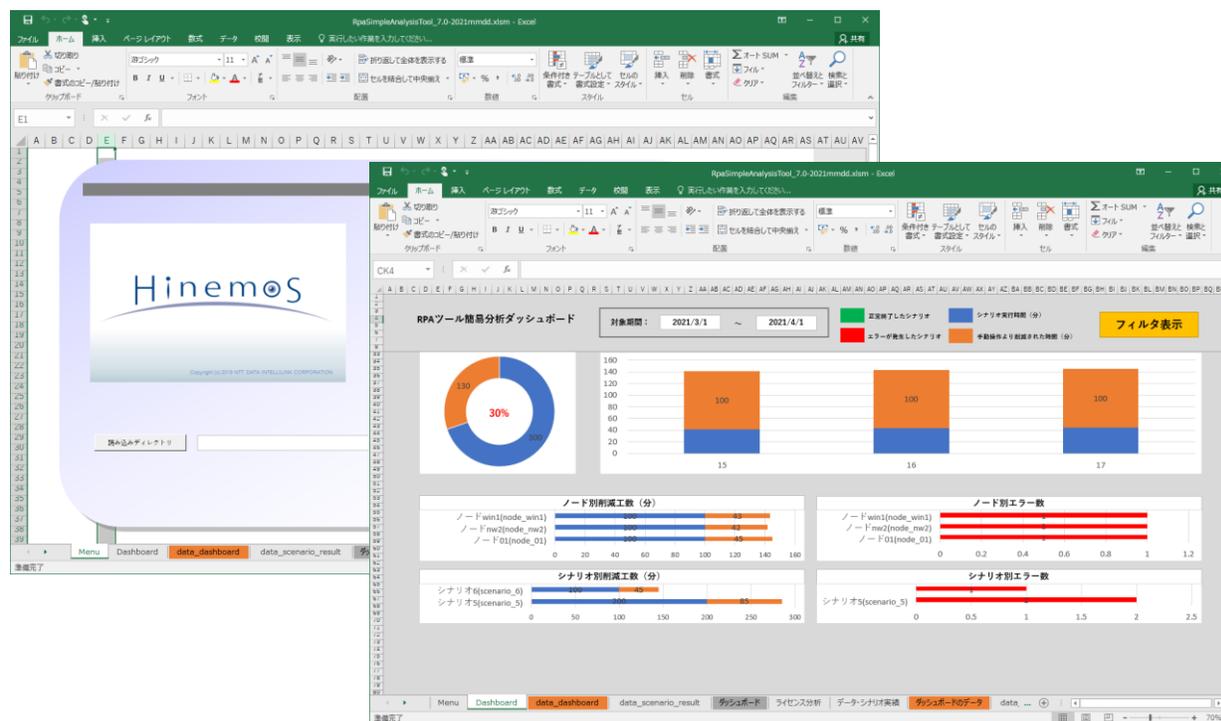


**管理対象のリポジトリ構成を簡単にセットアップ**

シナリオの稼働実績を様々な角度で集計しダウンロードが可能、また簡易分析ツール（Excel）により稼働状況のダッシュボード・同時実行数分析も可能



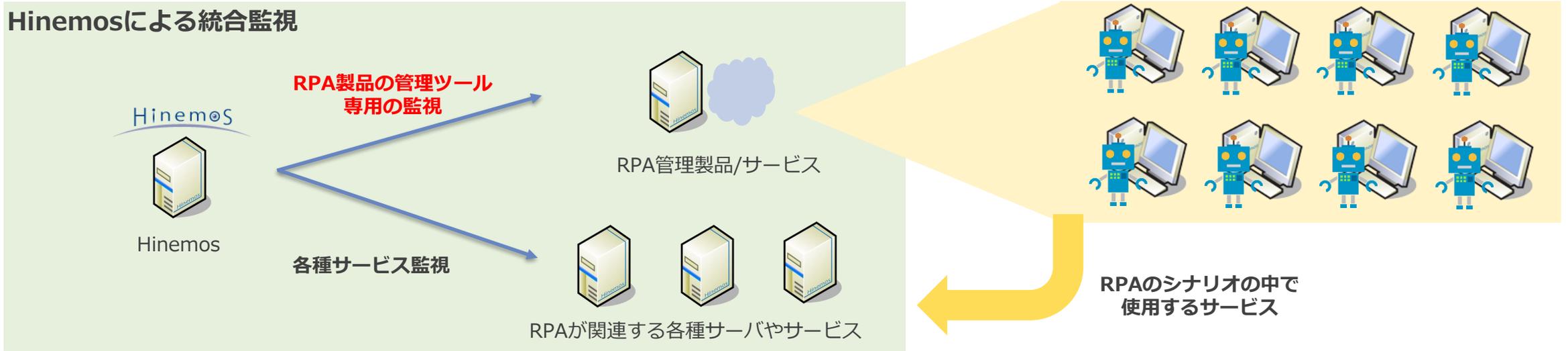
データの種類	データ名
メジャー	シナリオ実績データ
ディメンション	スコープ関連データ
	シナリオタグデータ
	シナリオタグ関連データ
名称	ファシリティ名データ
	シナリオデータ



シナリオ実績分析（Excel）

## RPA製品の管理ツールが正常に稼働していることを監視

### Hinemosによる統合監視



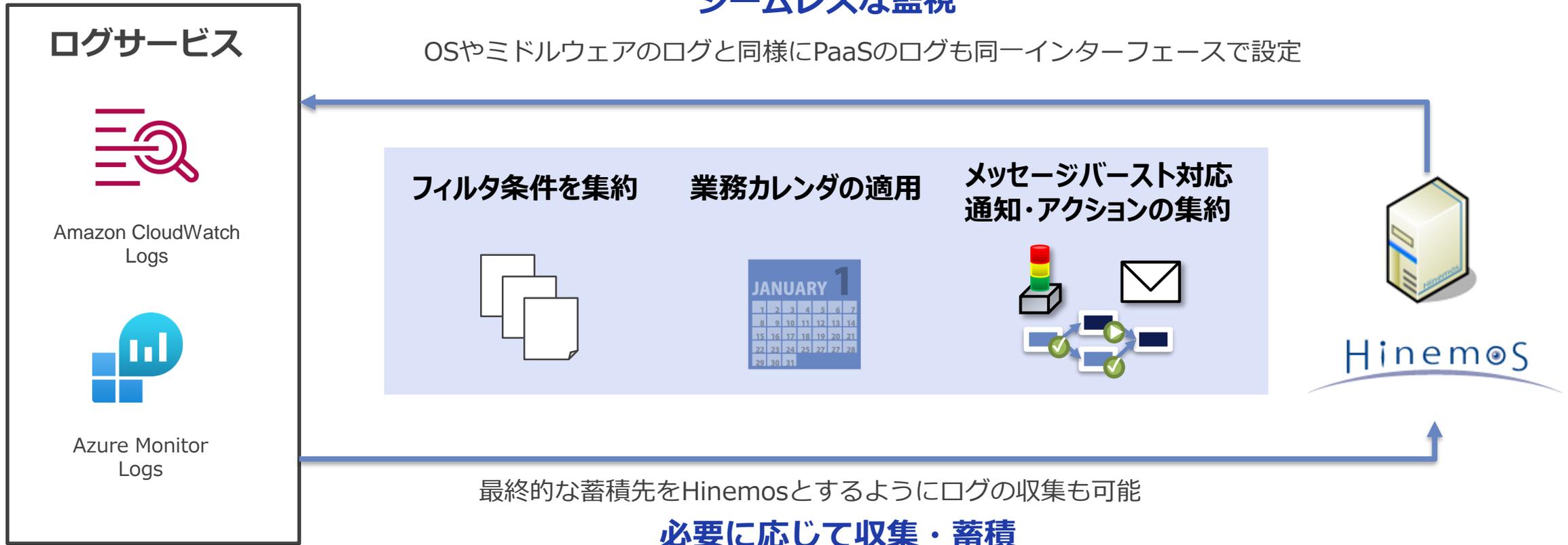
RPAツール	RPA管理製品/サービス	サービスの正常性監視	ミドルウェア監視
WinActor	WinActor MoC	ログイン認証の正常性 ダッシュボードAPIの正常性	-
	WinDirector	ログイン認証の正常性 シナリオ一覧APIの正常性	WinActor Manager Apache Tomcat、 PostgreSQL
UiPath	UiPath Orchestrator (Automation Cloud)	ログイン認証の正常性 ジョブ実行状態APIの正常性	-
	UiPath Orchestrator (オンプレミス)	ログイン認証の正常性 ジョブ実行状態APIの正常性	IIS

# 機能改善

## AWS/Azureのログサービス上に存在するログの監視・収集をシームレスに実現

### シームレスな監視

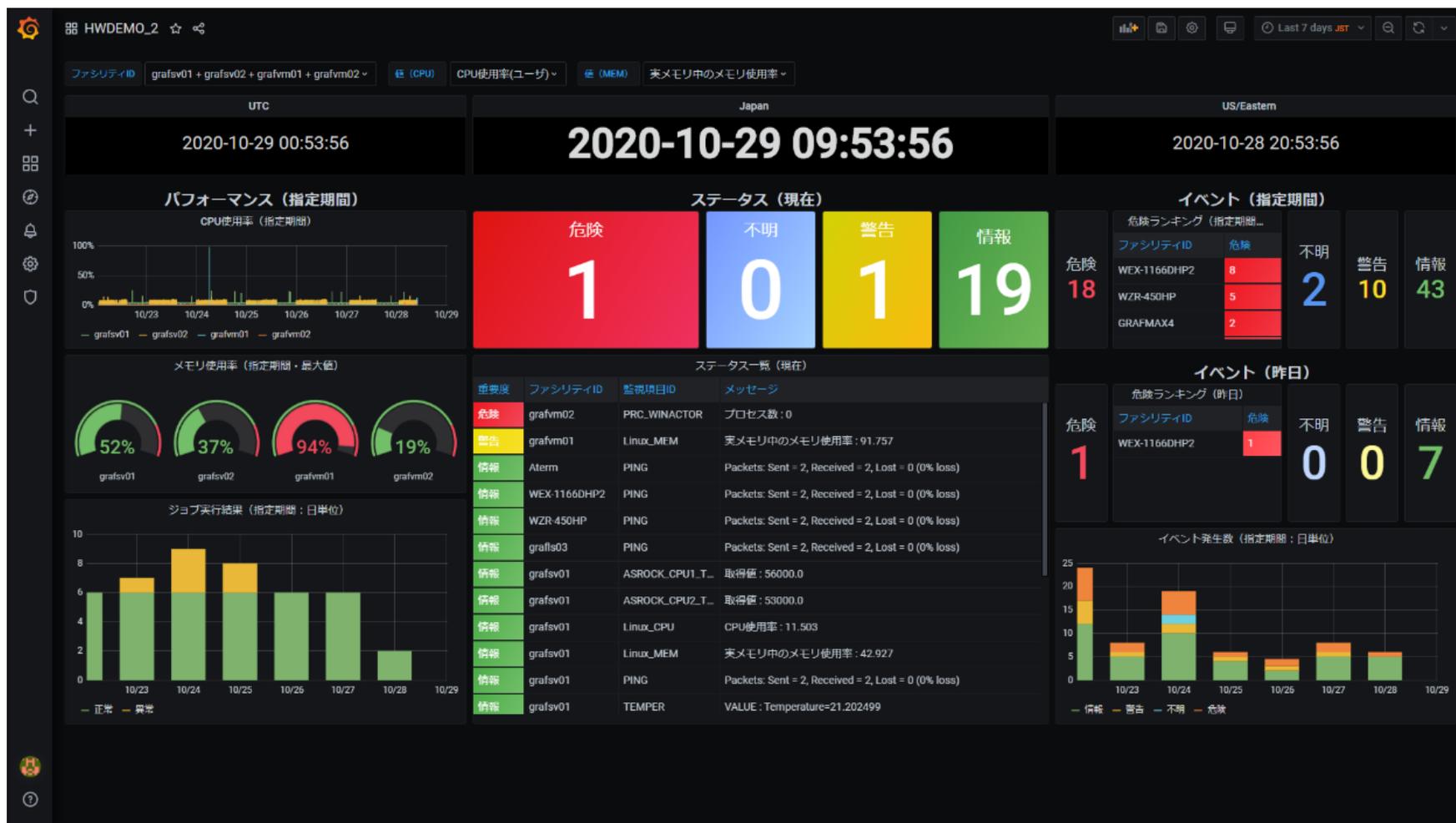
OSやミドルウェアのログと同様にPaaSのログも同一インターフェースで設定



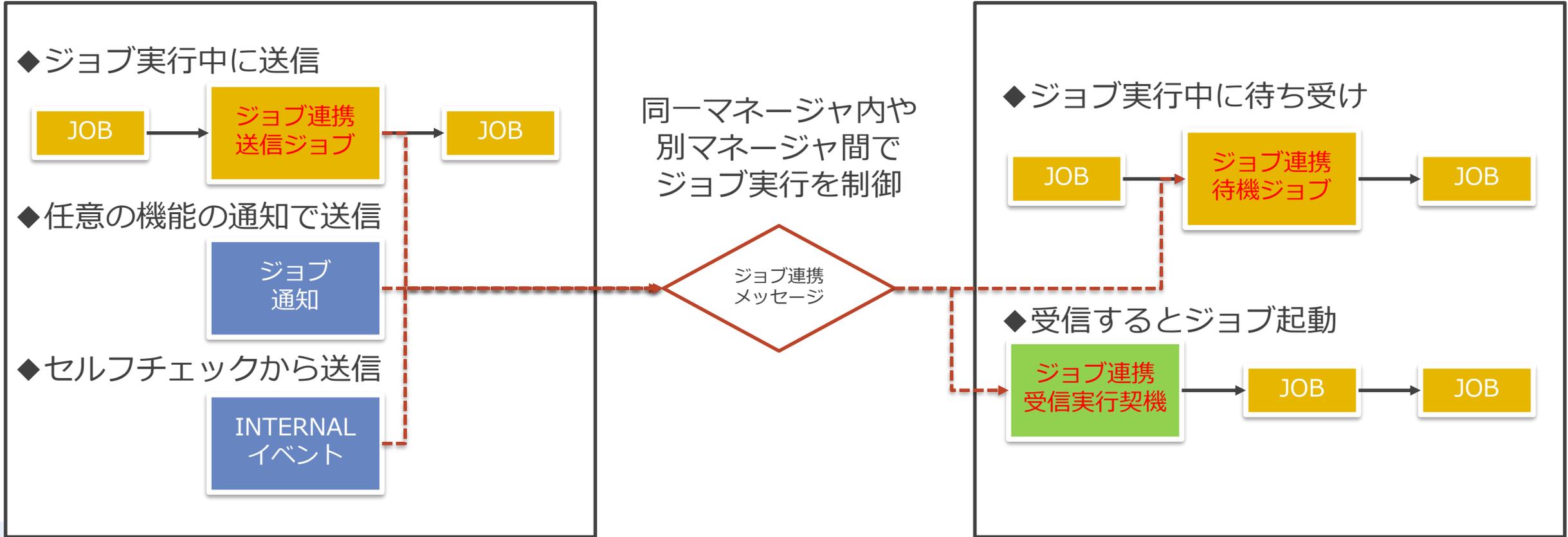
注1) 本機能を利用するにはクラウドのログサービスのログを中継するHinemosエージェントの導入が必要です

注2) Hinemosが蓄積したイベント・性能実績などを収集蓄積機能によりfluentdを介してクラウドのログサービスに転送する事も可能です

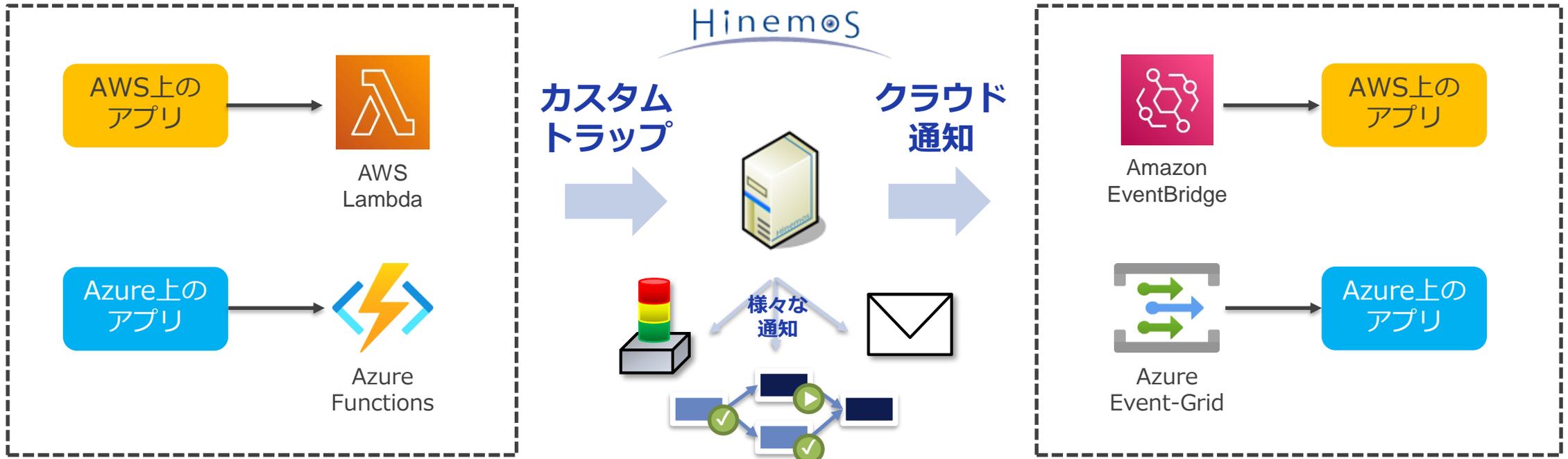
## Grafanaデータソースプラグインを提供



ジョブ連携メッセージを介してマネージャ間を跨るジョブ実行制御が可能に



## 通知のバリエーションとしてAmazon EventbridgeとAzure Eventgridに対応



通知の集約やクラウド間のブリッジの役割を実現

注1) AWS Lambda、Azure Functionからの連携は、カスタムトラップ監視設定のサンプルを提供

# まとめ

Hinemos ver.7.0による運用アナリティクスにより、DXを推進するITシステムの方法論を実現する監視・自動化ツールの統合を実現することをご紹介しました。

Hinemos ver.7.0が、収集・蓄積機能、監視・性能機能、自動化機能、そして共通基本機能から構成されることをご紹介しました。

Hinemos ver.7.0の新機能として、メッセージフィルタ、SDML、RPA管理の機能が追加されたことをご紹介しました。また、Grafanaプラグインなどの代表的な機能改善点をご紹介しました。

これまでの運用管理の枠組みを超え、進化を続けるHinemosにぜひご期待ください。



# NTT DATA

Trusted Global Innovator