

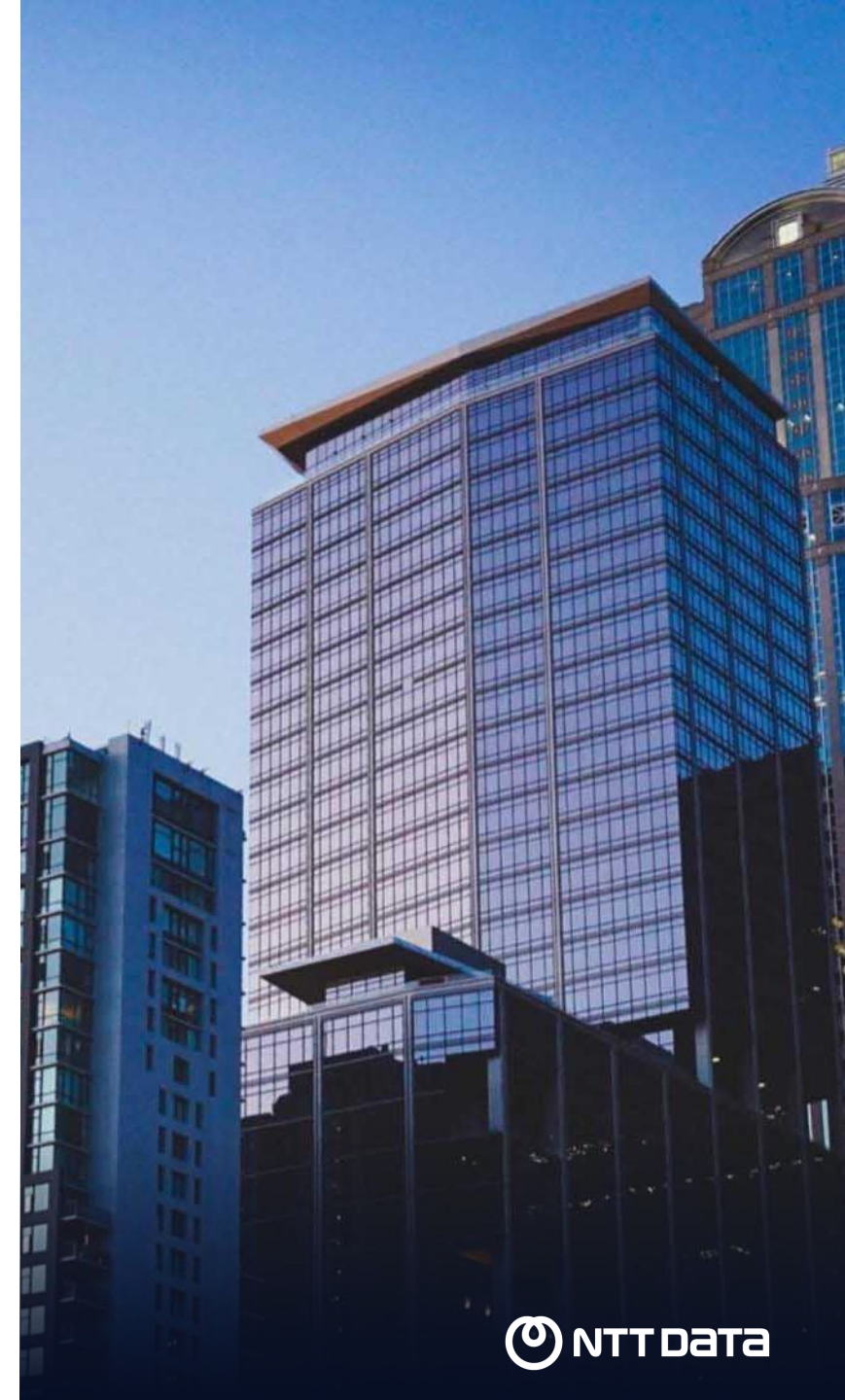
# Hinemos ver.7.2 機能紹介



Hinemos

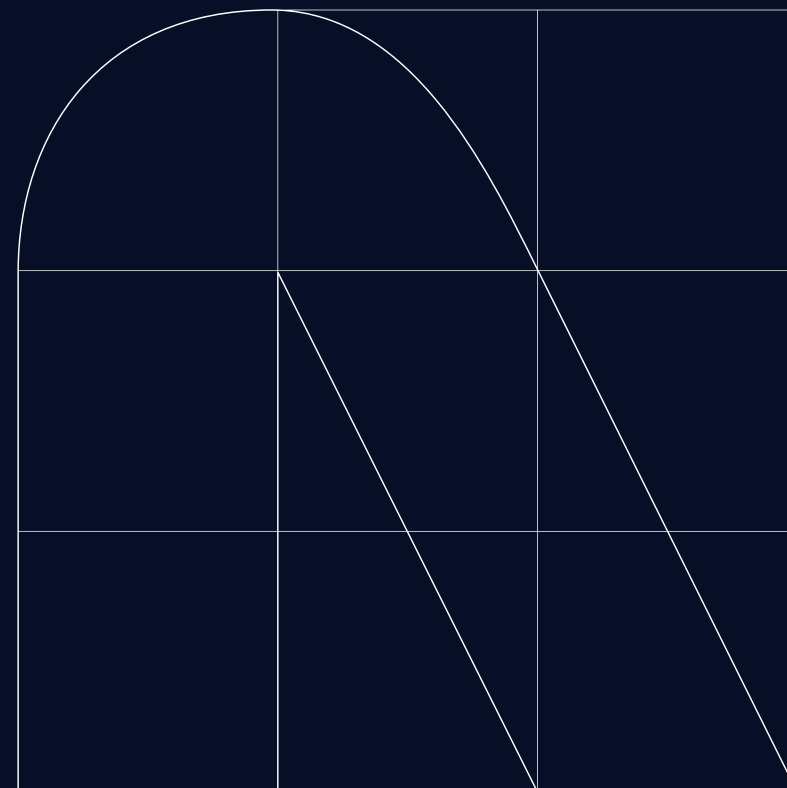
# 目次

1. Hinemosとは
2. Hinemos機能概要
3. Hinemosエンタープライズ機能
4. Hinemos VM・クラウド管理機能
5. Hinemosミッションクリティカル機能
6. Hinemosメッセージフィルタ
7. Hinemosインシデントダッシュボード
8. Appendix



# 01

## Hinemosとは



# Hinemosとは

Hinemos は AI ドリブン運用によりハイブリッドクラウド環境の運用自動化を実現

## 収集・蓄積

### パフォーマンス

- リソース値
- 応答時間
- ログ件数

### システム構成

- プロセス
- パッケージ
- OS

### プラットフォームログ

- システムログ
- NWパケット
- センサーデータ

### アプリ情報

- アプリログ
- プロセス
- リソース

Hinemos

AIトランスフォーメーションを推進する  
ITシステムの方法論をHinemosで統合

AI-Driven Ops

AI-Orchestrated Ops

Autonomous Ops

## 見える化・分析

### 障害検知



### 将来予測



### 構成情報の見える化

- プロセス
- RPM
- Linux
- プロセス
- msi
- Windows

### インテリジェントなフィルタリング



## 自動化

### ジョブ管理・ワークフロー制御



### リソース最適化



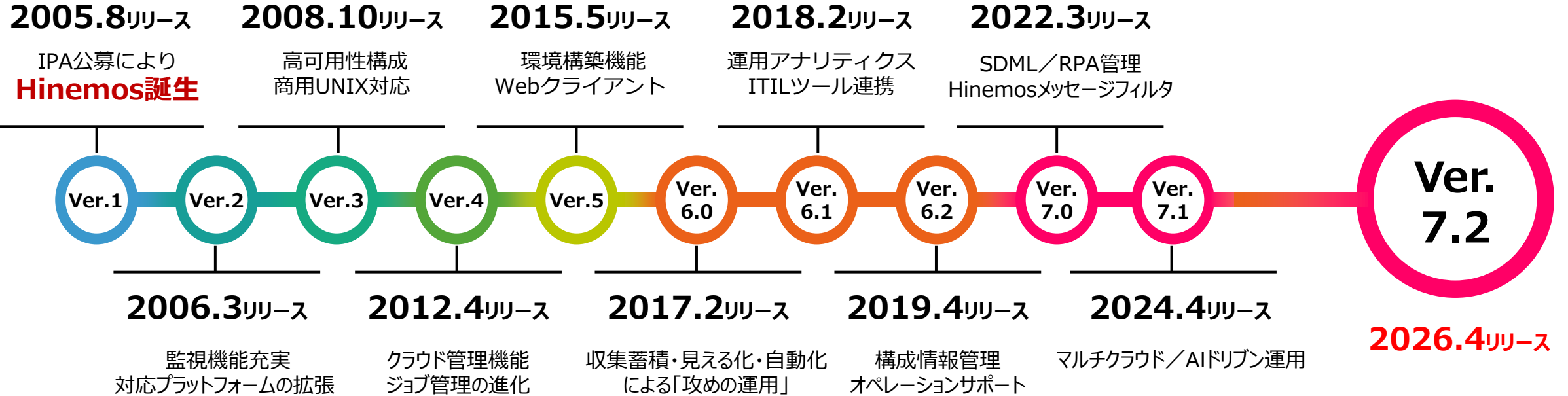
### パッケージの最新化



### AI エージェント



# Hinemosの歴史



統合運用管理ツール部門  
ログ管理システム部門  
ジョブ管理システム部門  
ネットワーク管理部門

「Leader」17期連続！  
「Leader」17期連続！  
「Leader」21期連続！  
「Leader」21期連続！

# Hinemosの導入実績

官公庁・自治体

証券・金融

製造・化学

小売



メディア

電気・ガス・水道

輸送・流通

データセンタ



愛知県庁様(庁内クラウドシステム)  
KNT-CTホールディングス様  
株式会社JTB様  
NTTデータ 各所データセンタ



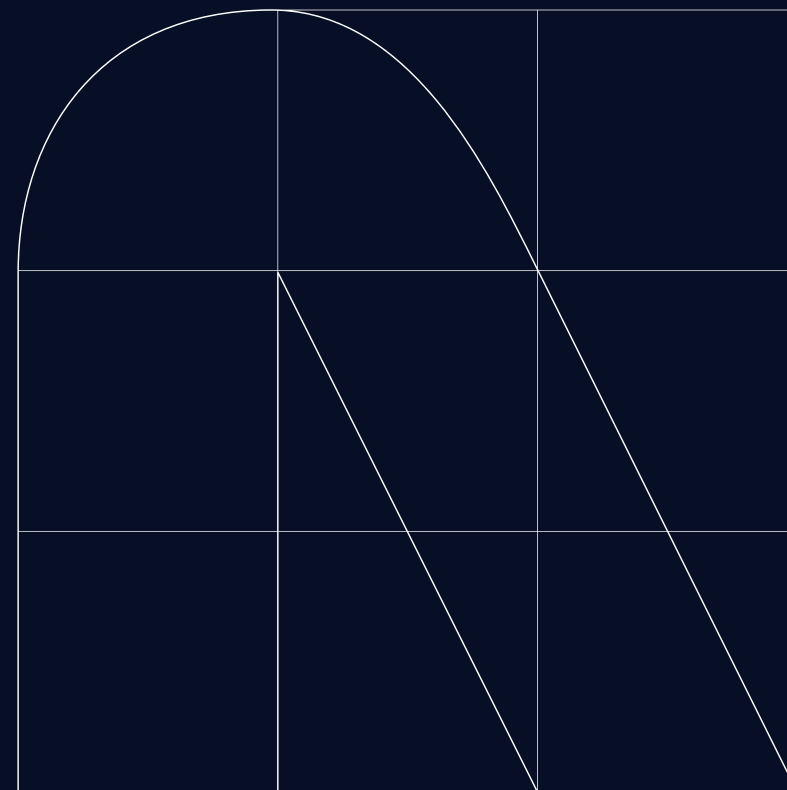
三井住友銀行様  
地方銀行様(多数)  
カード決済サービス(CAFIS)



東急電鉄グループ様  
電通国際情報サービス様  
株式会社NTTドコモ様  
ECONO-CREA (NTTデータ)  
A-gate (NTTデータ)

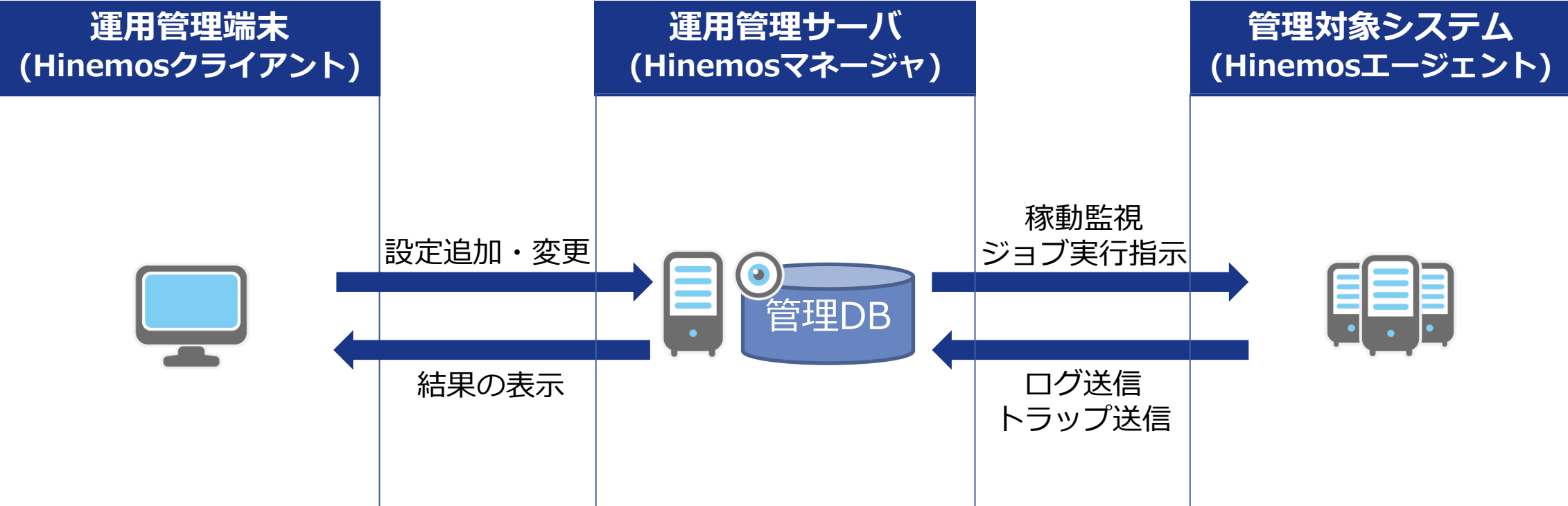
# 02

## Hinemos機能概要



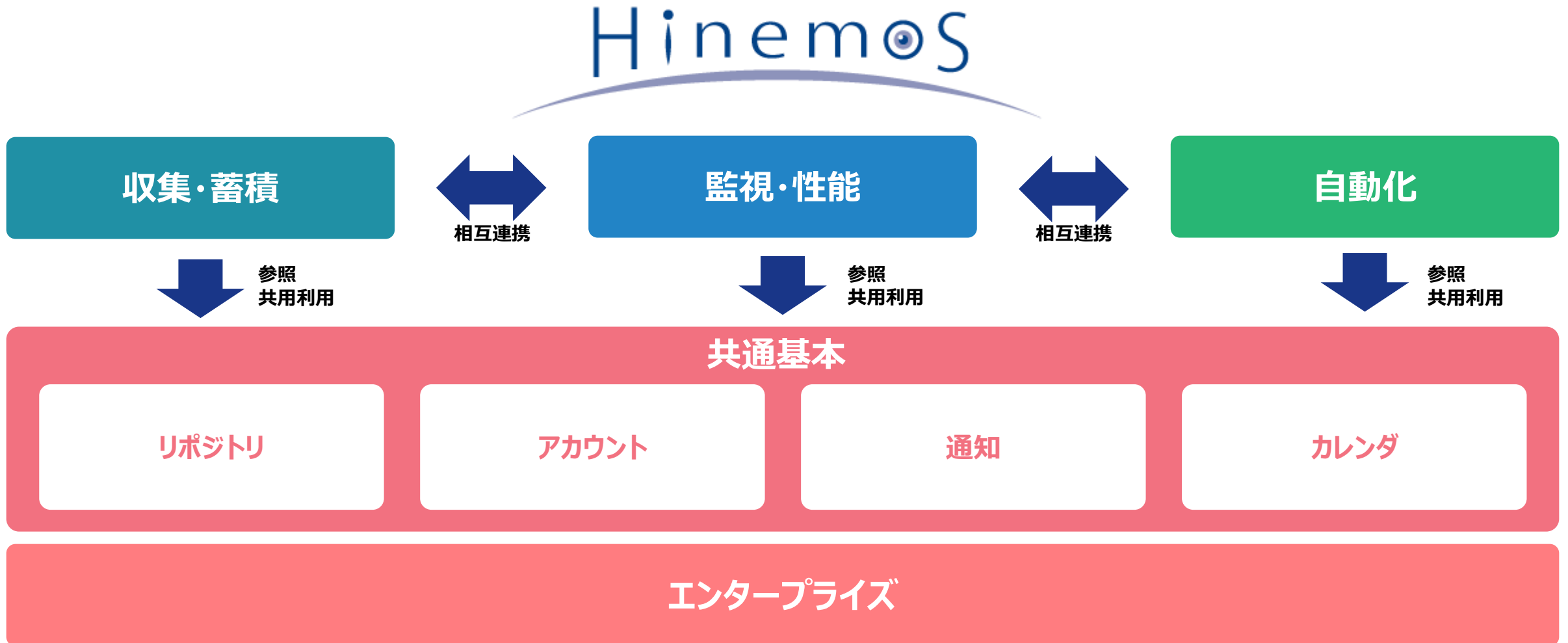
# Hinemosの基本構成

Hinemosはクライアント、マネージャ、エージェントから構成



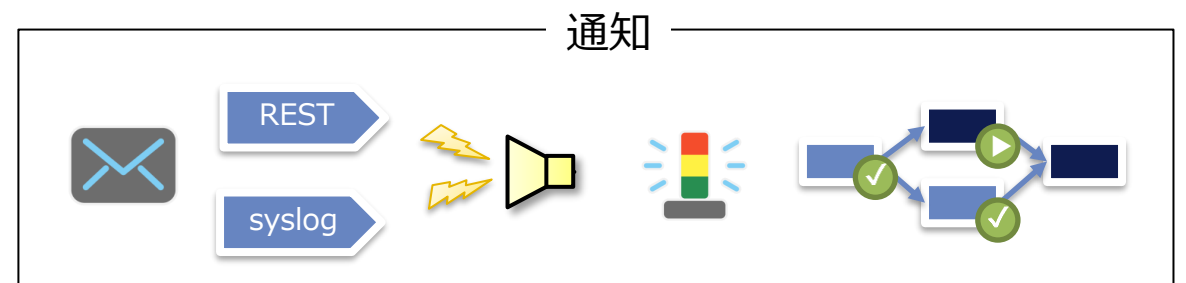
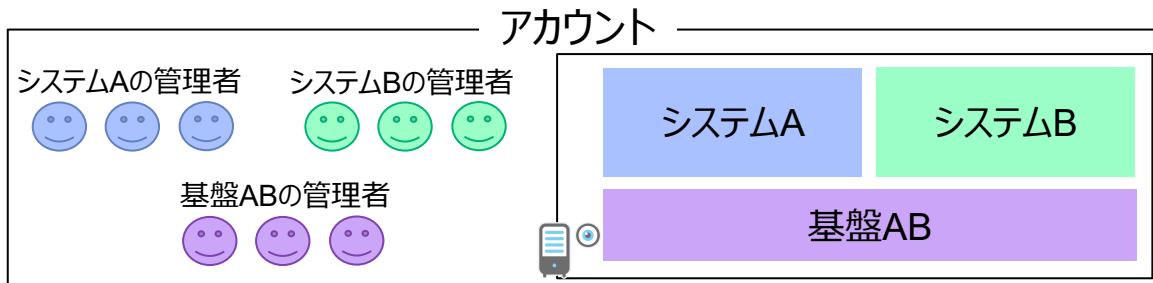
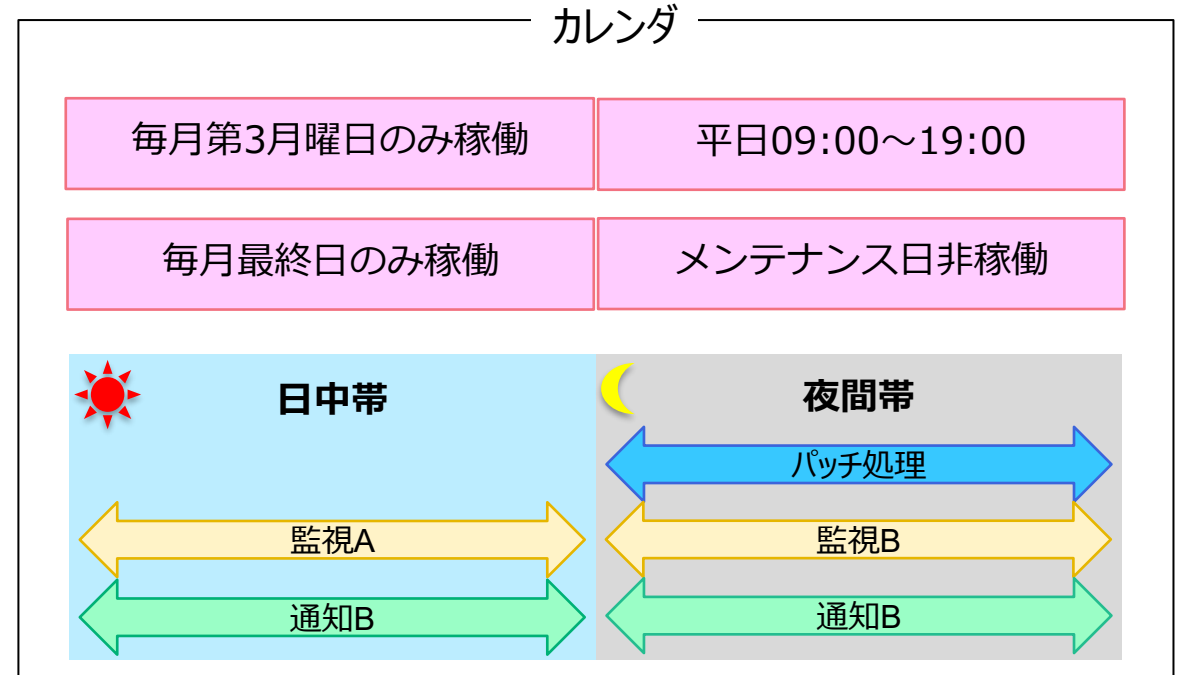
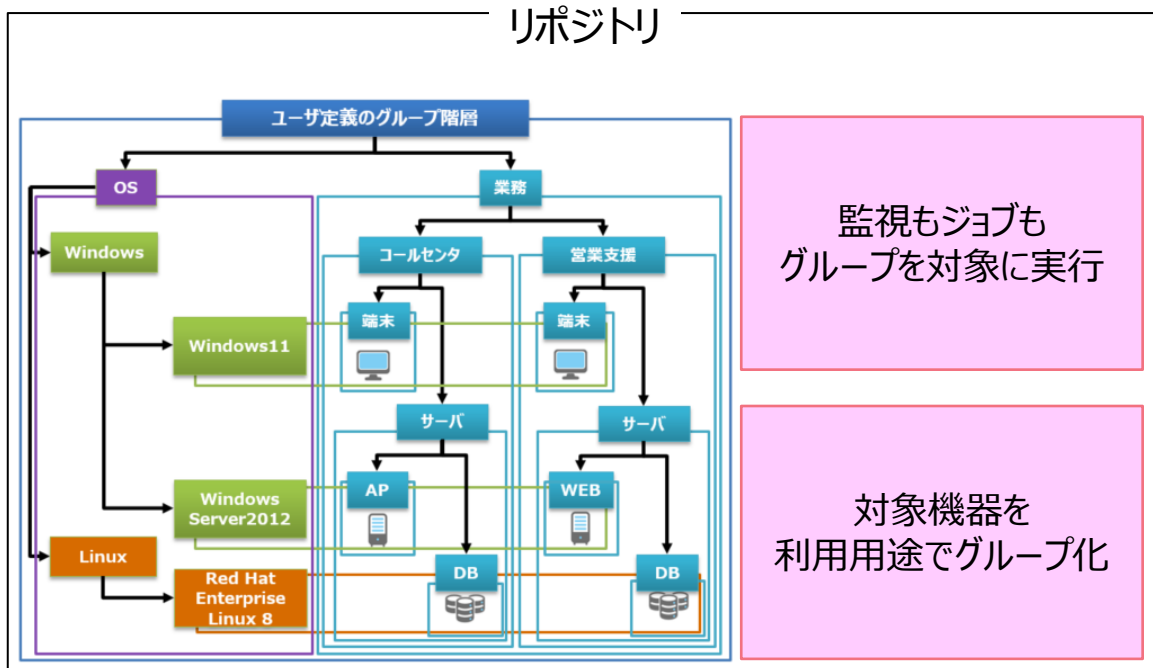
# Hinemos機能概要

共通基本機能とは、Hinemosのコアとして、機能間のシームレスな連携を実現する基本機能



# 共通基本機能

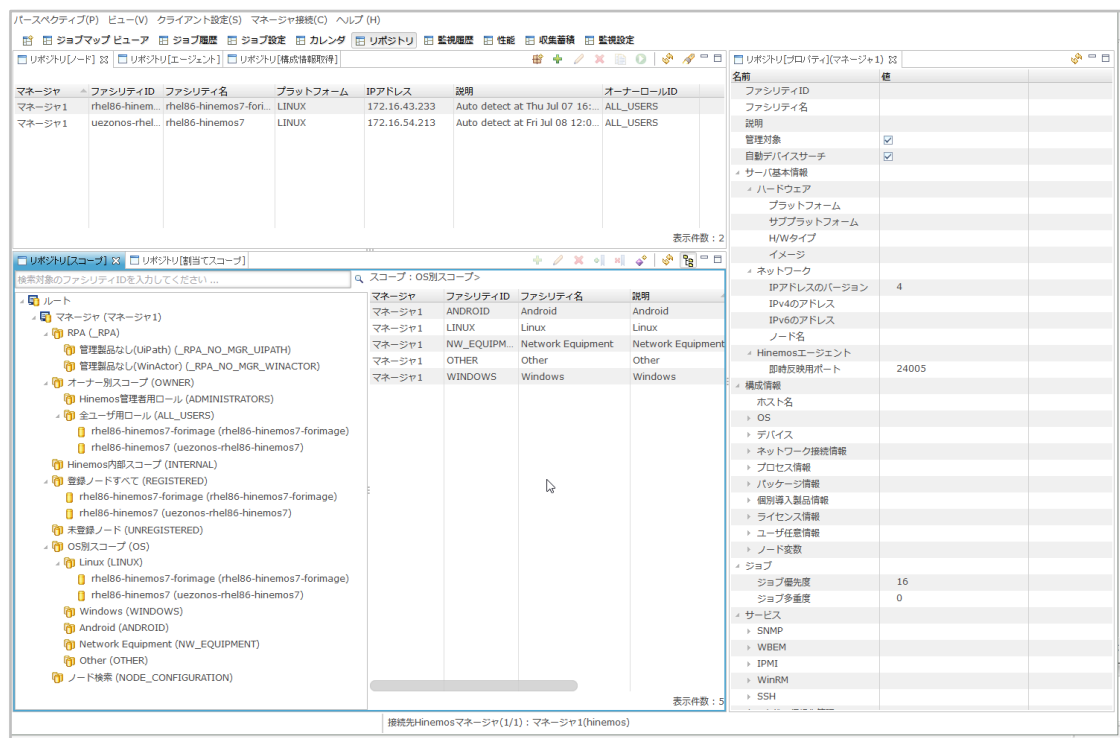
シームレスな機能間連携による統合運用管理を実現



# 共通基本機能 画面イメージ

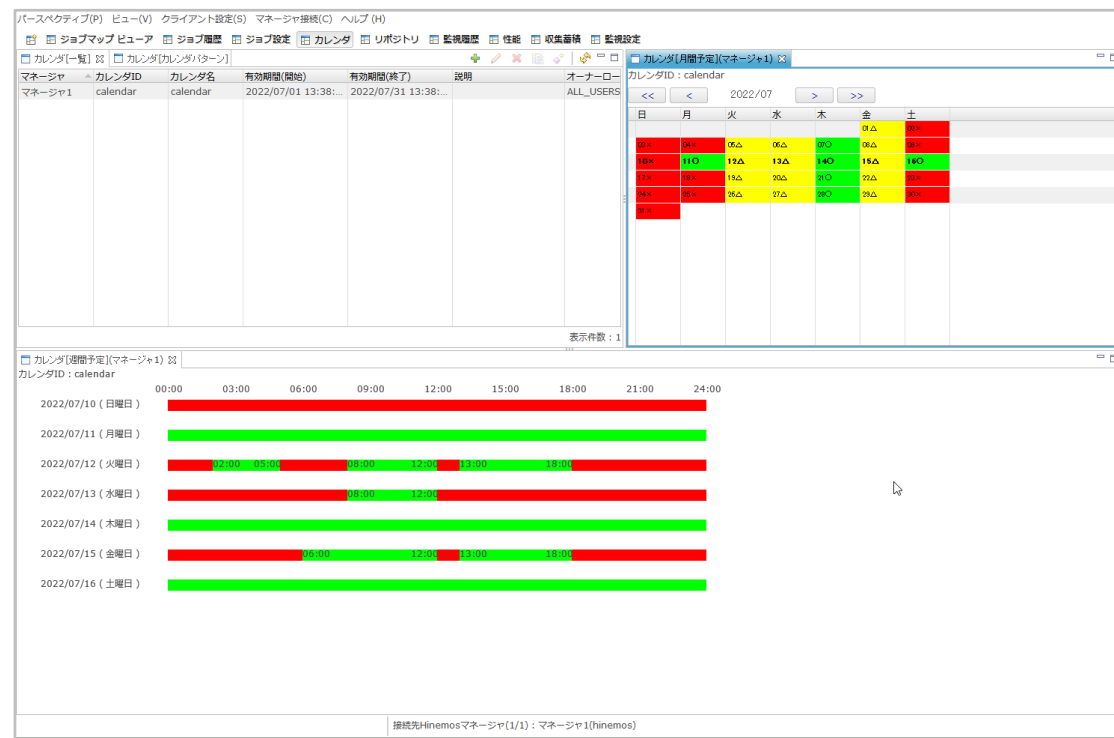
## リポジトリパースペクティブ

システム全体を管理上で必要な情報を統合する機能



## カレンダーパースペクティブ

稼働/非稼働の期間を設定し実行の有無を判断する機能



# 共通基本機能 リポトリ機能 システム構成の管理

管理対象機器（ノード）を利用用途でグルーピング（スコープ）可能

## ●システム構成の管理

### ノード登録

#### ノードサーチ

IPLレンジ指定による機器の自動登録

#### 仮想マシン自動検出

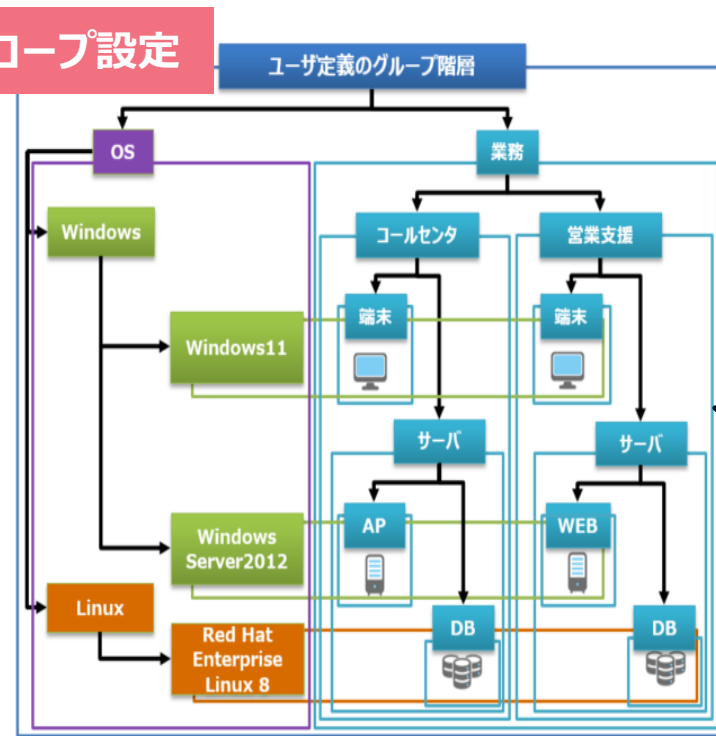
専用API(※1)を使った自動登録  
(VM管理・クラウド管理機能)

(※1)  
HinemosがVmwareやAWSのAPIを呼び出し  
ノード登録に必要な情報を自動で取得します。  
OS種別やVM・クラウド構成も自動識別します。

分類



### スコープ設定



多段階層の  
グルーピングが可能

ノードの  
複数所属が可能

監視・ジョブ実行対象に  
スコープ指定が可能

設計がシンプルに

論理的なグループ単位での監視・ジョブ  
設計が、そのまま設定に反映できる

監視・ジョブ設定数を集約

10,000ノードあっても1スコープに集約  
すれば1設定で監視・ジョブが可能に

テンプレート化が容易に

個々のノードに紐づかない、スコープ単  
位での監視・ジョブ定義が可能に

スコープの多段定義により、複数ノードの管理をサポート

# 共通基本機能 リポジトリ機能 マシン構成の管理

管理対象ノードのデバイス情報のみならず、パッケージ情報・プロセス情報の取得・履歴管理が可能

## ● マシン構成の管理

### ノード情報取得

#### デバイスサーチ

監視に必要なデバイス情報の取得

#### 構成情報取得・履歴管理

パッケージプロセス情報(※2)を含めた構成情報の取得と履歴管理

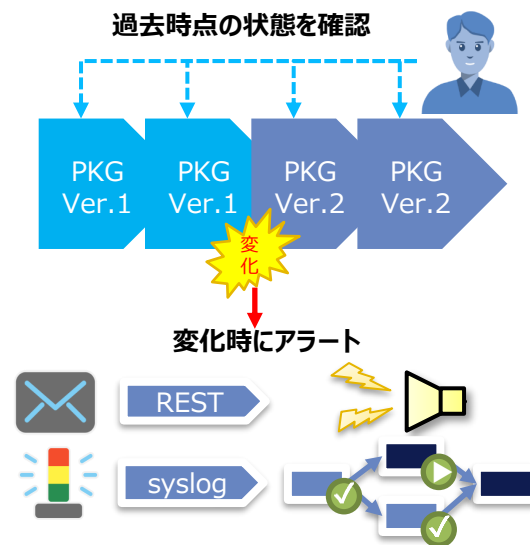
(※2)  
パッケージ・プロセス情報は以下を指します。  
・Windowsパッケージ情報：プログラム  
・Linuxパッケージ情報：RPM/deb  
・個別インストールパッケージ情報：プロセス

活用

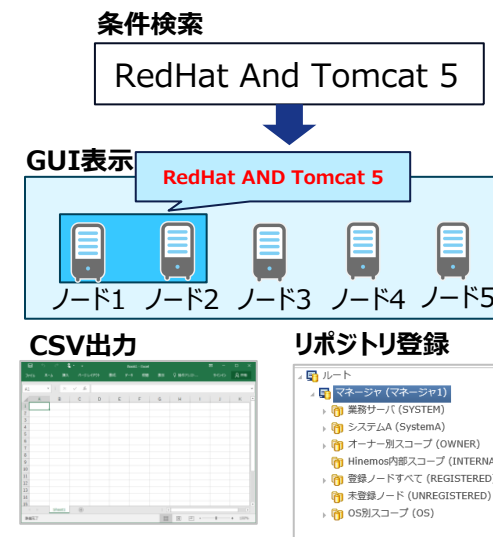


### ノード情報活用

#### ◆ 変更履歴の確認とアラート



#### ◆ 構成情報の検索と見える化



### 構成情報の管理工数削減

初期作成後、メンテナンスが潰れがちな構成情報もHinemosが自動で取得

### 早急な脆弱性点検が可能に

定期・不定期に発生する脆弱性問題に対する点検が検索するだけで実現

### 構成変更のサイクルを実現

対象特定→構成変更→結果確認のサイクルを全てHinemosだけで実現

構成情報管理の工数削減、早急な脆弱性対応を実現

# 共通基本機能

# 通知機能・カレンダー機能

Hinemosの各機能は、実行結果を様々な形で通知し、  
業務の営業日・メンテナンス日に合わせたカレンダー定義が可能

## 通知機能

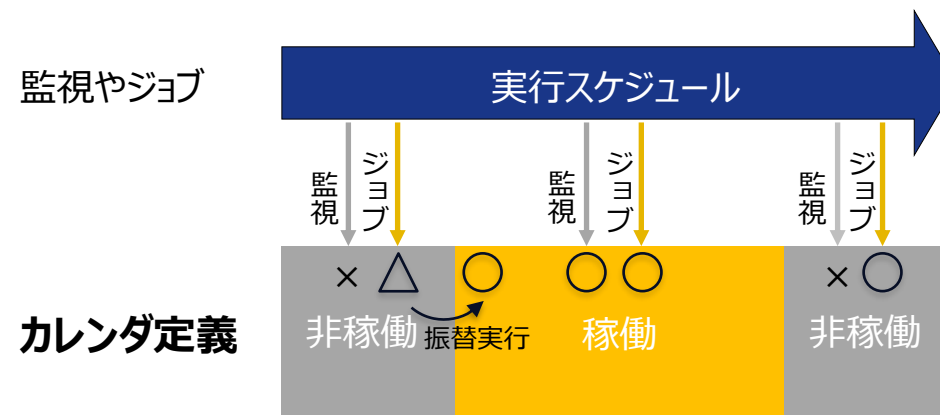
監視やジョブの実行結果に応じて通知を実施

種類	用途
ステータス通知	Hinemosクライアント表示（最新ステータス）
イベント通知	Hinemosクライアント表示（イベント履歴）
メール通知	メール送信、サービス連携
ジョブ通知	復旧処理等の処理の自動実行
ログエスケレーション通知	syslog送信、サービス連携
コマンド通知	コマンド実行、パトライド鳴動等
環境構築通知	サーバ、PCセットアップ処理の自動実行
クラウド通知	Amazon EventBridge、Azure Event Grid連携
REST通知	RESTメッセージ送信、サービス連携
メッセージ通知	メッセージフィルタ機能連携
電話通知	電話/SMSで担当者に自動連絡

瞬間的な異常やメッセージラッシュにおける  
通知抑制が可能

## カレンダー機能

監視やジョブを実行する日、実行しない日を定義可能



### 定期的なカレンダー

- ・年月日、曜日の定期ルールで指定可
- ・前後日指定で、日数ずらした定義可
- ・非稼働の場合の振替日、振替回数を指定可

### 不定期なカレンダーパターン

- ・定期的ではない祝日パターンを指定可
- ・会社の創立記念日の指定可
- ・緊急メンテナンス日の指定可




稼働スケジュールを視覚的に確認可能

# 共通基本機能 アカウント機能・マルチマネージャ対応

Hinemosは、ロールを指定するアカウント機能や、  
複数のマネージャに接続可能なマルチマネージャ対応を実現

## アカウント機能

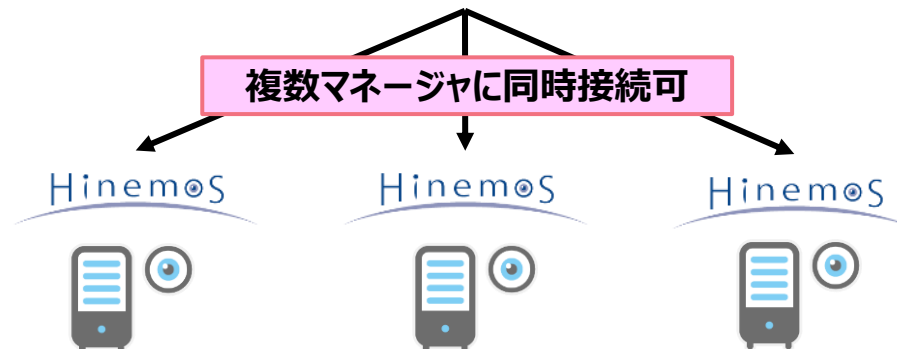
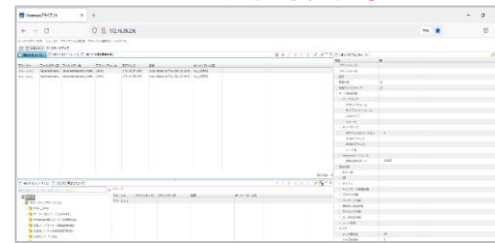
各アカウントごとにロール（役割）の指定可能

		可視範囲の制限	
		システム A	システム B
操作範囲の制限	監視設定参照 ジョブ定義参照	 システムA 運用者	 システムB 運用者
	監視設定変更 ジョブ定義実行	 システムA SE	 システムB SE

## クライアントのマルチマネージャ対応

一度に複数のマネージャに接続可能

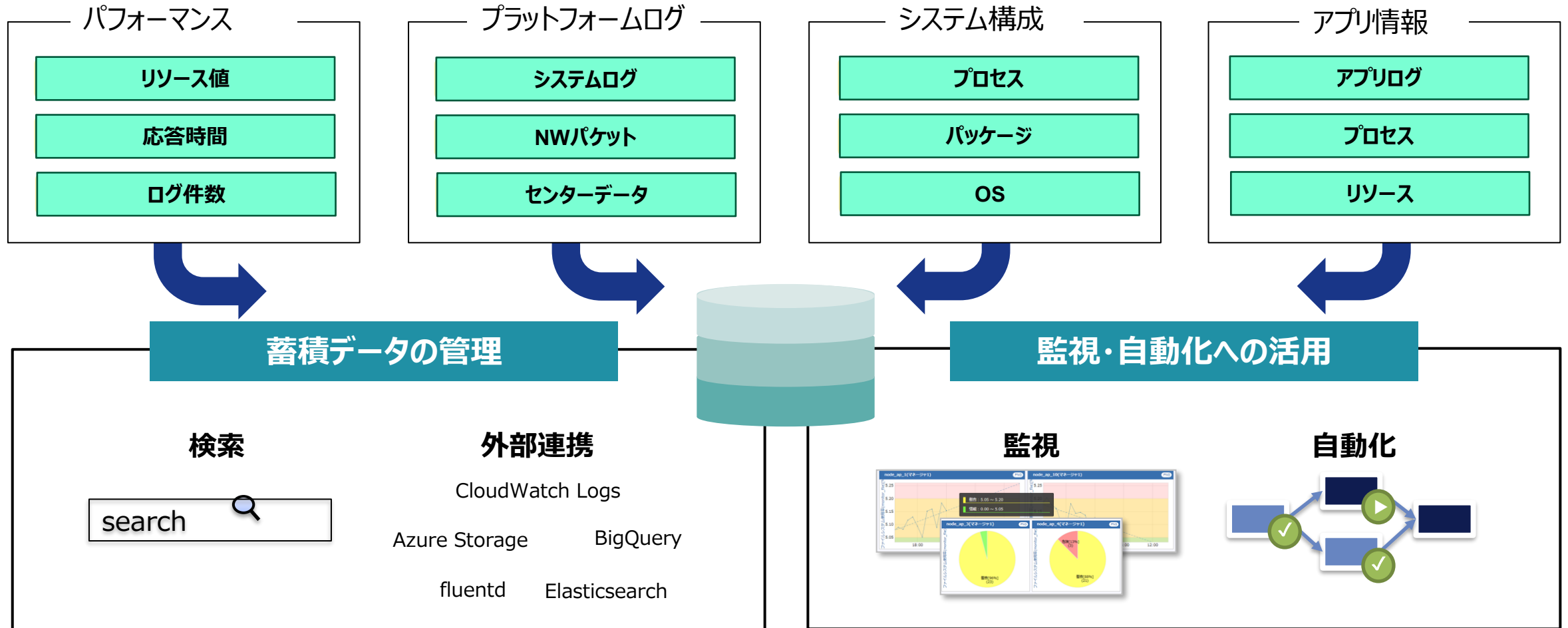
端末にインストール不要な  
Webクライアント



スコープの多段定義により、複数ノードの管理をサポート

# 収集・蓄積

サーバ・NW 機器のパフォーマンス情報、システムのインフラ情報のみならず、ミドルウェア、アプリケーション、IoT 端末など、多種多様な情報を収集・管理することが可能



# 収集・蓄積 画面イメージ

## 収集設定

詳細な収集設定が可能

通知 (イベント) [作成・変更]

マネージャ: マネージャ1

通知ID: [ ]

説明: [ ]

オーナーロールID: ALL\_USERS

カレンダーID: [ ]

重要度変化後の初回通知

同じ重要度の監視結果が [ ] 回以上連続した場合に初めて通知する

有効にした直後は通知しない

重要度変化後の二回目以降の通知

常に通知する

前回通知から [ ] 分間は同一重要度の通知はしない

通知しない

イベント通知

重要度	通知	状態
強制	<input type="checkbox"/>	未確認
警告	<input type="checkbox"/>	未確認
注意	<input type="checkbox"/>	未確認
不明	<input type="checkbox"/>	未確認

ステータス通知

重要度変化後の二回目以降の通知

常に通知する

前回通知から [ ] 分間は同一重要度の通知はしない

通知しない

ステータス情報の蓄積期間: 10分

蓄積期間経過後の処理:

情報を削除する

更新されていない旨のメッセージに置き換える

重要度: [ ]

この設定を有効にする

OK(O) キャンセル(C)

## 収集蓄積パースペクティブ

多種多様な情報を収集・管理することが可能

パースペクティブ(P) ビュー(V) クライアント設定(S) マネージャ接続(C) ヘルプ(H)

ジョブマップ ビューア | ジョブ履歴 | 性能 | 収集蓄積 | 監視設定 | 監視履歴

収集蓄積[スコープワザ] | 検索[マネージャ1:rhel86-hinemos7-forimage]

検索対象のファシリティIDを入力してください

文字列収集データ | バイナリ収集データ

期間指定: すべて | 2022/07/08 08:38:55 ~ 2022/07/08 08:38:55

監視項目ID: [ ]

キーワード: ex\_pattern%  AND  OR

エンコーディング: [ ] 検索 ダウンロード

時刻	ファシリティ	監視ID	メッセージ	タグ
2022/07/08 08:36	rhel86-hi	mon_log	Jul 8 08:30:36 rhel86-hinemos7-forimage systemd[1]: Started system activity accounting tool.	filename=/var/log/messages
2022/07/08 08:36	rhel86-hi	mon_log	Jul 8 08:30:36 rhel86-hinemos7-forimage systemd[1]: sysstat-collect.service: Succeeded.	filename=/var/log/messages
2022/07/08 08:36	rhel86-hi	mon_log	Jul 8 08:30:36 rhel86-hinemos7-forimage systemd[1]: Starting system activity accounting tool.	filename=/var/log/messages
2022/07/08 08:36	rhel86-hi	mon_log	Jul 8 08:20:36 rhel86-hinemos7-forimage systemd[1]: Started system activity accounting tool.	filename=/var/log/messages
2022/07/08 08:36	rhel86-hi	mon_log	Jul 8 08:20:36 rhel86-hinemos7-forimage systemd[1]: sysstat-collect.service: Succeeded.	filename=/var/log/messages
2022/07/08 08:36	rhel86-hi	mon_log	Jul 8 08:20:36 rhel86-hinemos7-forimage systemd[1]: Starting system activity accounting tool.	filename=/var/log/messages
2022/07/08 08:10	rhel86-hi	mon_log	Jul 8 08:10:25 rhel86-hinemos7-forimage systemd[1]: Started system activity accounting tool.	filename=/var/log/messages
2022/07/08 08:10	rhel86-hi	mon_log	Jul 8 08:10:25 rhel86-hinemos7-forimage systemd[1]: sysstat-collect.service: Succeeded.	filename=/var/log/messages
2022/07/08 08:10	rhel86-hi	mon_log	Jul 8 08:10:25 rhel86-hinemos7-forimage systemd[1]: Starting system activity accounting tool.	filename=/var/log/messages
2022/07/08 08:00	rhel86-hi	mon_log	Jul 8 08:00:36 rhel86-hinemos7-forimage systemd[1]: sysstat-collect.service: Succeeded.	filename=/var/log/messages
2022/07/08 07:50	rhel86-hi	mon_log	Jul 8 07:50:36 rhel86-hinemos7-forimage systemd[1]: Started system activity accounting tool.	filename=/var/log/messages
2022/07/08 07:50	rhel86-hi	mon_log	Jul 8 07:50:36 rhel86-hinemos7-forimage systemd[1]: Starting system activity accounting tool.	filename=/var/log/messages
2022/07/08 07:42	rhel86-hi	mon_log	Jul 8 07:42:46 rhel86-hinemos7-forimage systemd[1]: Started dnf makecache.	filename=/var/log/messages
2022/07/08 07:42	rhel86-hi	mon_log	Jul 8 07:42:46 rhel86-hinemos7-forimage systemd[1]: dnf-makecache.service: Succeeded.	filename=/var/log/messages
2022/07/08 07:42	rhel86-hi	mon_log	Jul 8 07:42:46 rhel86-hinemos7-forimage dnf[12313]: メタデータキャッシュは最近、リフレッシュされま	filename=/var/log/messages
2022/07/08 07:42	rhel86-hi	mon_log	Jul 8 07:42:46 rhel86-hinemos7-forimage dnf[12313]: このシステムは、エンタイトルドユーザーに登	filename=/var/log/messages
2022/07/08 07:42	rhel86-hi	mon_log	Jul 8 07:42:45 rhel86-hinemos7-forimage dnf[12313]: コンシューマー個別子を読みません	filename=/var/log/messages
2022/07/08 07:42	rhel86-hi	mon_log	Jul 8 07:42:45 rhel86-hinemos7-forimage dnf[12313]: サブスクリプション管理リポジトリを更新してい	filename=/var/log/messages
2022/07/08 07:40	rhel86-hi	mon_log	Jul 8 07:40:25 rhel86-hinemos7-forimage systemd[1]: Starting dnf makecache...	filename=/var/log/messages
2022/07/08 07:40	rhel86-hi	mon_log	Jul 8 07:40:25 rhel86-hinemos7-forimage systemd[1]: Started system activity accounting tool.	filename=/var/log/messages
2022/07/08 07:40	rhel86-hi	mon_log	Jul 8 07:40:25 rhel86-hinemos7-forimage systemd[1]: sysstat-collect.service: Succeeded.	filename=/var/log/messages
2022/07/08 07:40	rhel86-hi	mon_log	Jul 8 07:40:25 rhel86-hinemos7-forimage systemd[1]: Starting system activity accounting tool.	filename=/var/log/messages
2022/07/08 07:30	rhel86-hi	mon_log	Jul 8 07:30:36 rhel86-hinemos7-forimage systemd[1]: Started system activity accounting tool.	filename=/var/log/messages

合計: 396件中 1 ~ 100件表示 (0.07 s)

接続先hinemosマネージャ(1/1): マネージャ1(hinemos)

# 収集・蓄積 豊富な対応ログの種類(1/3)

ログの収集と監視を統一的なインターフェースで提供し、非常に様々な種類のログもGUI操作だけで収集可能

## Hinemosが収集できるログ

様々なログ収集が可能

### 数値データ

PING応答時間  
プロセス数  
Web応答時間  
各種リソース情報  
各種サービスの応答時間  
各種ポートの応答時間  
SNMPレスポンス  
SQLレスポンス  
JMXレスポンス  
ログ件数  
相関係数  
コマンド実行結果  
JSONメッセージ

### 文字列データ

ログファイル  
syslog  
Windowsイベント  
Webレスポンス  
SNMPレスポンス  
SNMPTRAP  
SQLレスポンス  
コマンド実行結果  
JSONメッセージ  
SDML制御ログ  
SDML監視ログ

### バイナリデータ

バイナリファイル  
NWキャプチャ

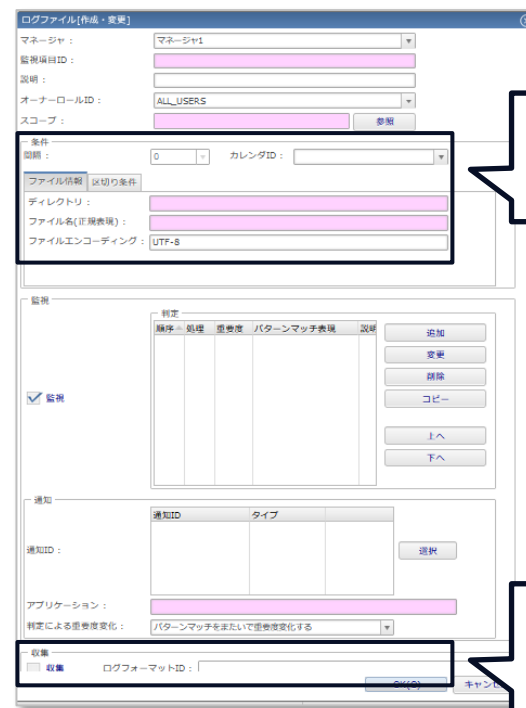
### Hinemos実行履歴

イベント履歴  
ジョブ履歴



## Hinemosのログファイル監視の設定

ログ種別ごとに詳細設定が可能



ログの種類別の  
固有設定

チェックを入れると  
収集・蓄積

収集対象が異なっても基本的な画面構成は同じであるため、非常に簡単に製品を習熟可能

# 収集・蓄積 豊富な対応ログの種類(2/3)

Hinemosは、複数行ログやクラウドログサービスに対応

## 複数行ログ対応

Hinemosは複数行ログに対応

### 代表的な複数行ログ

#### ■ Javaスタックトレース

```
java.lang.NumberFormatException: For input string: "1.1"  
    at java.lang.NumberFormatException.forInputString(Unknown Source)  
    at java.lang.Integer.parseInt(Unknown Source)  
    at java.lang.Integer.<init>(Unknown Source)  
    at ExceptionPrintDemo.formatInt(ExceptionPrintDemo.java:7)  
    at ExceptionPrintDemo.main(ExceptionPrintDemo.java:14)
```

#### ■ Oracleアラートログ

```
Sat Feb 07 12:35:53 2015  
create tablespace TEST_SPACE datafile size 5m autoextend on  
Completed: create tablespace TEST_SPACE datafile size 5m autoextend on  
Sat Feb 07 12:38:40 2015  
alter database datafile  
'/u01/app/oracle/oradata/CDB122/0/datafile/o1_mf_test_spa_bfc20s53_.dbf'  
resize 8m
```

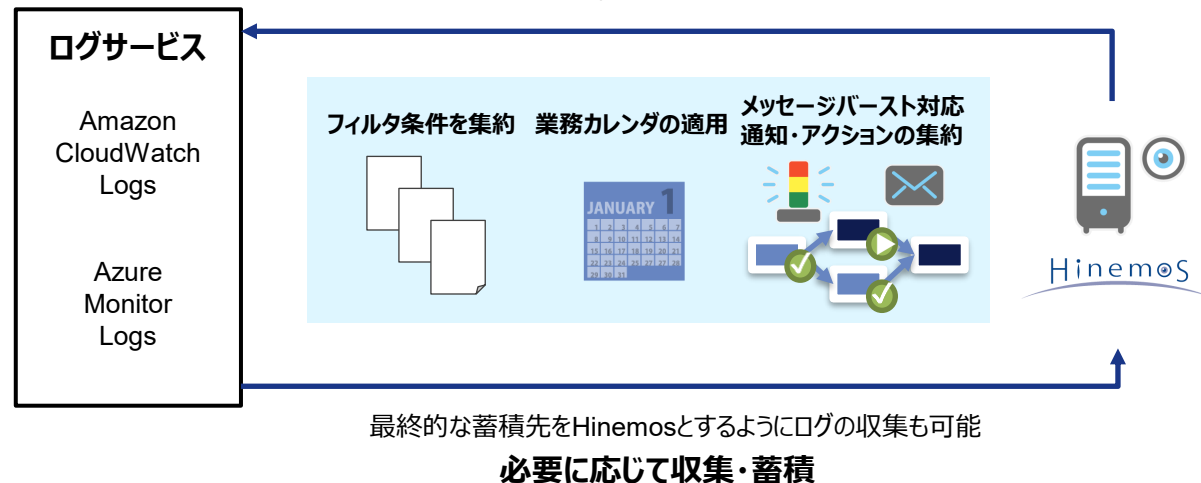
複数行ログ対応で作業効率向上

## クラウドログサービス対応

AWS/Azureのログサービス上に存在するログの監視・収集を実現

### 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



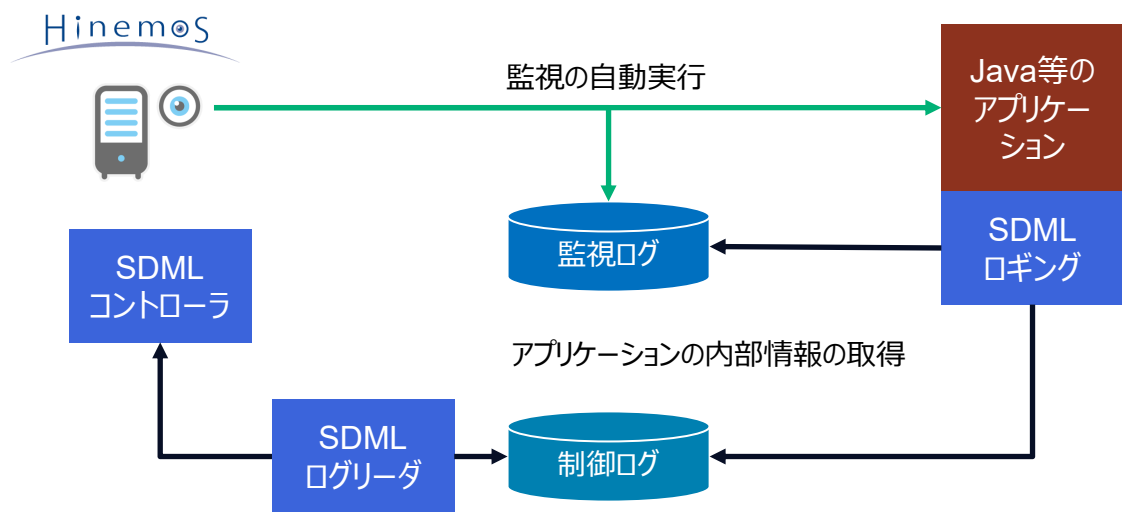
オンプレ・クラウドを跨った統合ログ管理を実現

# 収集・蓄積 豊富な対応ログの種類(3/3)

アプリケーションの内部情報の自動監視や、タグによるログ検索で管理者業務をサポート

## アプリケーションの内部情報の取得

SDML(Software Defined Monitoring and Logging)機能ではアプリケーションの自動監視を実現



自動監視、自動取得の実現

## タグを使った容易な検索

タグ付けをしてメタ情報として合わせて蓄積する  
ログフォーマット機能を完備

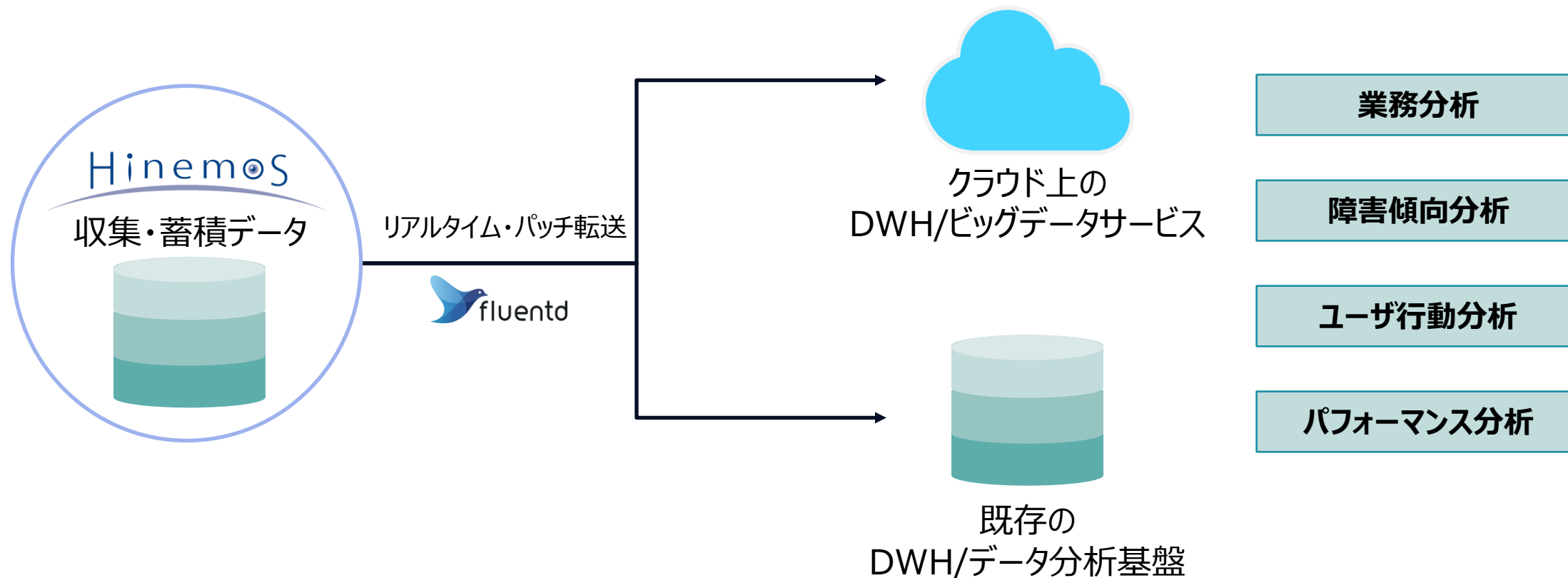
The screenshot shows a search interface. At the top, a box labeled '検索条件' (Search conditions) contains the text 'status\_code=200'. Below this, a log entry is shown: '■ Apacheアクセスログ' followed by a red link '127.0.0.1 - frank [10/Oct/2000:13:55:36 -0700] "GET /apache\_pb.gif HTTP/1.0" 200 2326'. Below the log entry, there are two database icons: '生ログ' (Raw log) and 'メタ情報' (Meta-information). To the right, a table lists the key-value pairs for the log entry.

キー	値
src_ipaddress	127.0.0.1
date	10/Oct/2000:13:55:36 -0700
method	GET
status_code	200
send_byte	2326

ログの検索の利便化を実現

# 収集・蓄積 データ分析基盤・クラウドストレージ連携

収集・蓄積データは、GUIの簡単な設定にて  
外部のDWH/BigDATAサービスやデータ分析基盤に転送し連携することが可能

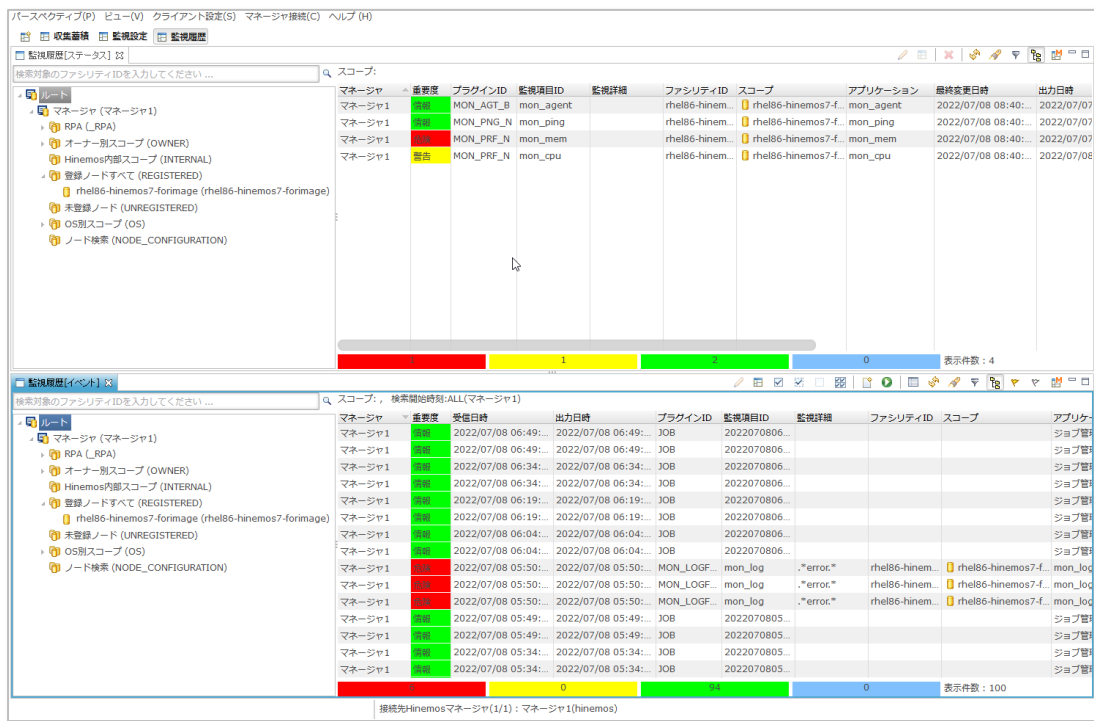


業務分析・ユーザ行動分析など、Hinemosで収集・蓄積したインフラ情報を組み合わせることで、  
これまでとは異なる視点での解析を実現

# 監視・性能 画面イメージ

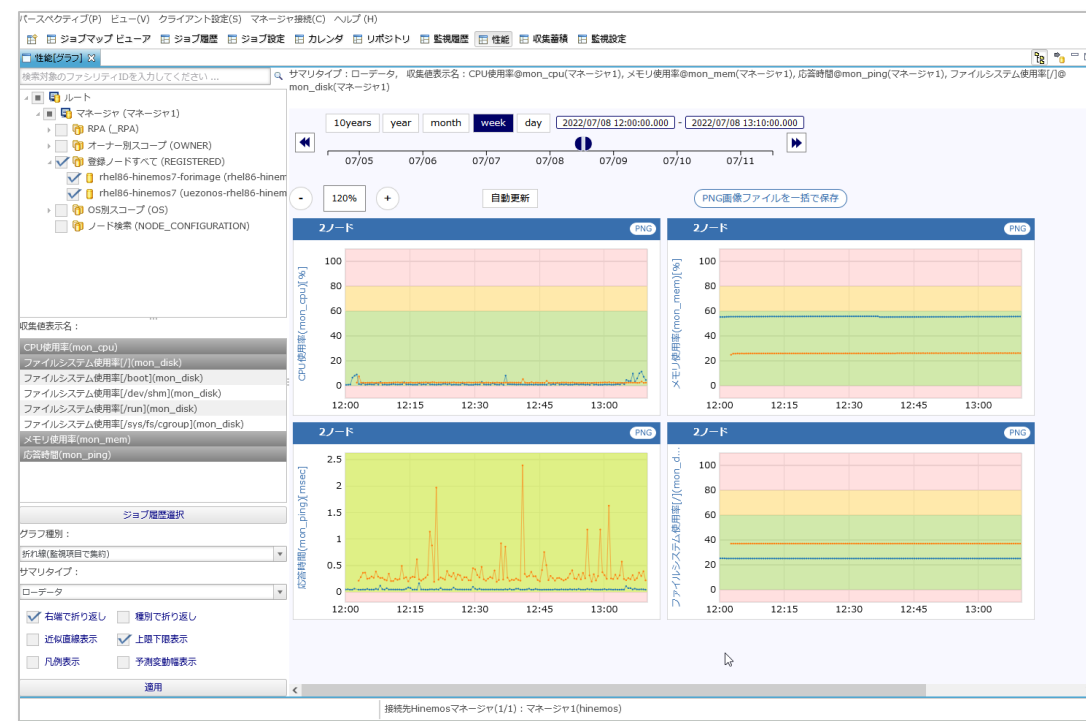
## 監視履歴パースペクティブ

運用オペレータによる管理を実施する機能



## 性能パースペクティブ

数値監視の収集値のグラフ表示、ファイル出力する機能



# 監視・性能 システム全体の集中・集約管理と運用自動化

Hinemosはシステムの集約管理、監視からの様々なアクション連携を実現

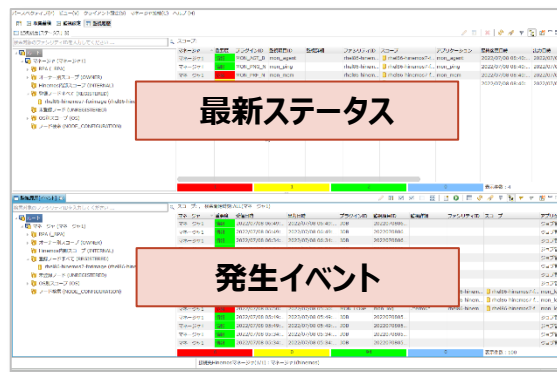
## システム全体の集中・集約管理

システムを構成する各環境を集中・集約管理する事が可能



様々な環境のシステム

発生イベント集約

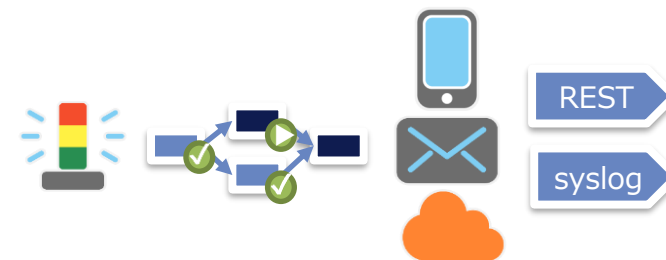


Hinemosによる集中・集約管理

- ・閾値判定
- ・OK/NG判定
- ・パターンマッチ
- ・組み合わせ判定

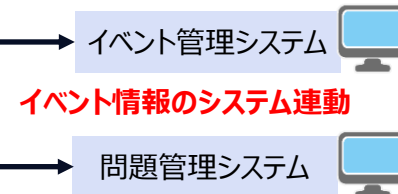
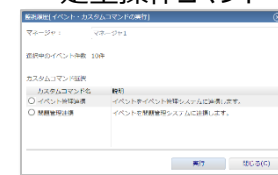
## 監視からの様々なアクション連携

監視結果に応じた様々なアクションの実行が可能



自動的な通知・アクション

ユーザ定義の  
定型操作コマンド



ユーザ判断によるシステム連携

各環境の管理簡易化

他システムへの連携運用も容易に実現可能

# 監視・性能 多彩な監視

26種類の監視項目を活用することで容易な作り込みを実現し  
SDML機能ではアプリケーションの自動監視が可能

## 作り込み不要な多彩な監視

監視設定は項目をGUIより入力するのみで簡易に設定可能

### エージェントレス監視（19種類）

- ・PING監視
- ・プロセス監視
- ・リソース監視
- ・サービス/ポート監視
- ・Windows サービス監視
- ・HTTP監視
- ・HTTPシナリオ監視
- ・SNMP監視
- ・SNMPTRAP監視
- ・SQL監視
- ・JMX監視
- ・システムログ監視
- ・ログ件数監視
- ・相関係数監視
- ・収集値統合監視
- ・カスタムトラップ監視
- ・クラウド課金監視
- ・クラウド課金詳細監視
- ・クラウドサービス監視

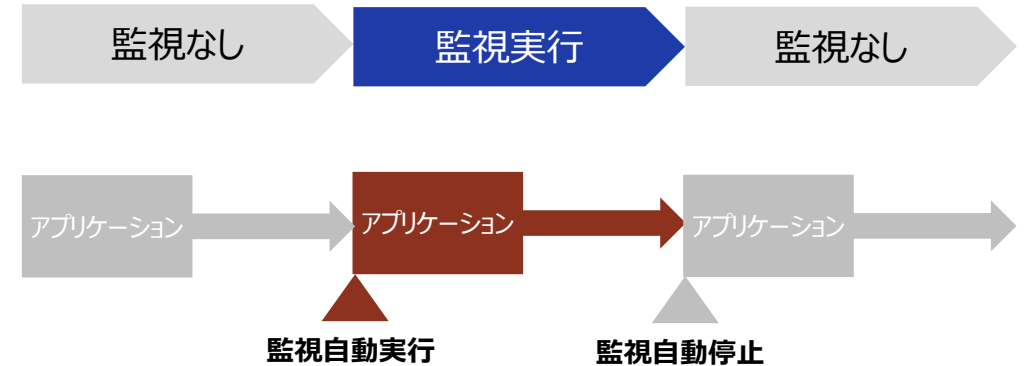
### エージェントが必要な監視（7種類）

- ・Hinemosエージェント監視
- ・Windowsイベント監視
- ・ログファイル監視
- ・バイナリファイル監視
- ・パケットキャプチャ監視
- ・カスタム監視
- ・クラウドログ監視

作り込みコストの削減

## アプリケーションの自動監視

ユーザによる監視の設計と設定が不要



- ① プロセスの死活状態
- ② プロセスの内部状態
- ③ プロセスのエラーログ

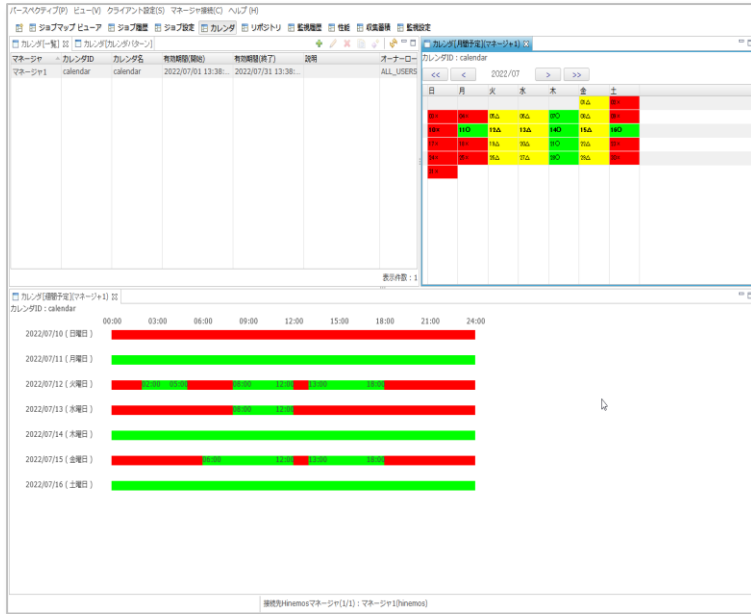
監視コストの削減

# 監視・性能 多彩な制御

高度なカレンダー設定やメッセージフィルタ機能によるルールベースのフィルタ設定が可能

## 高度な業務カレンダー連携

監視の細やかな実行制御を容易に自動化可能

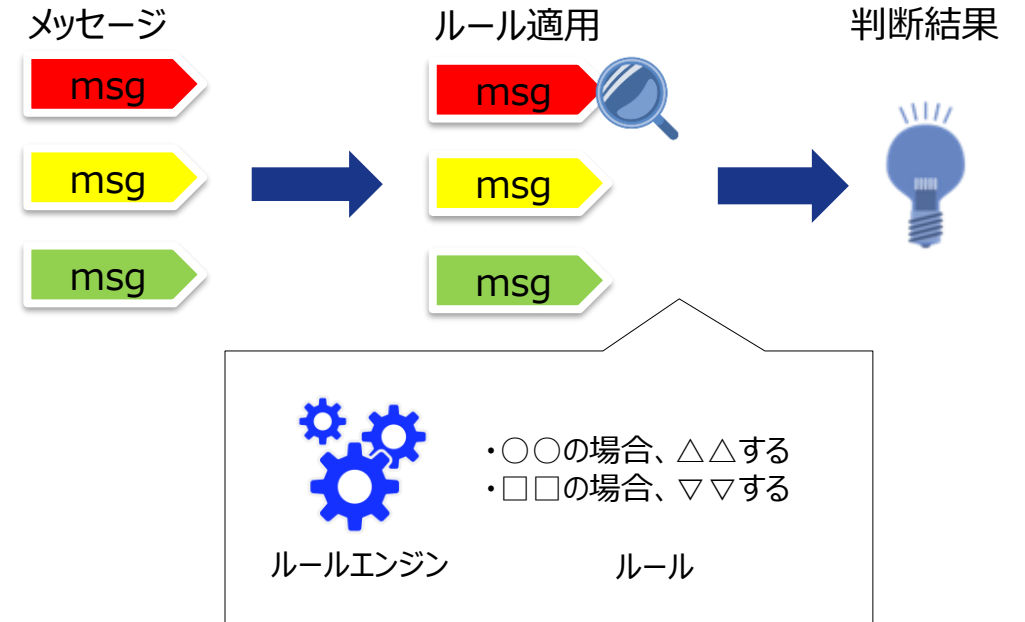


- 月の規則パターン指定
- 週の規則パターン指定
- 日の規則パターン指定
- 曜日の規則パターン指定
- 祝日等の不規則パターン指定
- 48時間カレンダー対応
- 特定曜日指定
- 月末月初指定
- 稼働日振替

様々な実行制御の要件に対応

## ルールベースのフィルタリング

不要なメッセージの抑制や関連メッセージの集約といった形で本質的なイベントにフィルタリングが可能



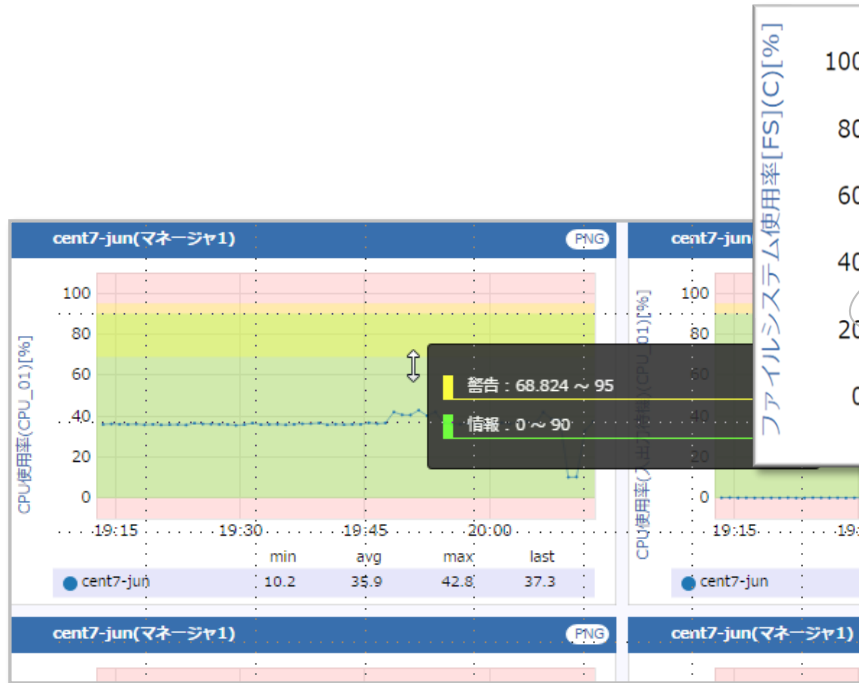
精度の高い判断結果

# 監視・性能 簡易なUIと将来予測

収集した各種データを分析するためのグラフ表示や、将来取り得る傾向(変化量)を予測、算出することが可能です。

## データの見える化・フィードバック

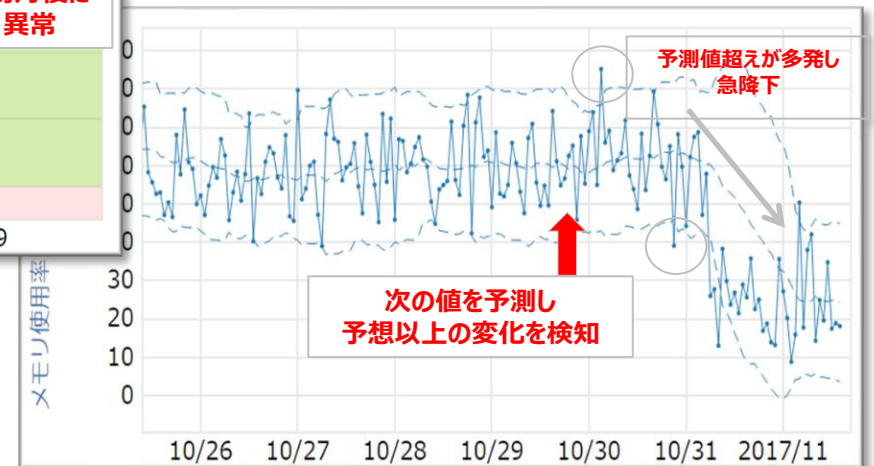
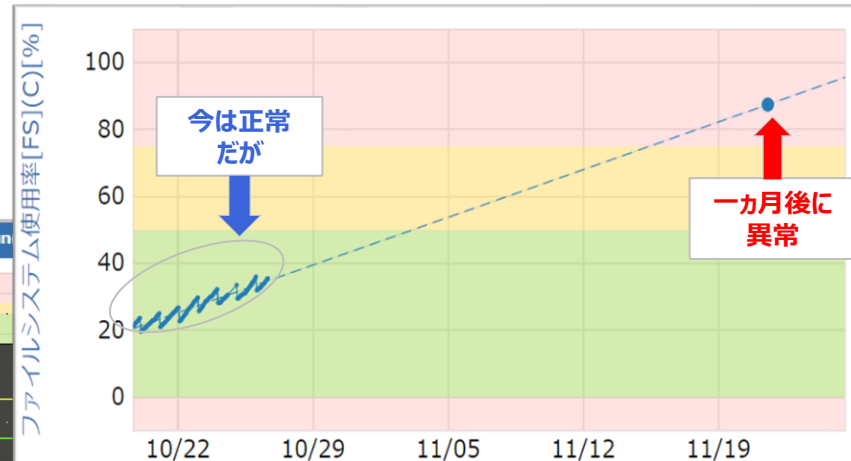
グラフ上でリソース監視の閾値確認・変更が可能



監視設定の適切な閾値を設定することが可能

## 将来予測・変化量の監視・見える化

将来予測の可視化を実現



人手を介する事なくいち早く検知することが可能

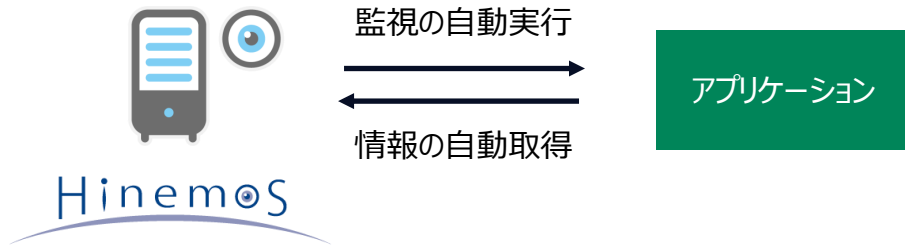
# Hinemosが提案する新たな運用 SDML

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



監視設計は不要。  
Hinemosが自動でアプリケーションの監視を実現

### ユーザ制御監視



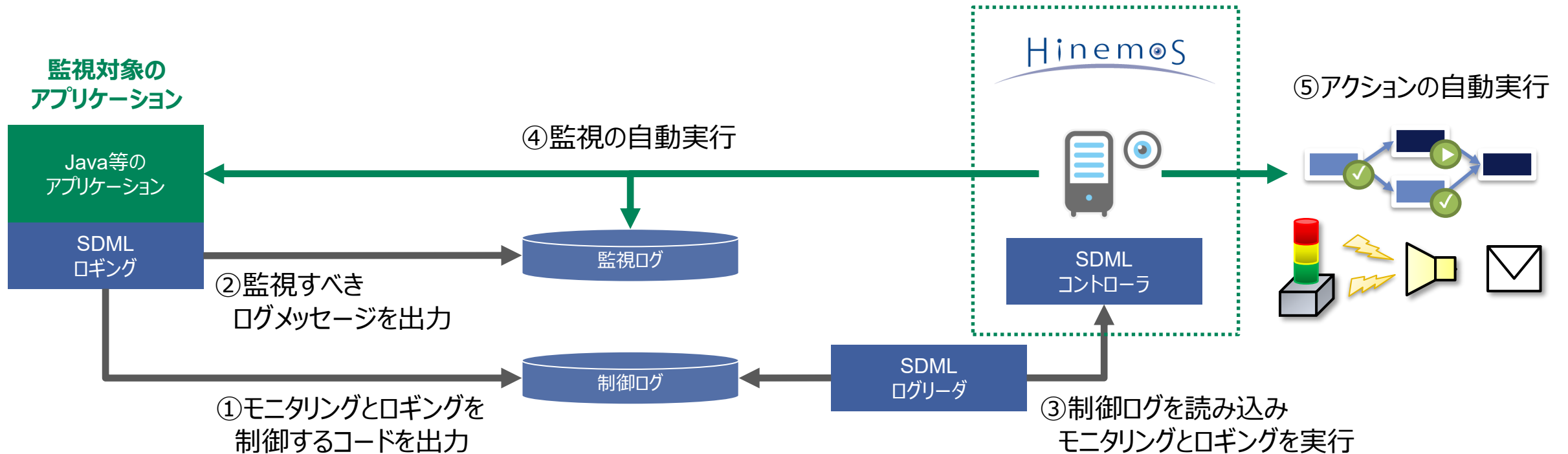
開発と同時に監視設計を組み込む。  
アプリケーションによるHinemosの制御を実現

特許出願中(特願2021-157931)

基盤だけではなくアプリケーションもソフトウェア制御の世界で、運用効率化・自動化を推進

# SDMLアーキテクチャ

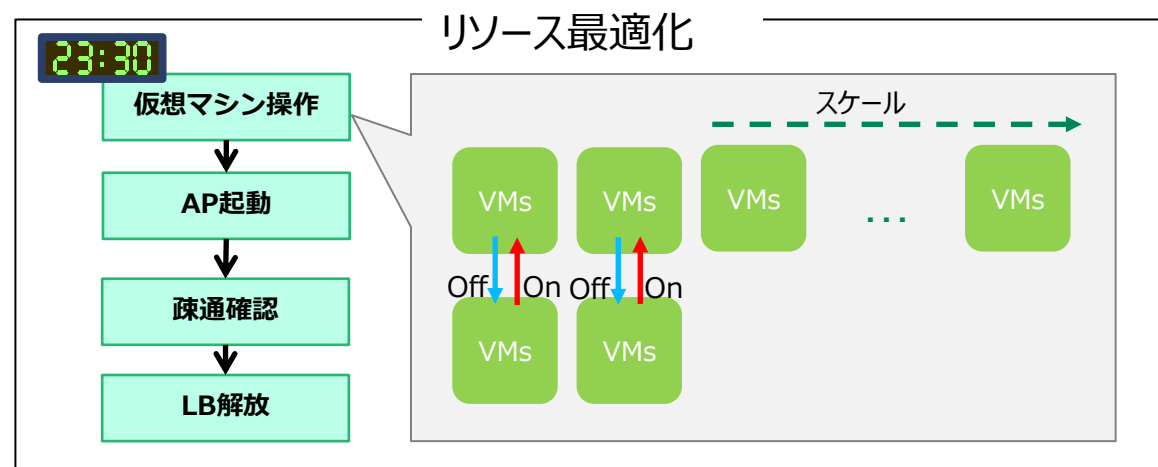
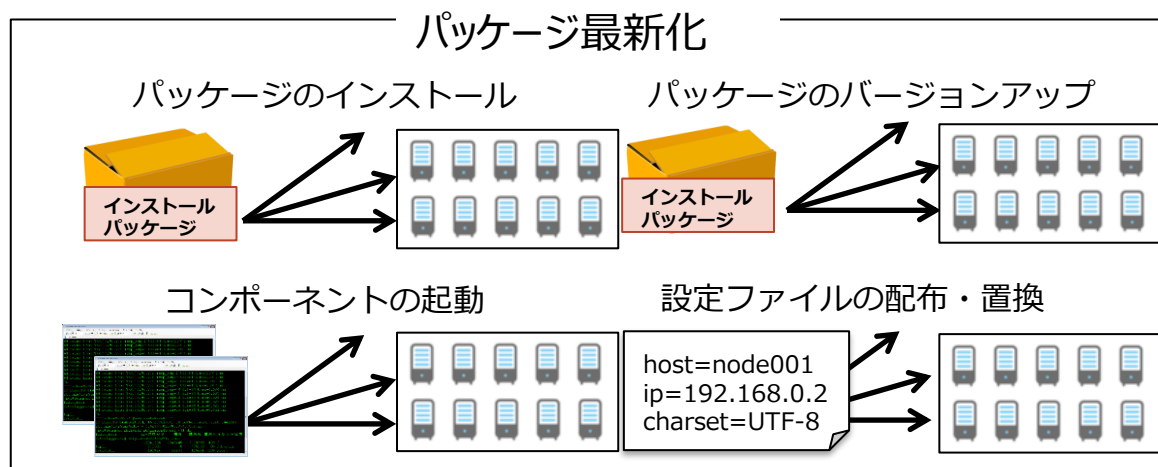
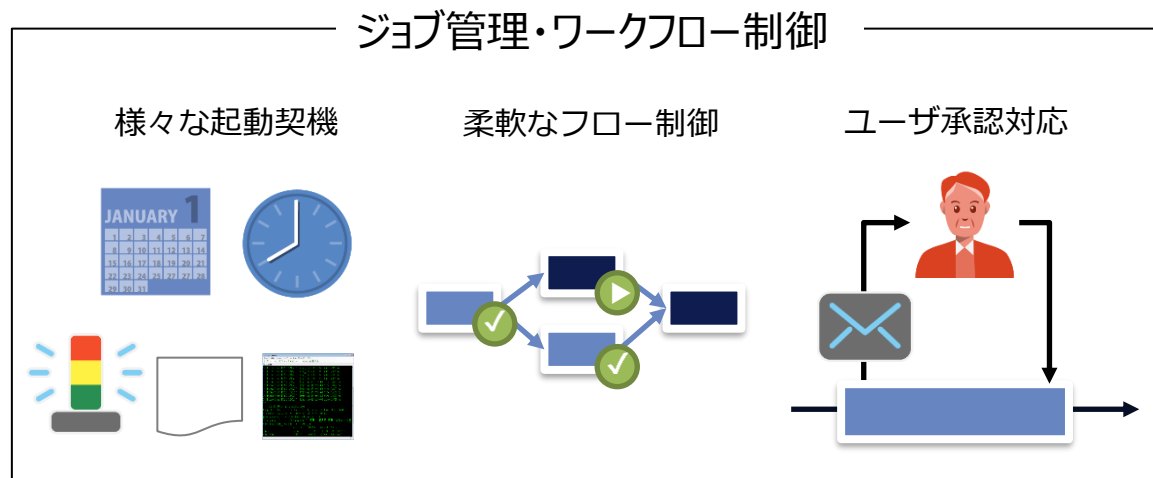
SDMLロギングをアプリケーションに組み込み、  
制御ログと監視ログを介在しモニタリング（監視）とロギングの定義と制御を実現



**自動監視**では、何もしなくてもアプリケーションの監視が可能になり、  
**ユーザ制御監視**では、アプリケーション開発に「監視制御」を含める事が可能になります

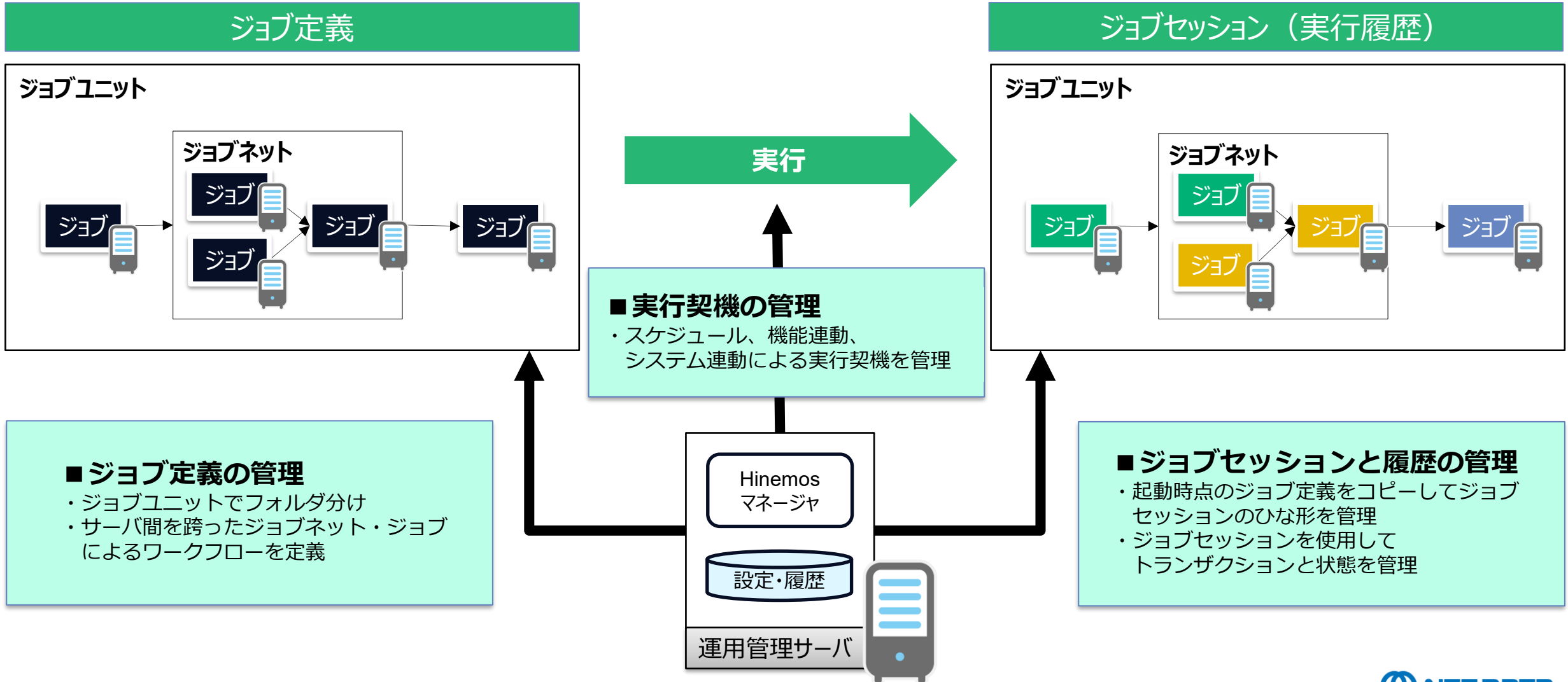
# 自動化

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



# 業務自動化（ジョブ管理）

ジョブ機能とはサーバを跨る処理フローを統合的に管理する機能



# 業務自動化 画面イメージ

## ジョブ履歴パースペクティブ

各ジョブの履歴を確認できる機能

パースペクティブ(P) ビュー(V) クライアント設定(S) マネージャ接続(C) ヘルプ(H)

ジョブマップビューア | ジョブ履歴 | 性能 | 収集番帳 | 監視設定 | 監視履歴

ジョブ履歴(一覧) | ジョブ履歴(同時実行制御) | ジョブ履歴(実行ジョブ連携メッセージ一覧)

マネージャ	実行状態	終了状態	終了値	セッションID	ジョブID	ジョブ名	ジョブユニット	種別	ファシリティID	スコープ	オーナーロールID	開始予定日時
マネージャ1	終了	正常	0	20220708083400-000	JOBUNIT	JOBUNIT	JOBUNIT	ジョブユニット		ALL_USERS	ALL_USERS	2022/07/08 08:15
マネージャ1	終了	正常	0	20220708080400-000	JOBUNIT	JOBUNIT	JOBUNIT	ジョブユニット		ALL_USERS	ALL_USERS	2022/07/08 08:04
マネージャ1	終了	正常	0	20220708074900-000	JOBUNIT	JOBUNIT	JOBUNIT	ジョブユニット		ALL_USERS	ALL_USERS	2022/07/08 07:49
マネージャ1	終了	正常	0	20220708073400-000	JOBUNIT	JOBUNIT	JOBUNIT	ジョブユニット		ALL_USERS	ALL_USERS	2022/07/08 07:34
マネージャ1	終了	正常	0	20220708071900-000	JOBUNIT	JOBUNIT	JOBUNIT	ジョブユニット		ALL_USERS	ALL_USERS	2022/07/08 07:19
マネージャ1	終了	正常	0	20220708070400-000	JOBUNIT	JOBUNIT	JOBUNIT	ジョブユニット		ALL_USERS	ALL_USERS	2022/07/08 07:04
マネージャ1	終了	正常	0	20220708064900-000	JOBUNIT	JOBUNIT	JOBUNIT	ジョブユニット		ALL_USERS	ALL_USERS	2022/07/08 06:49
マネージャ1	終了	正常	0	20220708063400-000	JOBUNIT	JOBUNIT	JOBUNIT	ジョブユニット		ALL_USERS	ALL_USERS	2022/07/08 06:34

ジョブ履歴(ジョブ詳細[マネージャ1]) | ジョブ履歴(同時実行制御状況)

セッションID: 20220708083400-000

実行状態	スキ...	終了状態	終了値	ジョブID	ジョブ名	ジョブユニット	種別	ファシリティID	スコープ	時刻	開始・再実行日時	終了・中断日時	実行時間	実行回
終了		正常	0	JOBUNIT	JOBUNIT	JOBUNIT	ジョブユニット				2022/07/08 08:34:...	2022/07/08 08:34:...	00:00:03	1
終了(条件未...		正常	0	CMD5	CMD5	JOBUNIT	コマンドジョブ	rhel86-hinemos7-f...	rhel86-hinemos7-f...		2022/07/08 08:34:...	2022/07/08 08:34:...	00:00:01	1
終了		正常	0	JOBNET1	JOBNET1	JOBUNIT	ジョブネット				2022/07/08 08:34:...	2022/07/08 08:34:...	00:00:01	1
終了		正常	0	CMD3	CMD3	JOBUNIT	コマンドジョブ	rhel86-hinemos7-f...	rhel86-hinemos7-f...		2022/07/08 08:34:...	2022/07/08 08:34:...	00:00:01	1
終了		正常	0	CMD4	CMD4	JOBUNIT	コマンドジョブ	rhel86-hinemos7-f...	rhel86-hinemos7-f...		2022/07/08 08:34:...	2022/07/08 08:34:...	00:00:01	1
終了		正常	0	JOBNET2	JOBNET2	JOBUNIT	ジョブネット				2022/07/08 08:34:...	2022/07/08 08:34:...	00:00:02	1
終了		正常	0	CMD1	CMD1	JOBUNIT	コマンドジョブ	rhel86-hinemos7-f...	rhel86-hinemos7-f...		2022/07/08 08:34:...	2022/07/08 08:34:...	00:00:01	1
終了		正常	0	CMD2	CMD2	JOBUNIT	コマンドジョブ	rhel86-hinemos7-f...	rhel86-hinemos7-f...		2022/07/08 08:34:...	2022/07/08 08:34:...	00:00:01	1

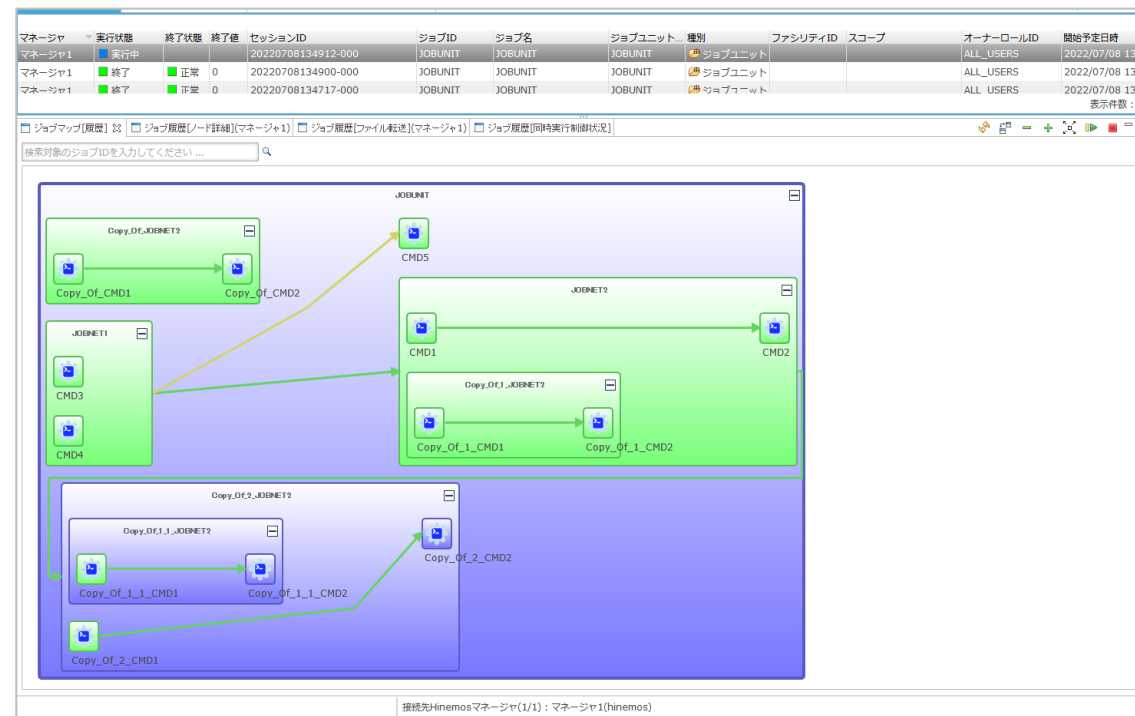
ジョブ履歴(リード詳細[マネージャ1]) | ジョブ履歴(ファイル転送[マネージャ1])

セッションID: 20220708083400-000, ジョブID: CMD3

実行状態	戻り値	ファシリティID	ファシリティ名	開始・再実行日時	終了・中断日時	実行時間	メッセージ
終了	0	rhel86-hinemos7-f...	rhel86-hinemos7-f...	2022/07/08 08:34:...	2022/07/08 08:34:...	00:00:00	[2022/07/08 08:34:01] stdout~hogehoge, stderr=[2022/07/08 08:34:01] コマンド終了待ち[2022/07/08 08:34:01]

## ジョブマップパースペクティブ

各ジョブをマップ形式で確認できる機能

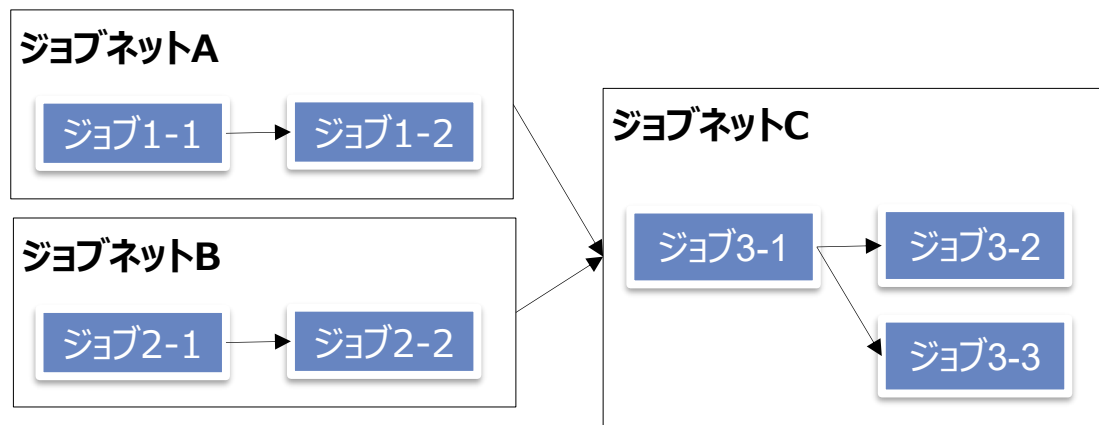


# 業務自動化 視覚的に分かりやすい業務フロー定義

階層的なグループ化や、ジョブの実行対象としてノードをグループ化したスコープの指定が可能

## 階層的な業務フロー定義

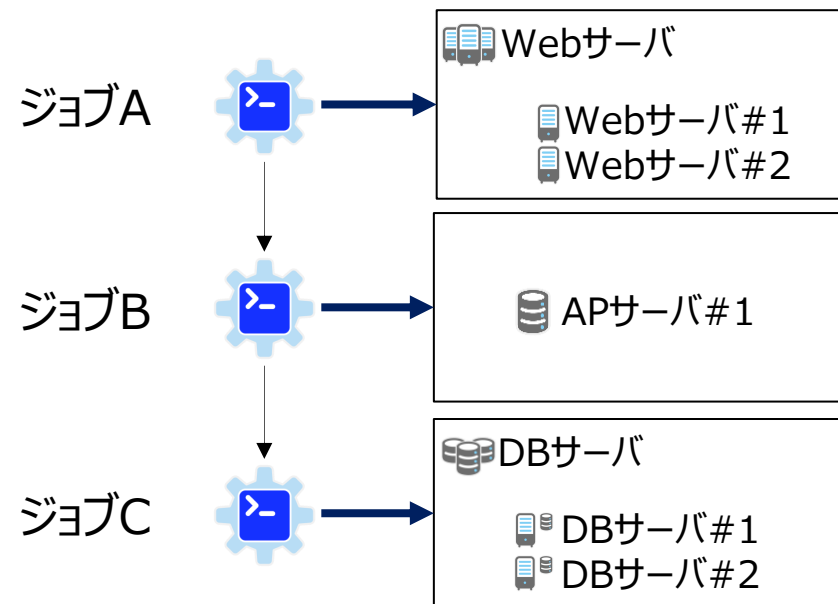
処理フローは「ジョブネット」として階層的に**グループ化**



様々な通知などをGUIから簡易に指定が可能

## スコープにより定義集約

グループ化により**複数サーバで同一処理**が可能



個々を意識せず論理的なグループに対しジョブ設定可能

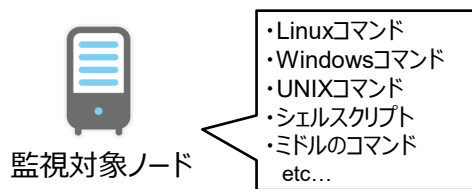
# 業務自動化 多彩なジョブの種類(1/2)

Hinemosが提供する機能により、豊富な実行処理を提供

## コマンドジョブ

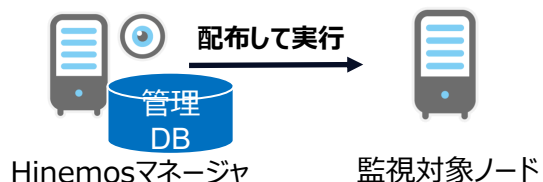
### ■コマンド・スクリプト実行

OS上で実行可能なコマンドをそのまま実行  
ジョブ毎に環境件数の指定も可能



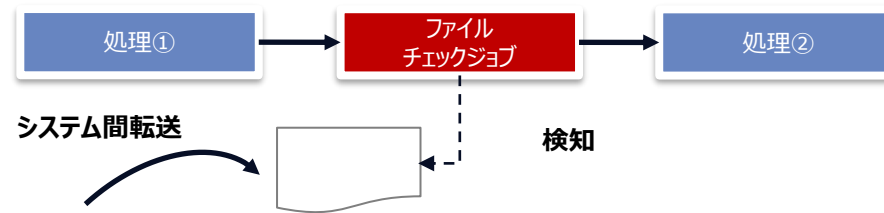
### ■スクリプト配布

Hinemosマネージャにスクリプトを登録し、  
起動時に配布して実行することが可能



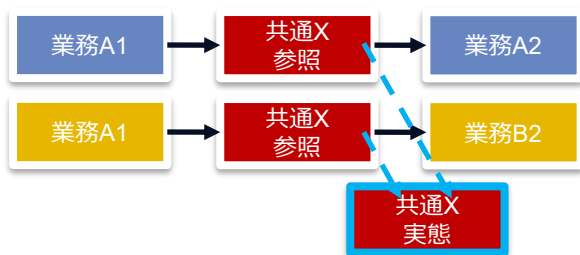
## ファイルチェックジョブ

ファイルの存在確認・作成・削除・変更（サイズ、タイムスタンプ）の  
契機を待ち受け、後続ジョブを実行可能



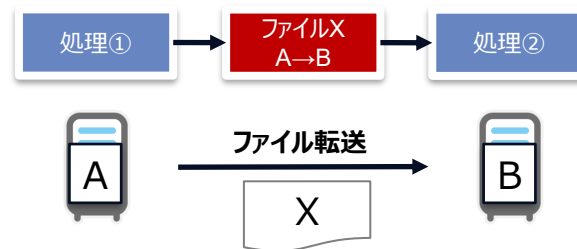
## 参照ジョブ

ジョブの参照を使用することで共通的な処理を1か所に集約が可能



## ファイル転送ジョブ

1つのジョブでファイルの転送元と転送先を指定して転送を実行



基本的な処理についても、効率的に実行が可能

# 業務自動化 多彩なジョブの種類(2/2)

Hinemosが提供する機能により、豊富な実行処理を提供

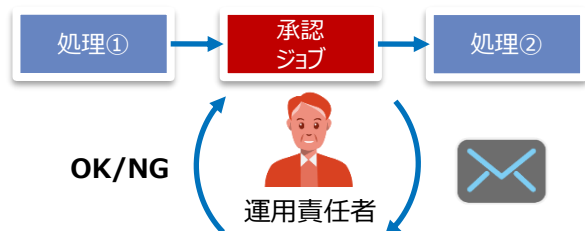
## リソース制御ジョブ

仮想マシン・ストレージの操作（起動・停止など）が実行可能



## 承認ジョブ

ジョブフロー内にユーザ判断（承認）を含めることが可能



## 監視ジョブ

Hinemosの監視機能をジョブとして指定が可能



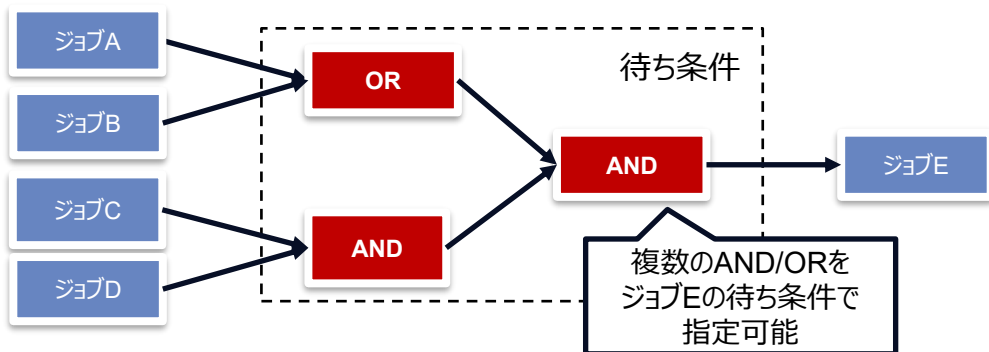
特殊な処理についても、作り込み不要で設定だけで利用可能

# 業務自動化 多様なジョブ実行の制御(1/2)

Hinemosが提供する機能により、豊富な実行制御に対応

## 待ち条件

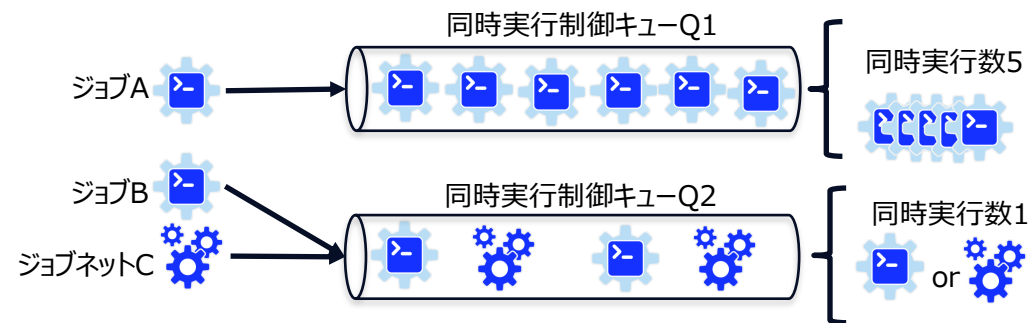
待ち条件は後続ジョブ側に指定



## 同時実行制御

同時実行制御キューによる複数ジョブ・ジョブネットを跨がる

同時実行制御が可能



## 遅延監視

終了遅延監視と、先行ジョブの遅延を監視

### 開始遅延監視

ジョブセッションの開始後の時間

時刻

### 終了遅延監視

ジョブセッションの開始後の時間

ジョブ開始後の時間

時刻

実行履歴からの変化量(×標準偏差)

## 繰り返し実行

様々な繰り返し実行の指定が可能

### ◆ 単純な繰り返し



### ◆ 異常終了時のリトライ



複雑な業務要件についても、作り込み不要で設定するだけで制御可能

# 業務自動化 多様なジョブ実行の制御(2/2)

Hinemosが提供する機能により、豊富な実行制御に対応

様々なジョブの実行制御		
ジョブのアクセス制御	先行ジョブとの情報連携	待ち条件のAND/OR対応
ジョブの編集モード	ジョブ変数の利用	Hinemosエージェント異常時の状態遷移
ジョブ実行優先度	ジョブ変数にノード変数の指定	戻り値の飛び番対応
エージェント単位の多重度実行制御	コマンドの標準出力/エラー出力の通知	開始遅延/終了遅延監視
キューによるジョブ同時実行制御	コマンドのノードプロパティ参照	実行時間予測による終了遅延検知
日跨ぎ対応（48時間カレンダー）	スクリプト配布機能	ジョブ終了時のシグナル送信
運用時刻の設定	OS環境変数の指定	標準出力のファイル出力
スケジュールの繰り返し実行	セッションを跨いだ連携	ジョブの実行時間表示
ジョブの繰り返し実行	マネージャを跨いだ連携	ジョブ履歴表示フィルタ条件の複数指定
ジョブのテスト実行	待ち条件のIf/Else対応	ジョブ履歴表示フィルタ条件の保存

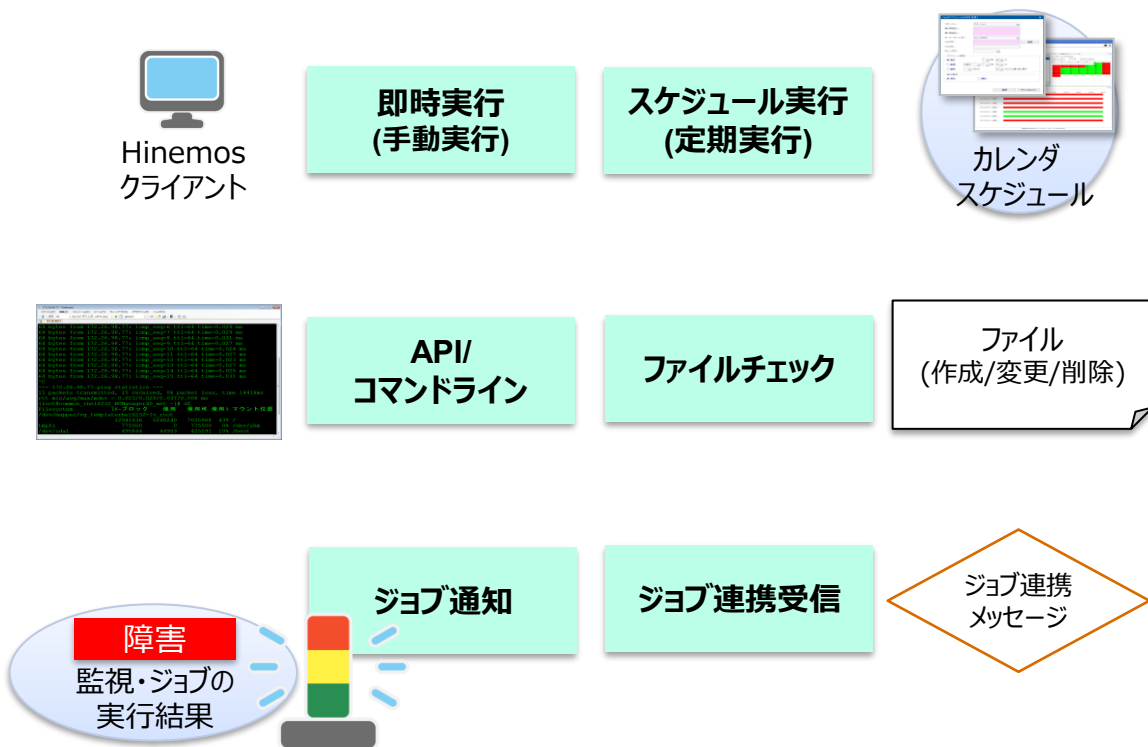
様々なパターンに対応できる

# 業務自動化 様々な実行契機と実行予定の管理

豊富なジョブネットの実行と実行予定の管理が可能

## 様々な実行契機

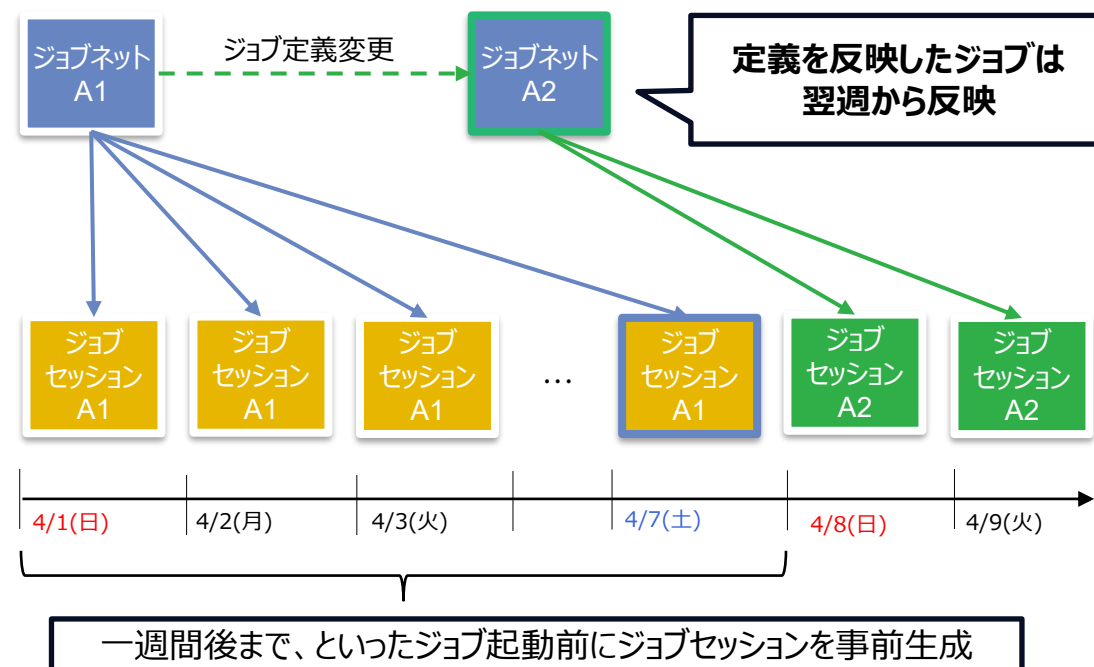
監視の細やかな実行制御を容易に自動化可能



様々な実行契機要望に対応

## 実行予定の管理

ジョブ起動前にジョブセッションの事前生成が可能



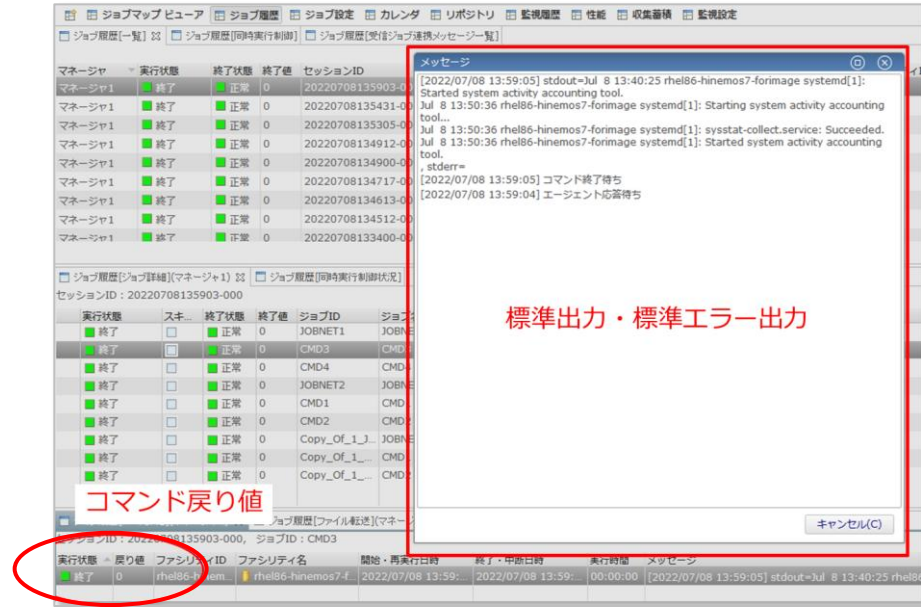
事前にスケジューリングが可能

# 業務自動化 実行結果の集約とセッション・マネージャ連携

Hinemosの画面より、コマンドの標準出力・標準エラーや戻り値を確認、  
ジョブセッションやマネージャ間を跨るジョブ実行制御が可能

## 実行結果の集約

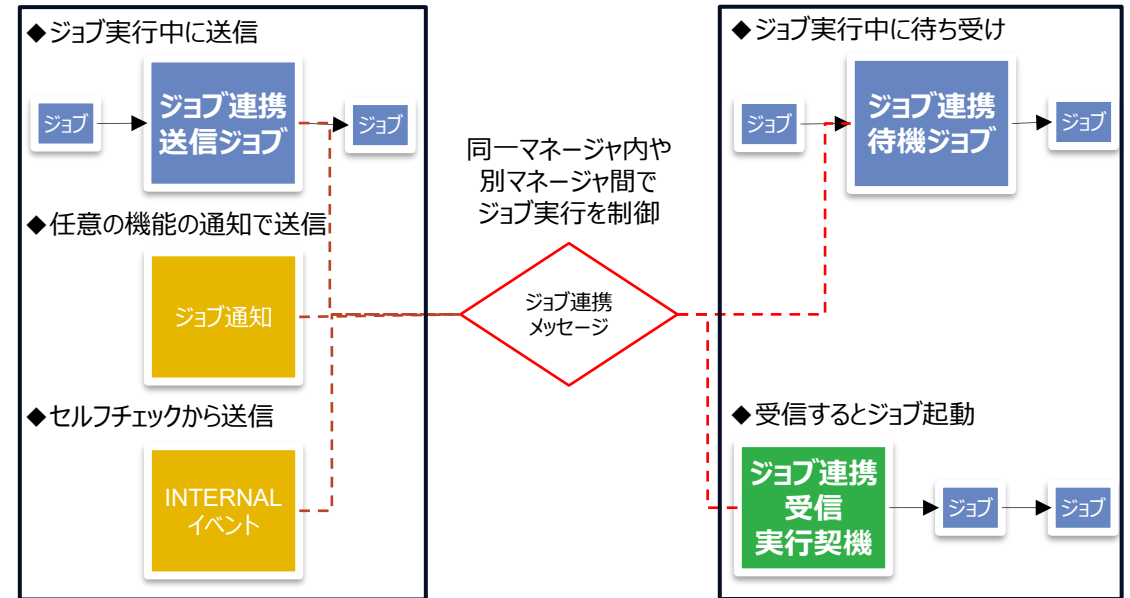
コマンドの戻り値、標準出力・標準エラー出力の  
確認が可能



Hinemosの画面から、実行結果の確認が可能

## セッション・マネージャ連携

ジョブセッションやマネージャ間を跨る  
ジョブ実行制御が可能



複雑な業務フロー間の連携設定をサポート

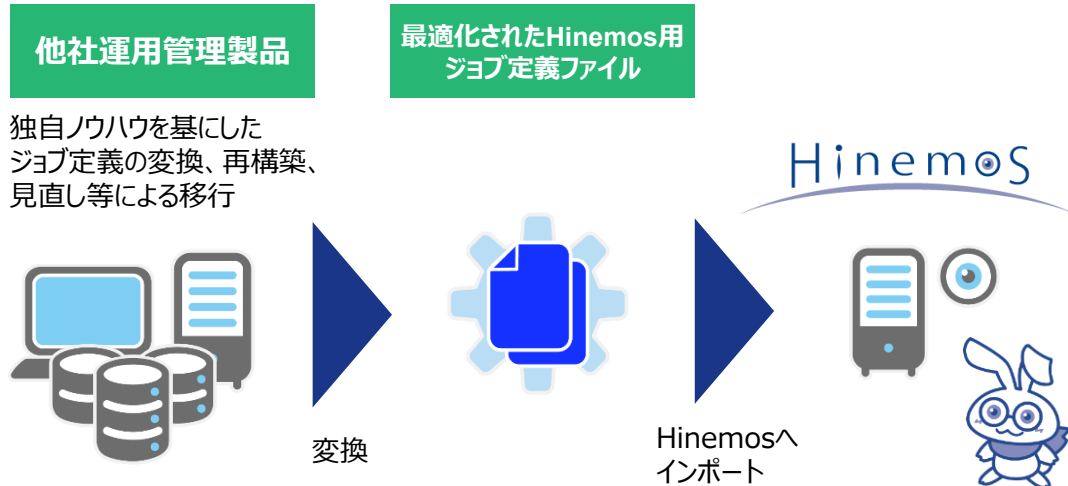
# 業務自動化 Hinemosへの移行と様々なサービス連携

他のジョブ管理製品からの移行やServieNowなどのSaaSやミドルウェアと連携をサポート

## 他製品からの移行性

Hinemos Migration Assistant  
(他製品からの移行支援ツール)  
を用いることで簡易に移行が可能

Hinemosのジョブ定義に変換する移行サービスも提供

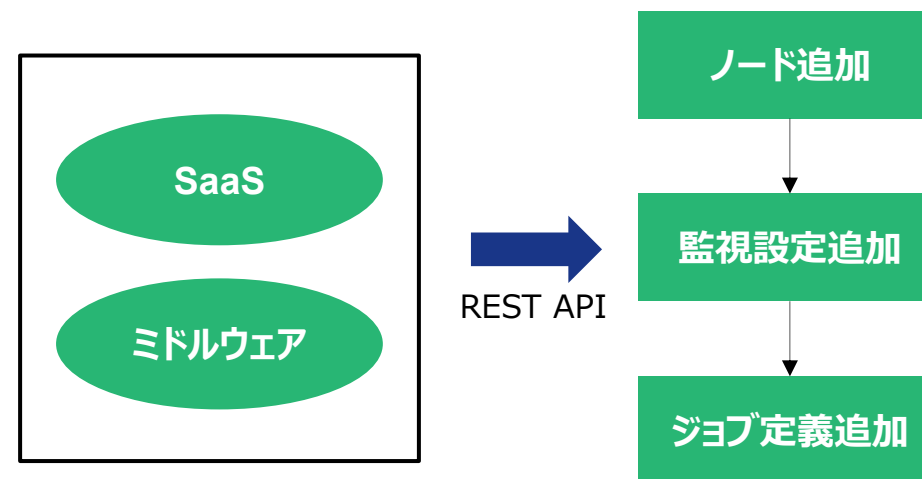


移行作業の簡易化を実現

## サービス連携

Hinemosを介した幅広い操作を自動化

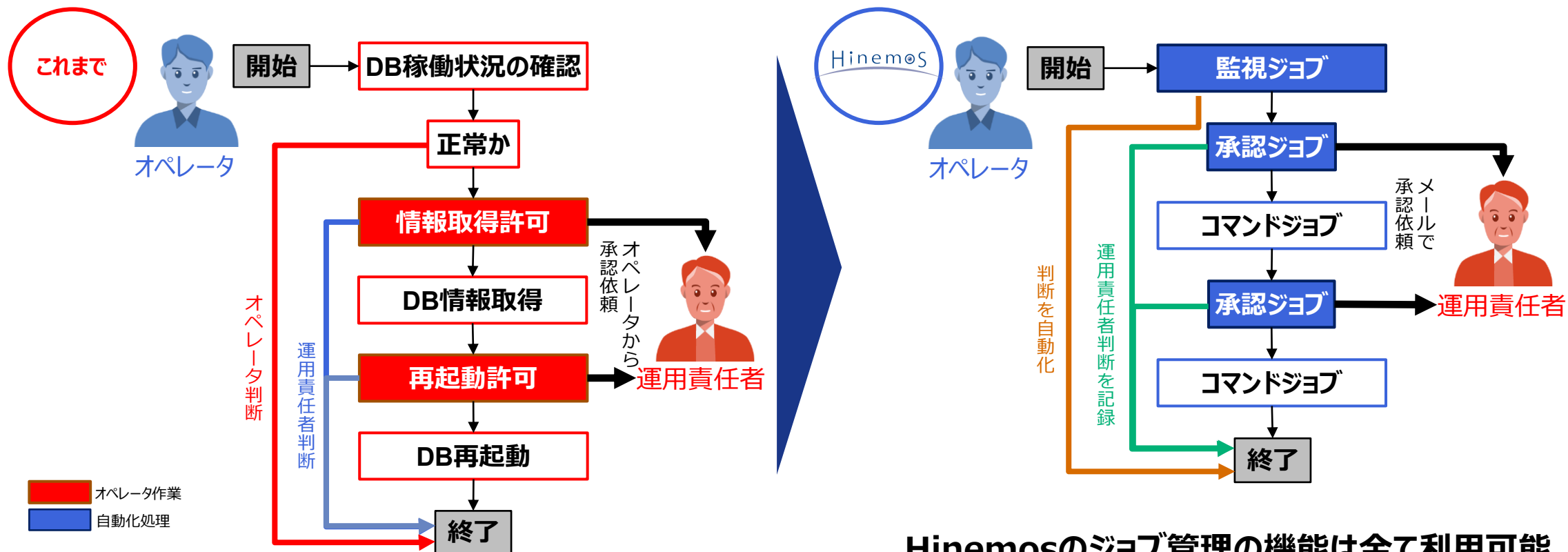
Hinemos設定の自動制御例



自動制御による操作性の向上

# 運用自動化 (Runbook Automation)

システム運用に必要な一連のプロセスを、定型化、自動化することが可能  
オペレーションの実行結果はすべてHinemos上で履歴管理される

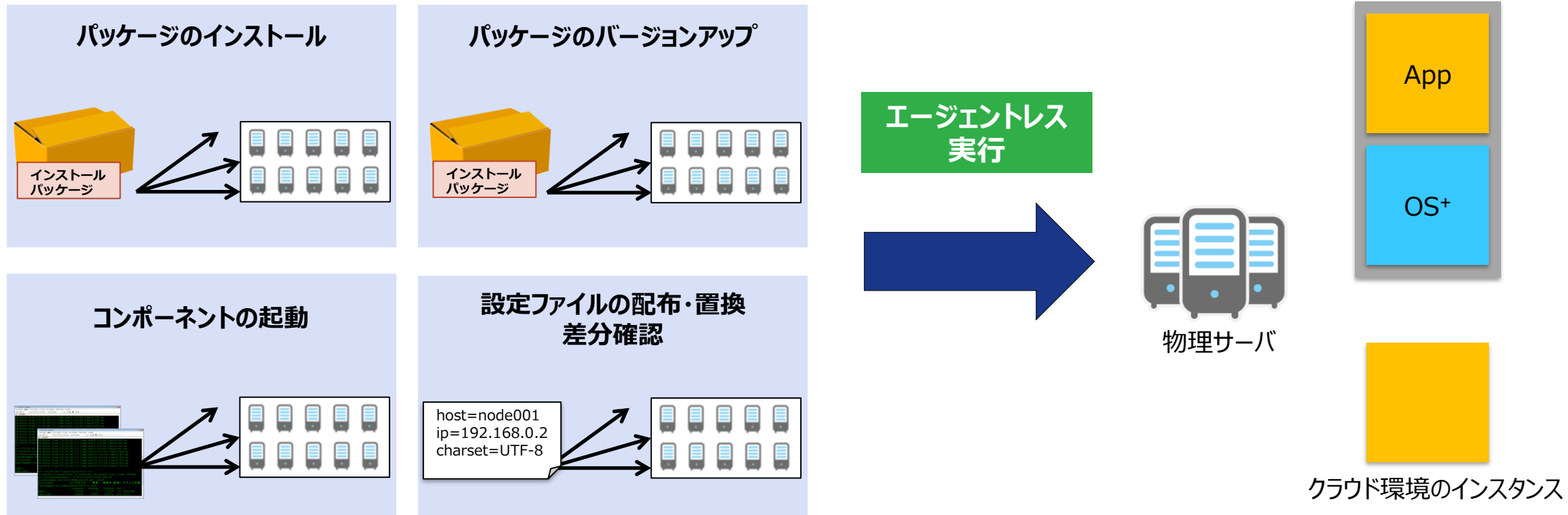


Hinemosのジョブ管理の機能は全て利用可能

自動化によるコスト削減を実現

# 構築自動化（環境構築）

エージェントレスでOS上の環境構築に必要な一連の作業を定型化



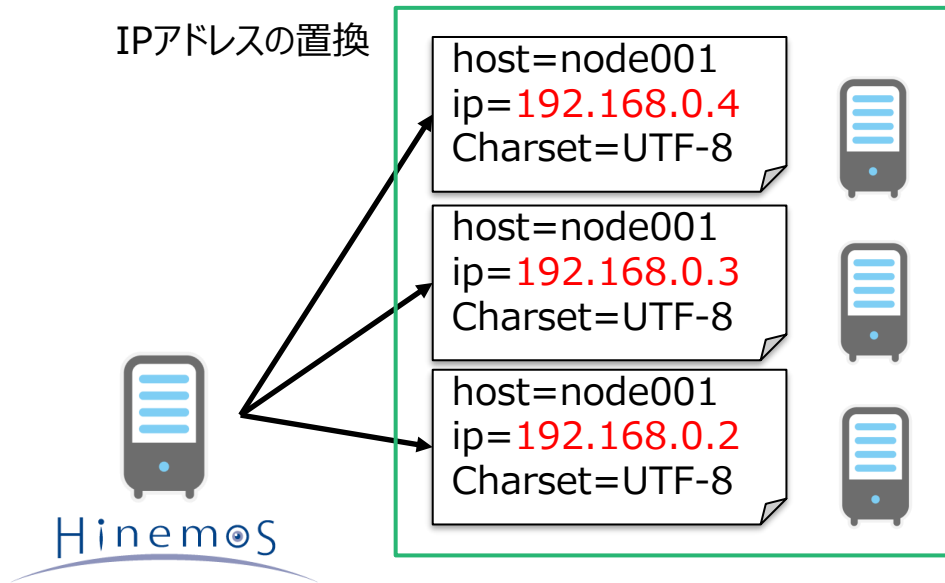
頻繁なOS初期セットアップや定期的なバージョンアップ作業等を効率的に実現

# 構築自動化 細やかな制御

ファイル配布やファイル差分確認をサポート

## 置換配布

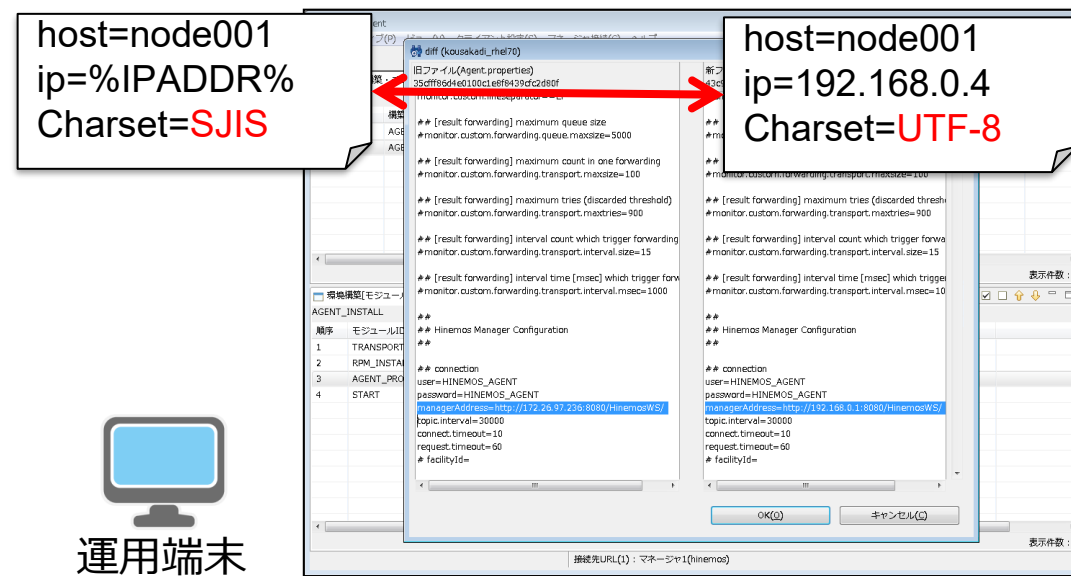
ファイル配布時にリポジトリ情報を利用した制御が可能



ファイル配布の効率化

## 差分確認

ファイル配布時に既存ファイルとの差分を確認

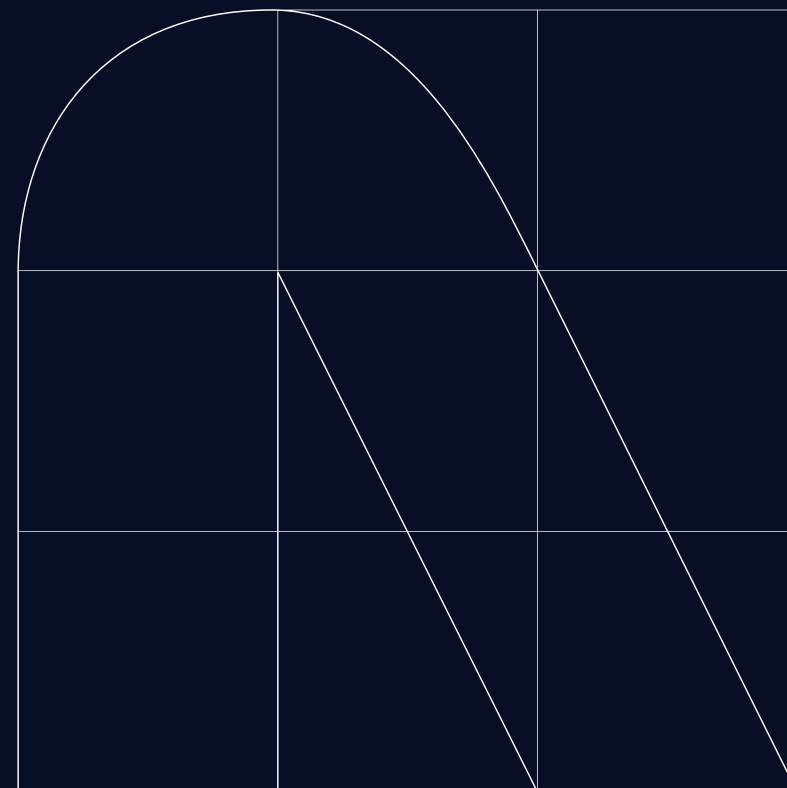


運用端末

差分比較を簡易化

# 03

## Hinemosエンタープライズ機能



# システム運用管理における課題

エンタープライズシステム運用管理の現場では、大規模・複雑化の影響により様々な課題が発生している

## 属人化

- システムの複雑化により運用が不明瞭になっている。  
→ 見える化して運用を明確にしたい。
- 特定の人しか作業ができない。  
→ 誰でも作業できるよう見える化／自動化したい。

## コスト／ 運用作業負荷の増大

- 煩雑なシステムの運用に大量の人員を動員している。  
→ システムを明確化して動員数を減らしたい。
- 同一の作業を手作業で実施していることにより時間がかかり、かつ人的ミスが発生することがある。  
→ 自動化や効率化でコスト削減、人的ミスを減らしたい。
- 作業状況／結果の確認・把握することが難しく、手間がかかる。  
→ 作業状況／結果をレポートとして出したい。

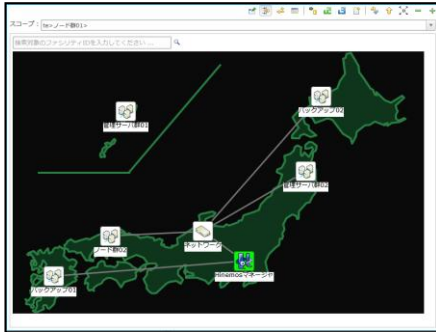


# エンタープライズ機能

エンタープライズ運用をサポートする様々な機能を提供

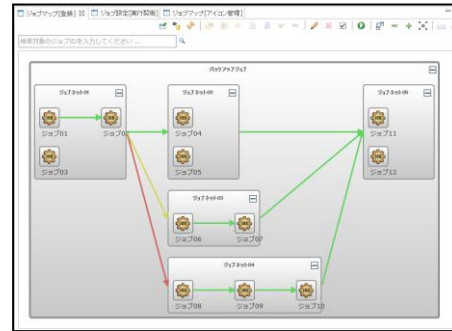
## Hinemosノードマップ

監視対象を2次元マップで可視化



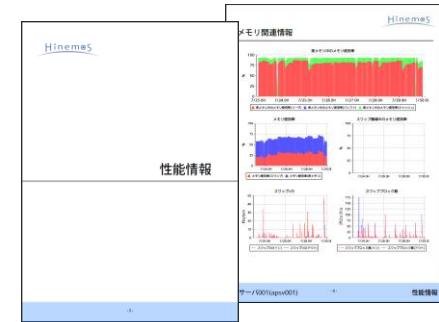
## Hinemosジョブマップ

ジョブフローを可視化し、運用操作を容易化



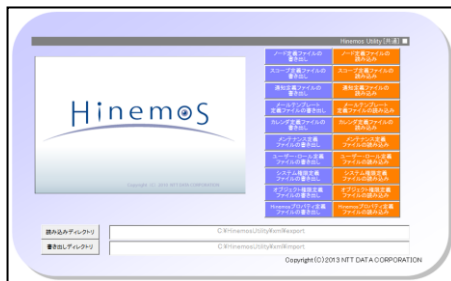
## Hinemosレポート

監視結果やジョブ実行状況をレポート出力



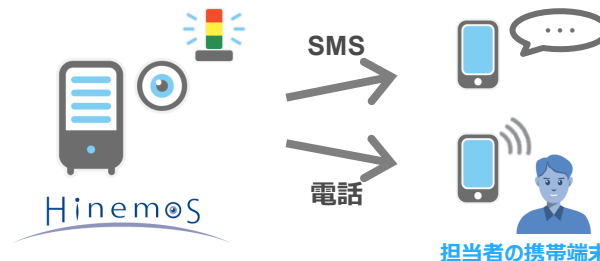
## Hinemos Utility

ExcelでHinemosの設定操作、一括入出力を実現



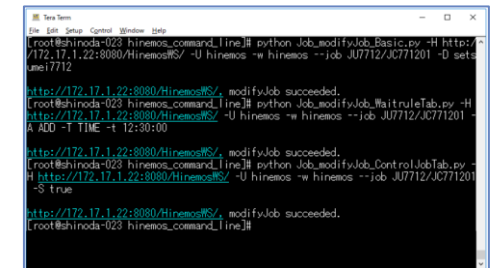
## 電話通知

電話/SMSによる自動通知で  
オペレータ対応工数削減



## ユーティリティツール

Hinemosをより活用できる  
便利ツール類を提供



# システム運用管理における課題への対応

課題に対応したエンタープライズ機能

属  
人  
化  
対  
策

■ 見える化して運用を明確にしたい。  
→ **Hinemosジョブマップ、Hinemosノードマップ**

■ 誰でも作業できるよう見える化／自動化したい。  
→ **Hinemosジョブマップ、ユーティリティツール**

コ  
ス  
ト  
／  
運  
用  
作  
業  
負  
荷  
の  
増  
加  
対  
策

■ システムを明確化して動員数を減らしたい。  
→ **Hinemosジョブマップ、Hinemosノードマップ**

■ 自動化や効率化でコスト削減、人的ミスを減らしたい。  
→ **Hinemosジョブマップ、Hinemosノードマップ、Hinemos Utility、ユーティリティツール、電話通知**

■ 作業状況／結果をレポートとして出したい。  
→ **Hinemosレポートイング**

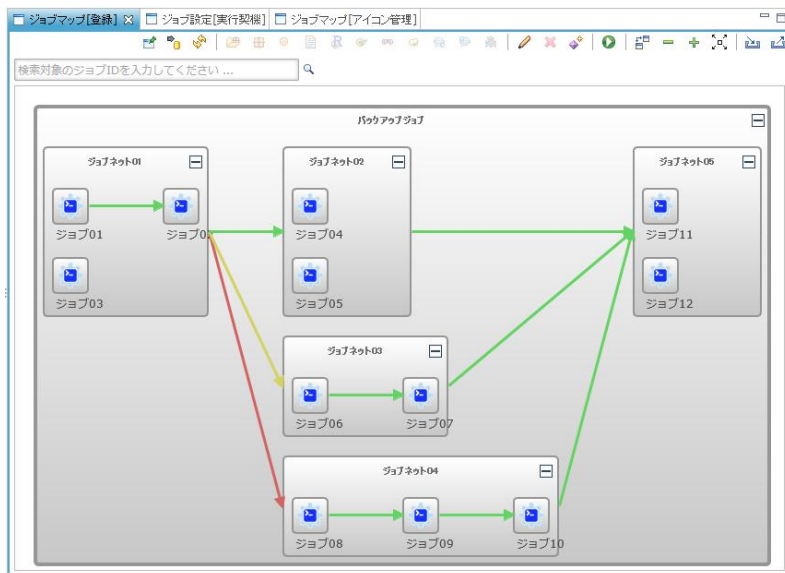
Hinemosサブスクリプション  
ユーザであれば  
全ての機能を利用可能



# ジョブマップ概要

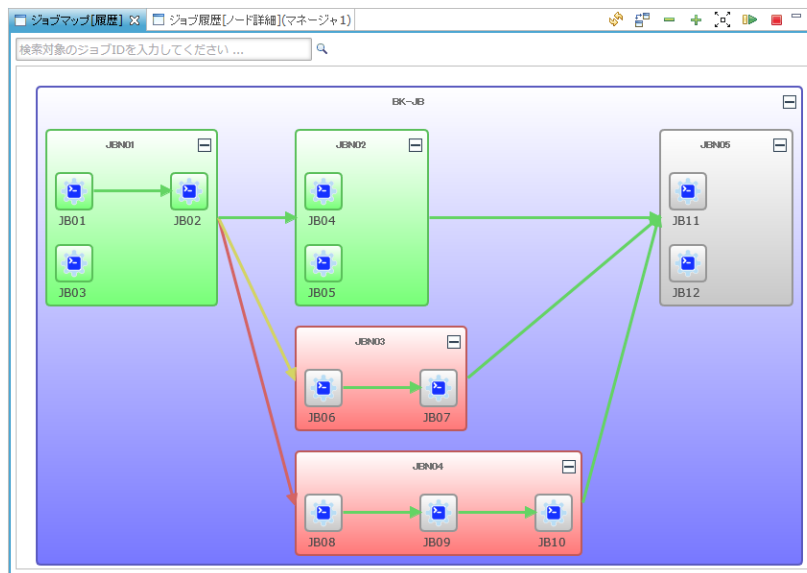
ジョブフローの設計構築から運用の把握まで容易に行える

## ジョブマップエディタ



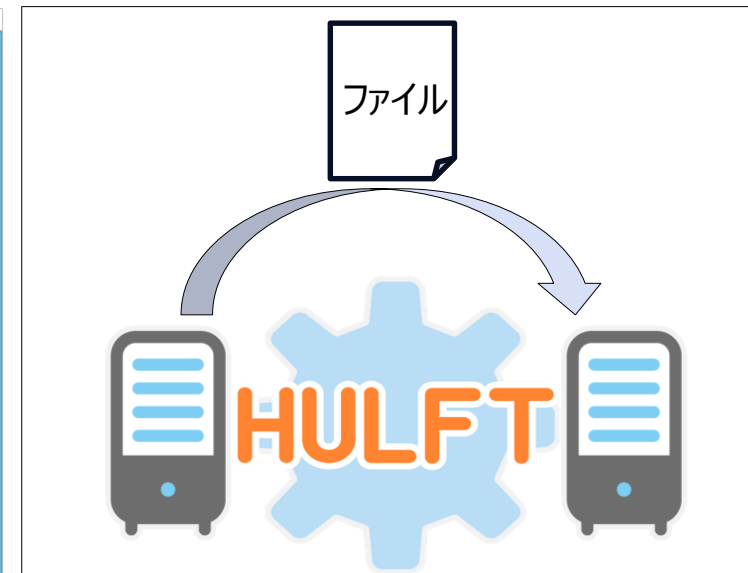
直感的な操作で容易にジョブ構築

## ジョブマップビューア



グラフィカルなジョブ遷移で状況を一括把握

## HULFT連携機能



ファイル転送ツールHULFTを使った  
ファイル転送ジョブの作成が可能

# ノードマップ概要

イベントやスコープの概念を2次元マップ上に拡張し、監視対象のインシデントの見える化を実現

## システムマップ

直感的操作でシステムマップを作成。

## ネットワークポロジ可視化

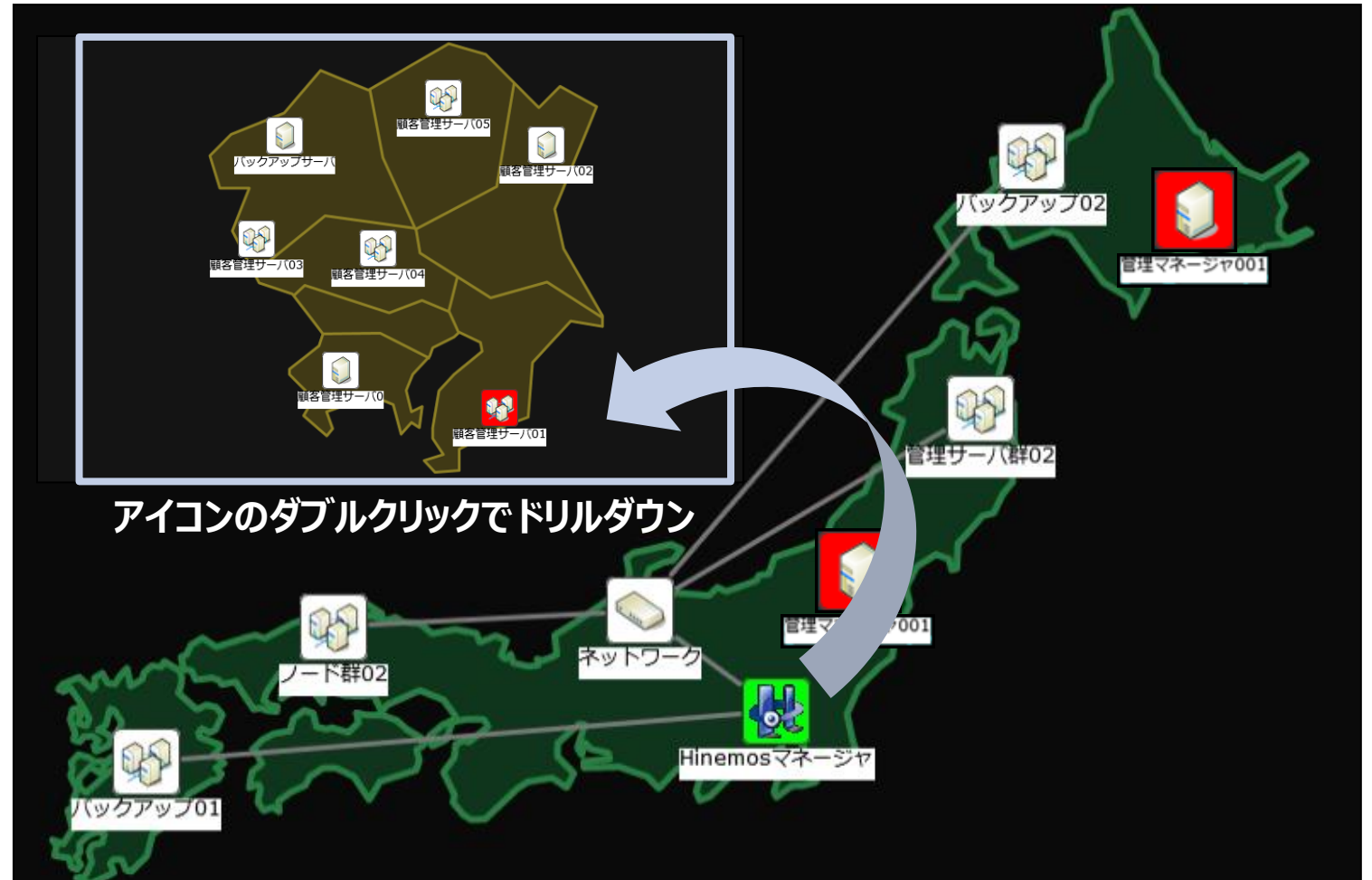
ネットワーク内の機器や接続状態をシステムマップ上に視覚的に表示。

## システムの状況可視化

機器の状態はアイコンの背景色として表示。

## ドリルダウン・死活確認

アイコンをダブルクリックして下位のスコープへドリルダウン。

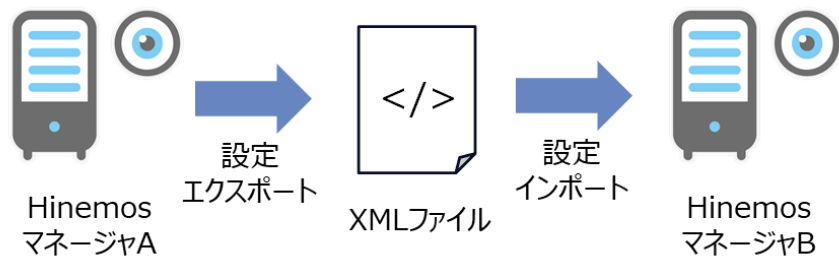


# Hinemos Utility概要

Hinemos Utilityでは以下の機能が提供される

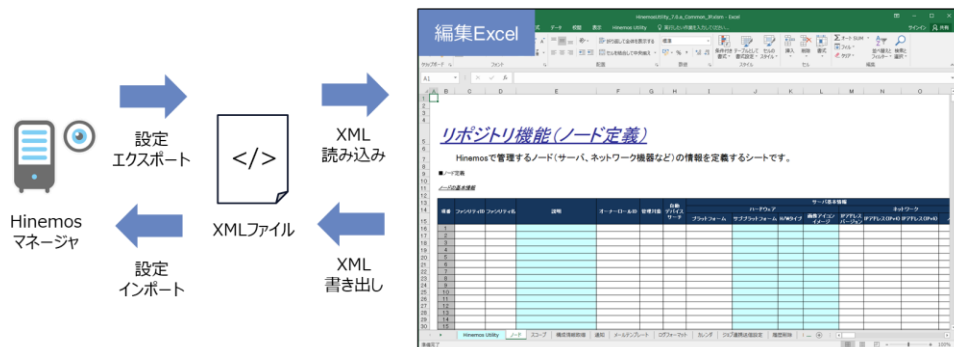
## 設定インポートエクスポート

Hinemosマネージャの設定をXMLファイルとしてエクスポート、またはXMLからインポート



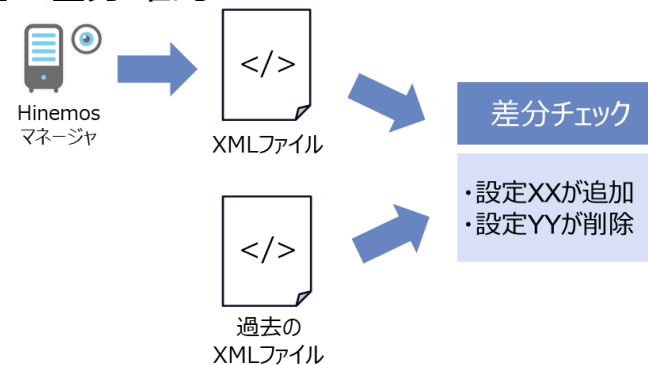
## 編集Excelによる設定の編集

Excelのインターフェイスを使用した、効率的なHinemosマネージャの設定編集



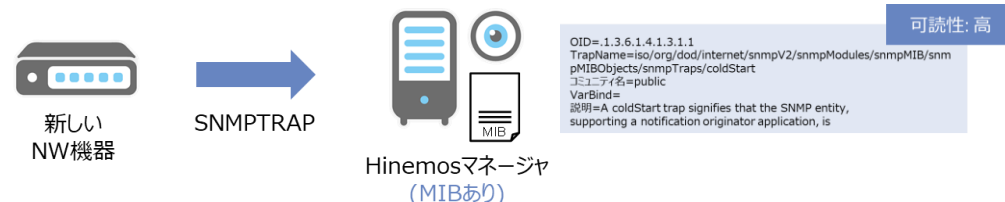
## エクスポートした設定の差分チェック

HinemosマネージャからエクスポートしたXMLファイル同士の比較、設定の差分確認



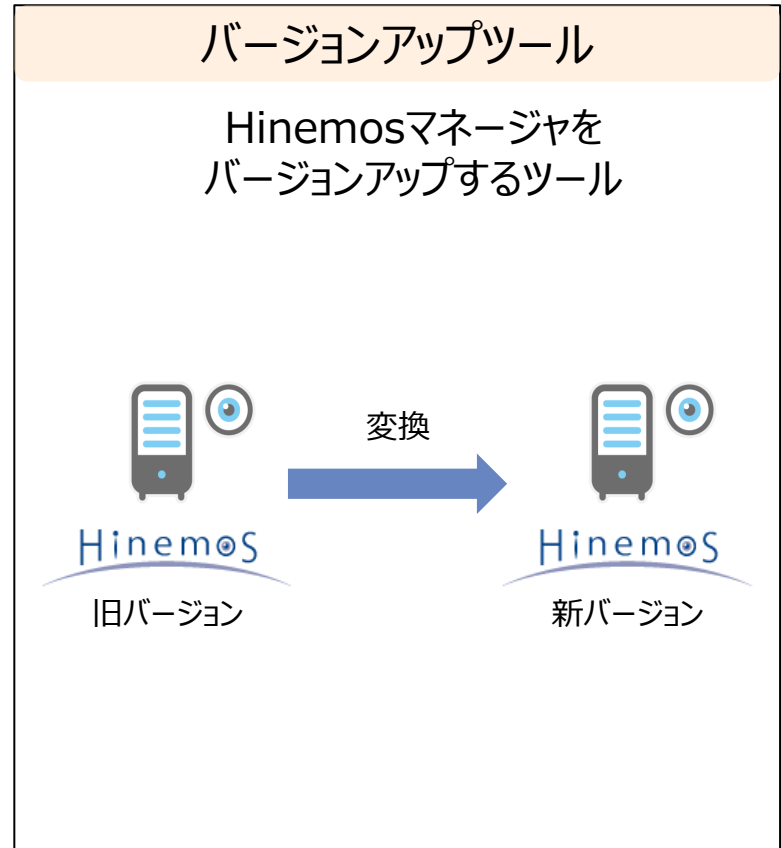
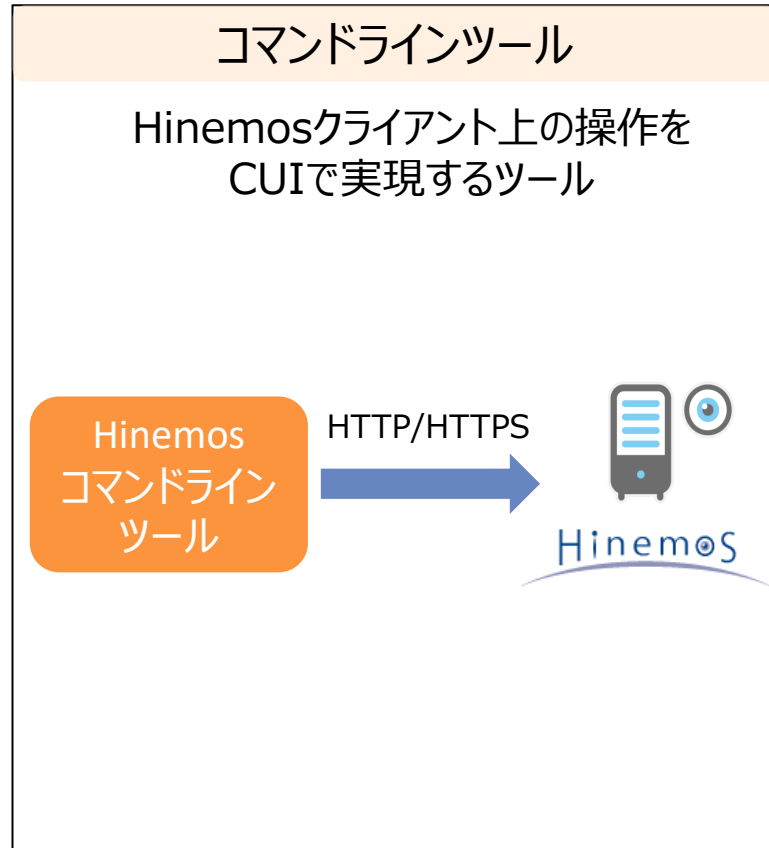
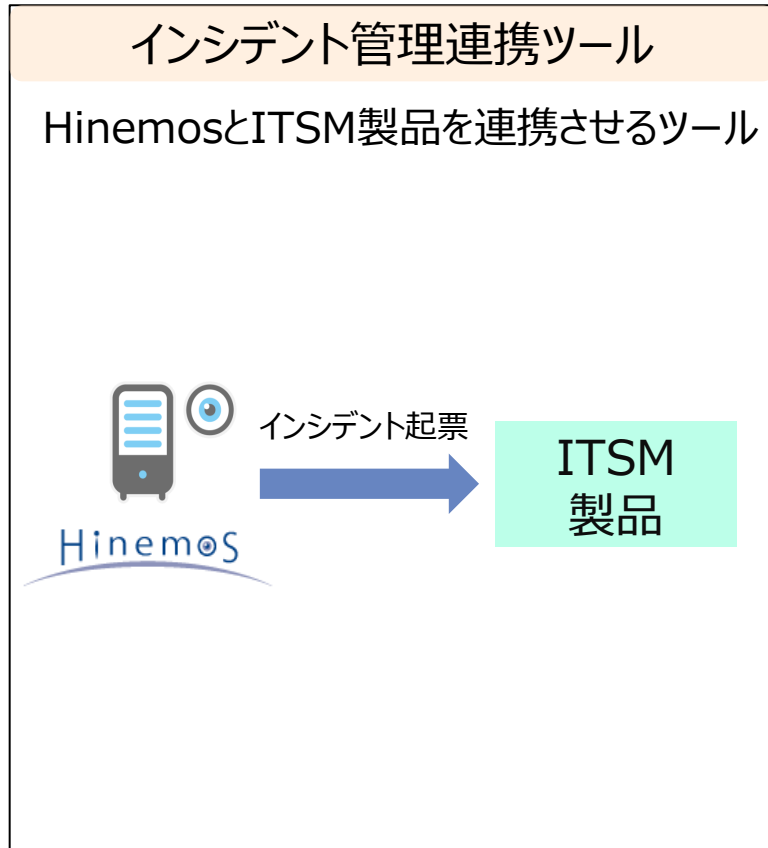
## MIBインポート

MIBファイルインポートにより、ベンダ固有のSNMPTRAPでも可読性の高いメッセージを監視可能



# Hinemos ユーティリティツール概要

Hinemos ユーティリティツールではHinemosをより活用できる以下のツール類をご提供します

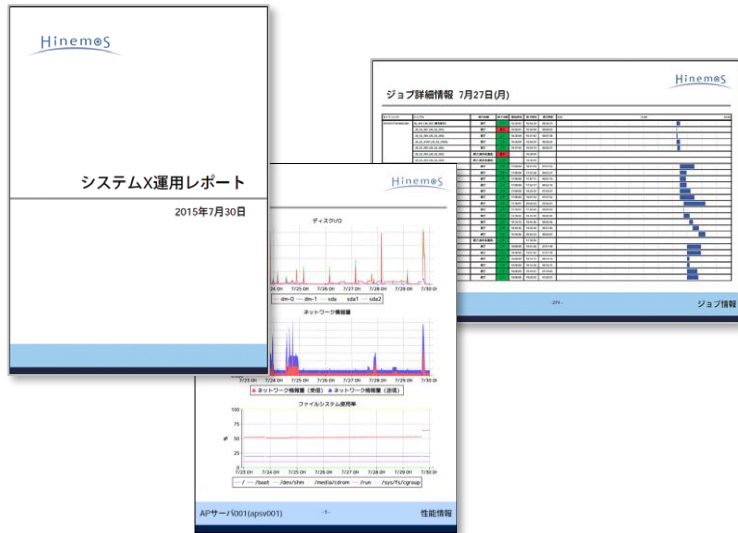


# Hinemosレポート機能概要

Hinemosに蓄積されている各種履歴情報の可視化を実現

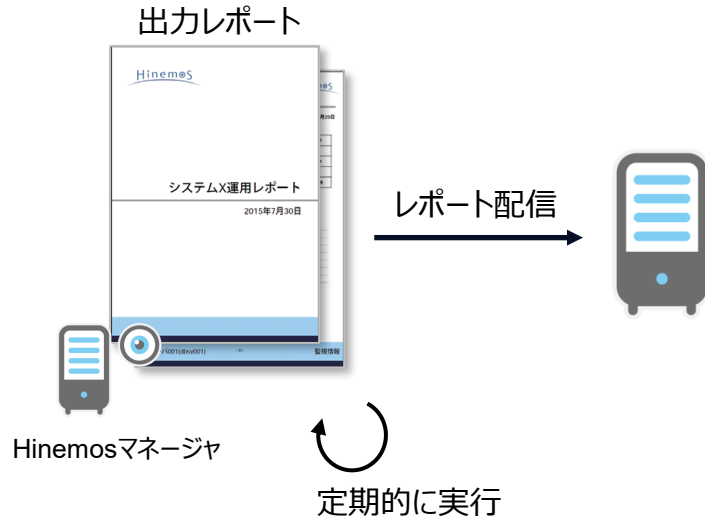
## レポート作成

システム稼働状況レポートを自動生成



## レポート運用の自動化

定期的なレポートの配信が可能



## テンプレートのカスタマイズ

テンプレートのレイアウトが変更可能



自動的にレポートを作成し、監視や実行の結果を簡単に分析

# 電話通知の概要

イベント発生時に担当者への電話・SMSによる自動通知を実現



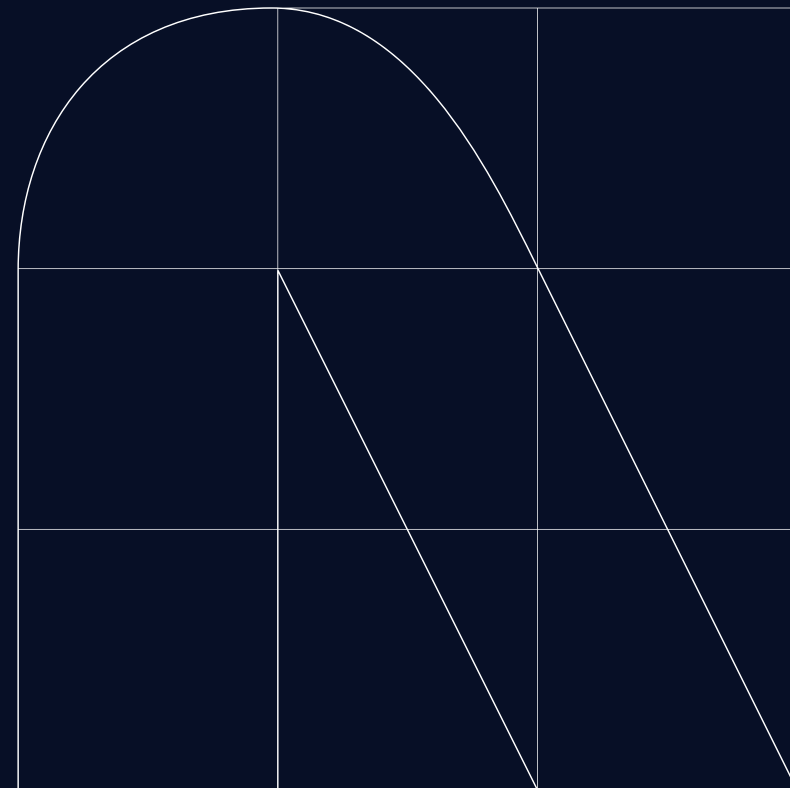
オペレーターレスで担当者の携帯端末へ通知可能

応答確認による確実な担当者への通知を実現

注1) 本機能における電話・SMSの発信にはTwilioを利用します。本機能の利用にはTwilioのご用意が必要です。

# 04

## Hinemos VM・クラウド管理機能



# Hinemosによるクラウド管理

クラウドの特徴をカバーする専用機能により、オンプレミス同様の運用が作り込み不要で実現

## クラウド管理機能

クラウドの特徴を活かしつつオンプレミス同様の運用を可能に

### 対応プラットフォーム

AWS

VMware

Azure

Hyper-V

Google Cloud

KVM

OCI

### クラウドの特徴

迅速性、柔軟性、拡張性

従量課金制

PaaS/SaaS

クラウドの組み合わせ

### 専用機能

リソースの自動検出・追従

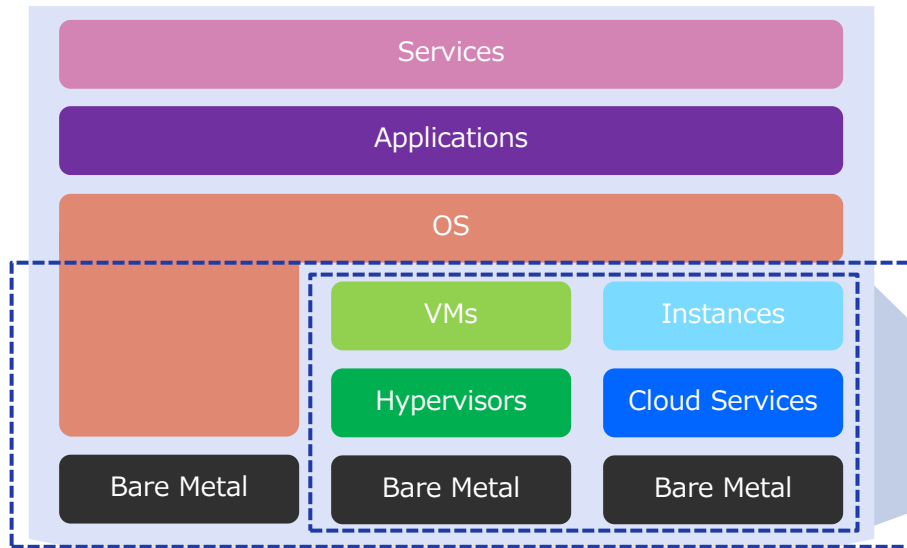
プラットフォーム監視

専用リソース監視

リソース制御

クラウド双方向通知

クラウドログ監視

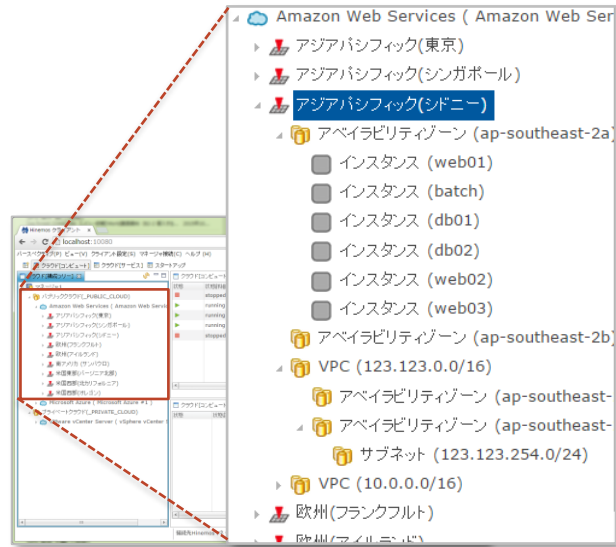


# リソースの自動検出・追随

VM・クラウドのリソースを検出・識別してリポジトリに自動反映し監視・ジョブの自動継続へ

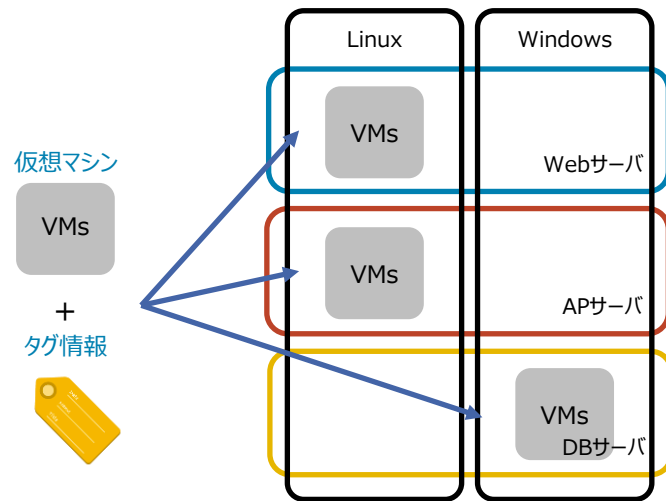
## ①自動検出

クラウド上の仮想マシン、リージョン、アベイラビリティゾーン、VPC、サブネット等の情報を自動取得



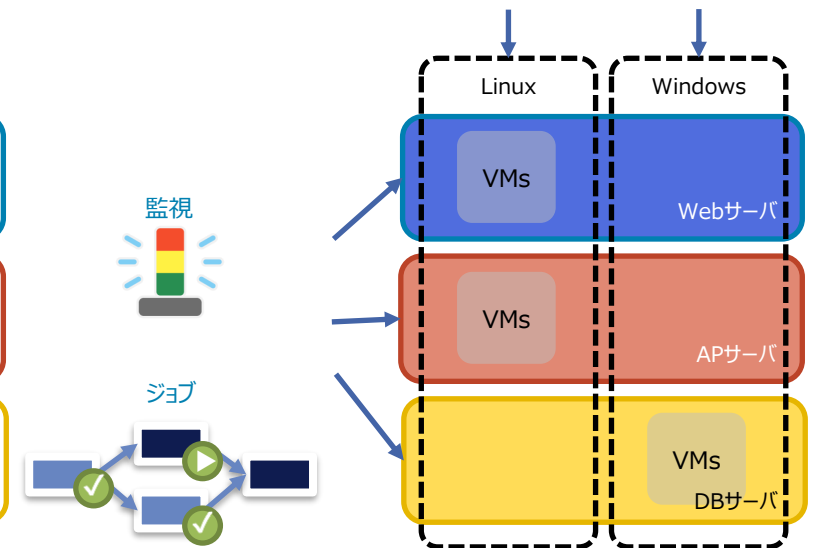
## ②識別

タグを使ってWebサーバやDBサーバといった、ユーザが管理したい単位にグルーピング



## ③監視・ジョブ開始

ユーザはWebサーバやDBサーバといった、ユーザが管理したい単位だけで監視やジョブを実行管理



VM・クラウドのリソース変更に柔軟に対応しオンプレ同様の管理を提供

# リソース変更の自動検出・追隨

マルチVM・マルチクラウドでの統合運用管理を実現

The screenshot shows the Hinemos Cloud Management Console interface. On the left, a tree view displays the cloud environment structure:

- クラウド[構成ツリー]
- マネージャ1
  - パブリッククラウド(\_PUBLIC\_CLOUD)
    - Amazon Web Services ( Amazon Web Service)
      - アジアパシフィック(東京)
      - アジアパシフィック(シンガポール)
      - アジアパシフィック(シドニー)
      - 欧州(フランクフルト)
      - 欧州(アイルランド)
      - 南アメリカ (サンパウロ)
      - 米国東部(バージニア北部)
      - 米国西部(北カリフォルニア)
      - 米国西部(オレゴン)
    - Microsoft Azure ( Microsoft Azure #1 )
    - Google Cloud (Google Cloud #1)
    - Oracle Cloud Infrastructure (Oracle Cloud Infra)
  - プライベートクラウド(\_PRIVATE\_CLOUD)
    - VMware vCenter Server ( vSphere vCenter )
    - Hyper-V (WindowsServer2016)
    - KVM (Kernel Virtual Machine)

On the right, eight panels show detailed views of resources for each cloud provider and VM type:

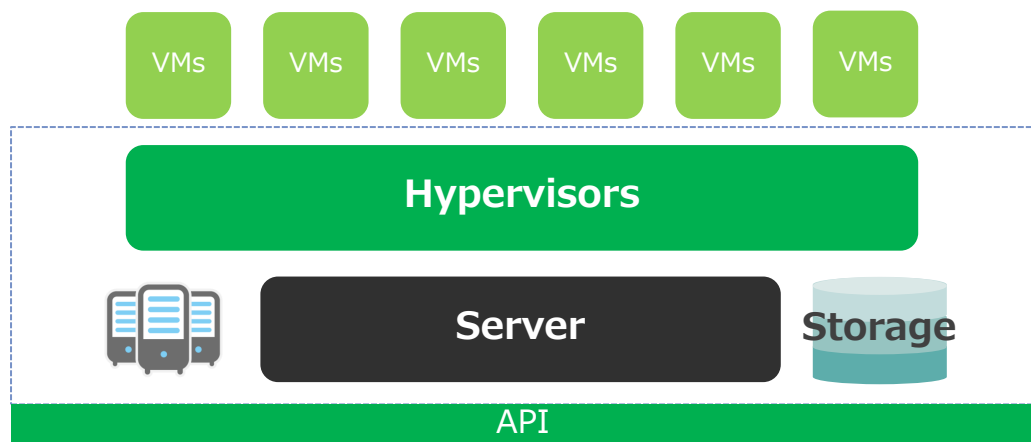
- AWS:** Amazon Web Services ( Amazon Web Ser )
  - アジアパシフィック(東京)
  - アジアパシフィック(シンガポール)
  - アジアパシフィック(シドニー)
    - アベイラビリティゾーン (ap-southeast-2a)
      - インスタンス (web01)
      - インスタンス (batch)
      - インスタンス (db01)
      - インスタンス (db02)
      - インスタンス (web02)
      - インスタンス (web03)
    - アベイラビリティゾーン (ap-southeast-2b)

- Azure:** Microsoft Azure ( Microsoft Azure #1 )
- ブラジル南部
- 米国中央部
- 東アジア
- 米国東部
- 米国東部 2
- 日本 (東)
  - クラウド サービス (cloud-jp-east-01)
    - インスタンス (east101)
    - インスタンス (east201)
  - クラウド サービス (cloud-jp-east-02)
  - Virtual Network (172.16.0.0/20)
- Google Cloud:** Google Cloud platform (Google Cl)
- アジア太平洋(シドニー)
- アジア太平洋(台湾)
- アジア太平洋(大阪)
- アジア太平洋(東京)
  - Subnet (10.17.1.0/24)
  - Subnet (10.17.2.0/24)
    - Instance (Instance A)
    - Instance (Instance B)
    - Instance (Instance C)
- OCI:** Oracle Cloud Infrastructure (Oracle C)
- ブラジル東部(サンパウロ)
- オーストラリア南東部(メルボルン)
- オーストラリア東部(シドニー)
- 日本東部(東京)
  - VCN (10.18.0.0/20 | HinemosD)
    - Subnet (10.18.0.0/21 | VcnP)
      - Availability Domain (sZVx:
        - Instance (Instance A)
        - Instance (Instance B)
        - Instance (Instance C)
- VMware:** VMware vCenter Server ( vSphere vCenter )
- データセンター (DC01)
  - ホスト (172.16.1.10)
    - vCenter\_HOST (172.16.1.10)
    - 仮想マシン (takahatat\_vcloud\_5.0a)
    - 仮想マシン (01\_esx1\_vSphereClient)
    - 仮想マシン (00\_esx1\_ssh)
    - 仮想マシン (02\_vCenterServerAct)
    - 仮想マシン (nagatsumas\_rhel7)
  - リソースプール (system\_a)
    - リソースプール (web)
    - リソースプール (batch)
- Hyper-V:** Hyper-V (WindowsServer2016)
- VMs
  - 仮想マシン (agent-lin01)
  - 仮想マシン (agent-lin02\_gn1)
  - 仮想マシン (agent-lin03)
  - 仮想マシン (agent-lin04)
  - 仮想マシン (agent-lin05\_gn1)
  - 仮想マシン (agent-lin06\_gn1)
  - 仮想マシン (agent-lin07)
  - 仮想マシン (agent-lin08\_gn1)
  - 仮想マシン (agent-lin09\_gn1)
  - 仮想マシン (agent-lin10)
- KVM:** KVM (Kernel Virtual Machine)
- qemu+ssh://172.17.4.111/system
  - ドメイン (rhel9.5)
  - ドメイン (rhel9.5\_2)
  - ドメイン (rocky-linux)
  - ドメイン (win2k22)
  - ドメイン (rhel9.5)
  - ドメイン (rhel9.5\_2)
  - ドメイン (rocky-linux)
  - ドメイン (win2k22)


# プラットフォーム監視

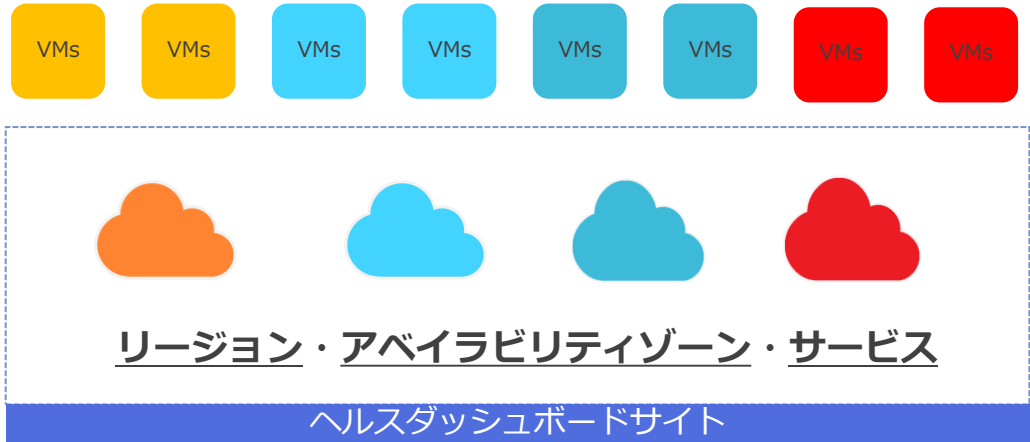
オンプレミス環境のHWに該当する、VM・クラウドの基盤の正常性を監視を開始可能

## VM基盤の正常性



- **vCenter/ESXi API**  
データストア、ホスト、ネットワーク
- **Hyper-V**  
vmmsのWindowsサービス

## クラウドの基盤の正常性



- **AWS** Service Health Dashboard
- **Azure** Azure Monitor
- **Google Cloud** Google Cloud Service Health
- **OCI** OCI Status



システム障害時にプラットフォームとアプリの問題切り分けが簡易に

# 専用リソース監視

PaaSを含めたVM・クラウド専用メトリクスの監視をリソース監視のインタフェースで提供

		オンプレ	VM	クラウド
OS(SNMP等)	-	○	○	○
専用API	OS	-	○	○
	HW/Hypervisor	-	○	-
	PaaS	-	-	○

## ・作り込み不要のシームレスな監視

リソース監視で項目を選ぶだけで、専用メトリクスを収集・蓄積・監視が可能

## ・マスタ編集機能でメトリクス追加/変更が可能

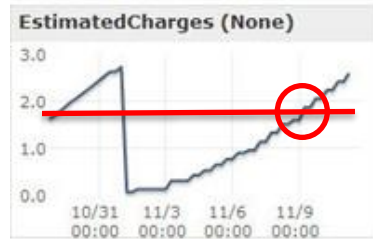
新サービス、シンメトリクスが登場したタイミングで簡易に登録も可能

VM・クラウド専用メトリクスと通常のOSリソース値を意識せずに監視可能

# 課金配賦管理

簡易な課金アラートから、配賦管理まで統合的に管理

## 課金アラート



- アカウント単位
- サービス単位



クラウド課金監視を使用する事で、クラウドサービスが提供する**アカウント単位**と**サービス単位**の課金状況に対して、そのままHinemosからアラートが可能です。

## 課金配賦管理

Webサーバ

APサーバ

DBサーバ

- スcope単位

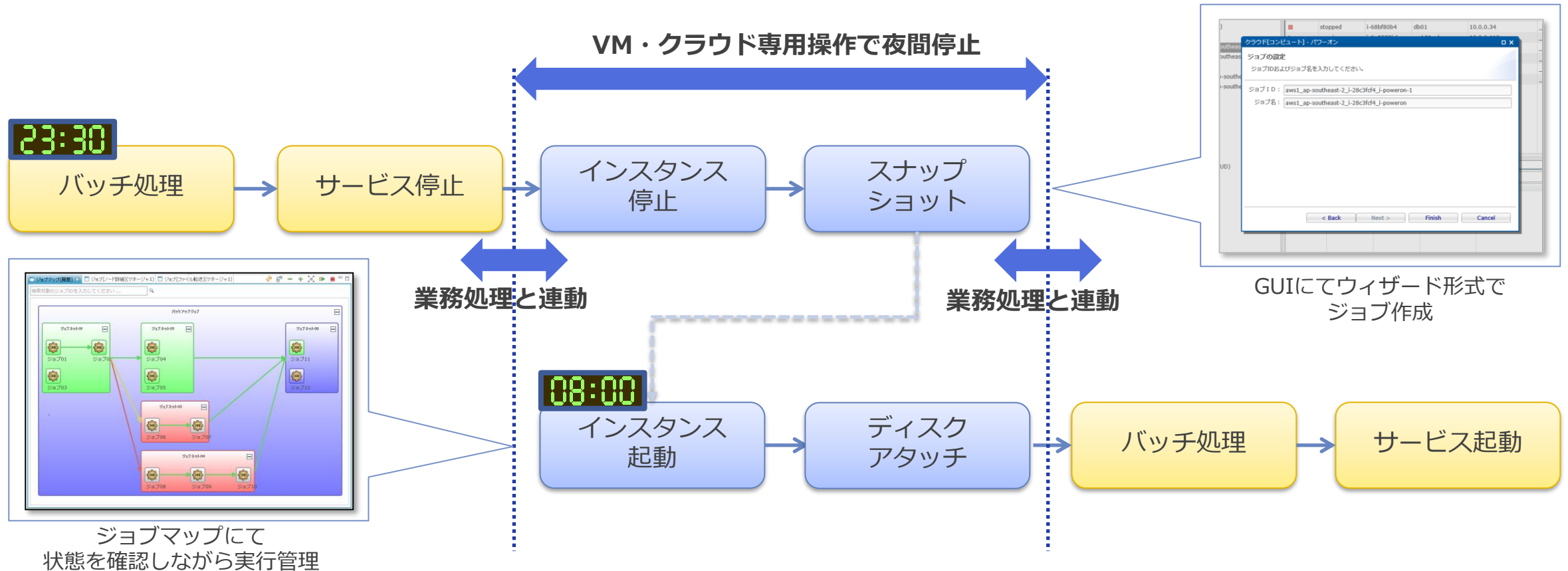


クラウド課金詳細監視を使用する事で、クラウドサービス側だけではカテゴライズが難しい、用途別の課金情報（最新情報・日増分）でアラートが可能です。

プラットフォーム視点ではなく、ユーザ視点で必要な課金情報を管理可能

# リソース制御

仮想マシン・ストレージ操作を専用ジョブにより提供

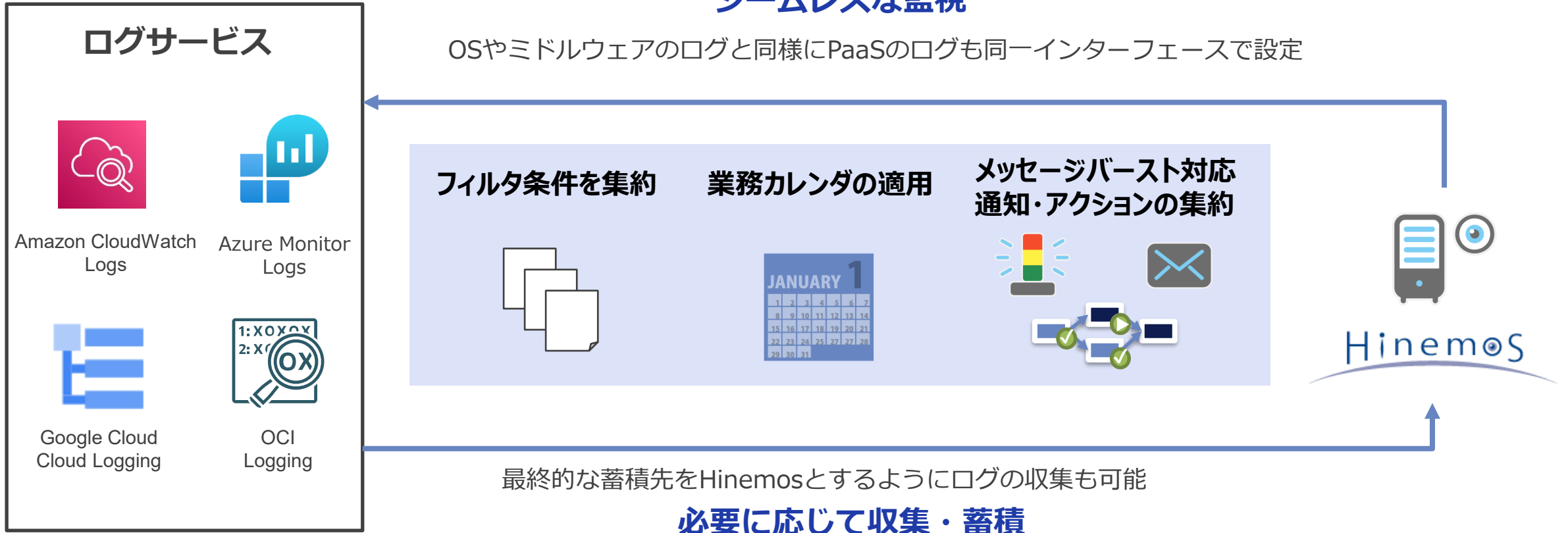


業務処理と連動したリソース制御を含めた**自動化**を簡単に実現

# クラウドログ監視

パブリッククラウド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



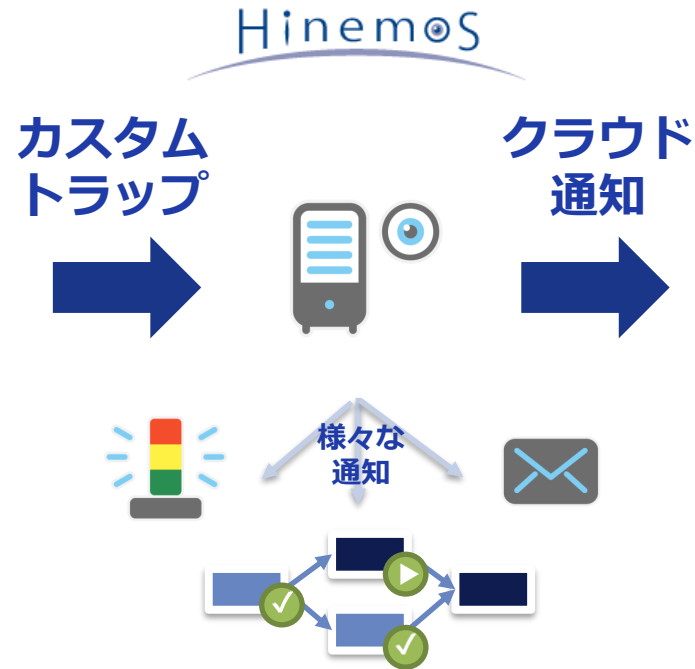
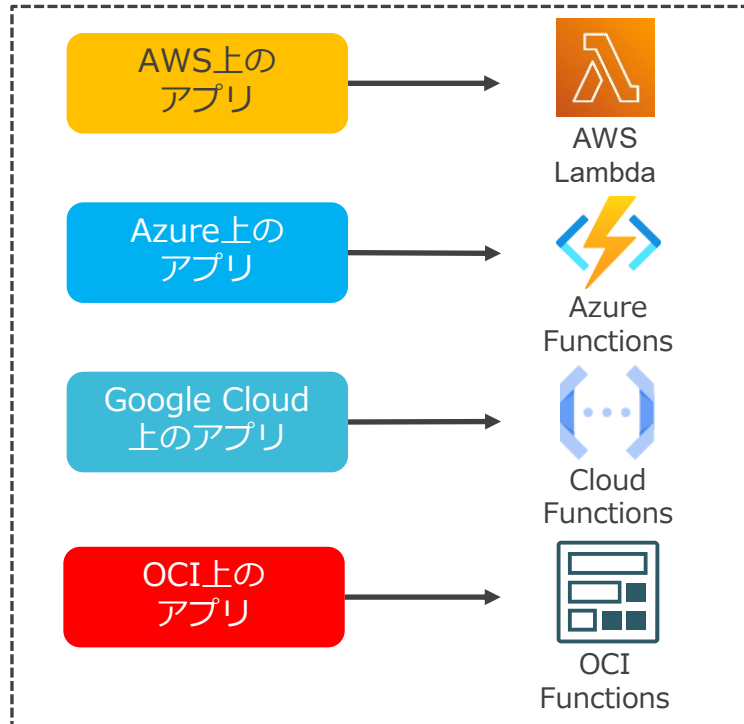
オンプレ・クラウドを跨った統合ログ管理を実現

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  
注3) クラウド管理機能Google Cloud版、OCI版については、ログ監視用のサンプルスクリプト/設定の提供となります

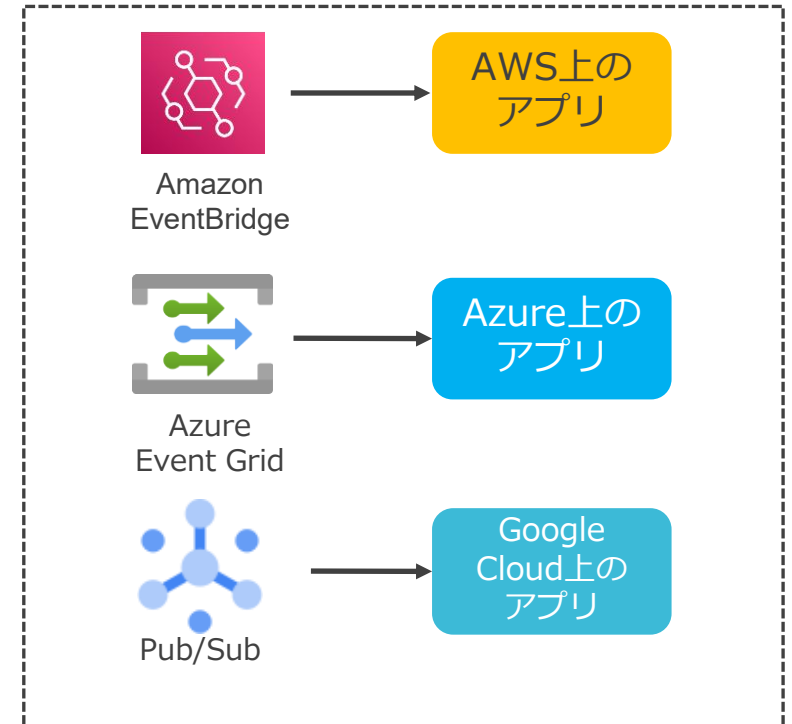
# クラウド双方向通知

通知のバリエーションとしてAmazon EventBridge、Azure Event Grid、Google Cloud Pub/Subに対応

クラウドから



クラウドへ

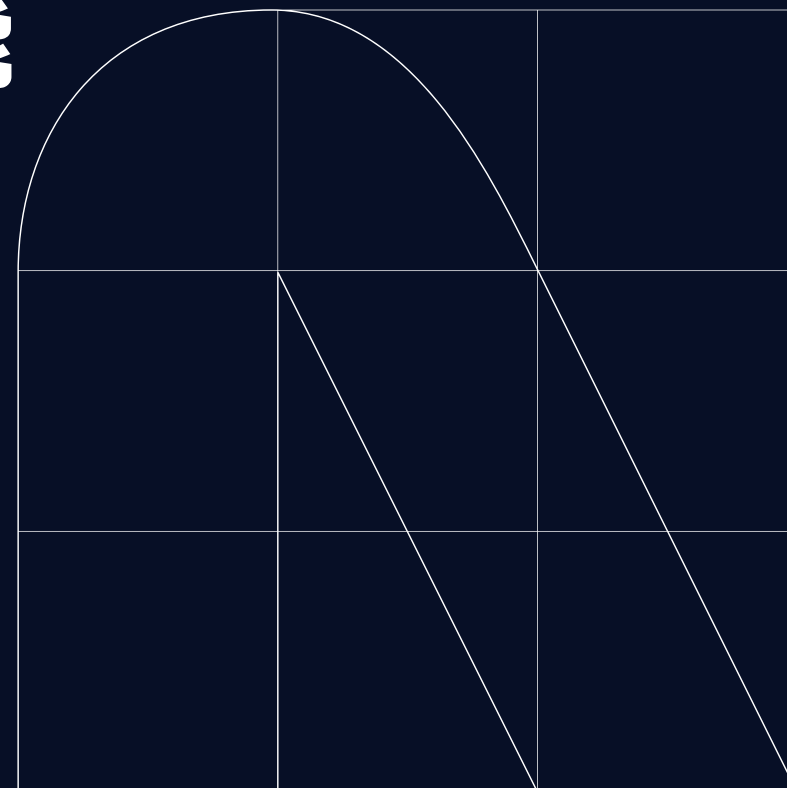


通知の集約とクラウド間のブリッジによりシームレスな自動化を実現

注1) AWS Lambda、Azure Functions、Cloud Functions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

# 05

## Hinemos ミッションクリティカル機能



# Hinemosミッションクリティカル機能とは

統合運用管理機能を提供するHinemosマネージャの多重化をワンパッケージで提供！  
収集・監視・自動化の高可用性を実現

## Hinemosミッションクリティカル機能の特徴

簡易構築・低コスト

- ① クラスタリングソフト・共有ディスク不要
- ② マルチプラットフォーム対応

統合運用管理機能の可用性

- ③ 障害検知と自動系切替による運用継続
- ④ 系切替中のメッセージロスト防止機構

簡易運用・オンライン復旧

- ⑤ オンライン系切戻しによる障害復旧
- ⑥ 1コマンドによる系切戻し

# ①クラスタリングソフト・共有ディスク不要

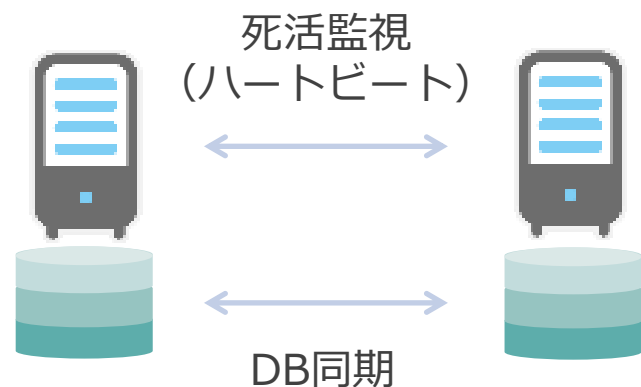
簡易構築・低コスト

通常のOS2台だけで、Hinemosマネージャを多重化！

## Hinemosマネージャ

マスタサーバ

スタンバイサーバ



Hinemos自身がソフトウェアで  
死活監視とDB同期を実施

クラスタリングソフトの  
追加費用が不要

設計・構築時の  
SE・CEコストを削減

基盤要件がシンプル  
マシンを2台用意するだけ

共有ディスクの  
追加費用が不要

障害発生時にも  
ワンストップサポート

マルチプラットフォーム  
対応

ソフト・ハードウェア費用とエンジニア工数を一気に削減

## ②マルチプラットフォーム対応

簡易構築・低コスト

オンプレミス・仮想化・クラウド環境を全て同一アーキテクチャで対応

オンプレミス



IAサーバを2台用意するだけ

仮想化



ハイパーバイザが異なる  
仮想マシンを2台用意するだけ

クラウド



インスタンスを2台用意するだけ  
VPC Peer/DirectConnectにも対応

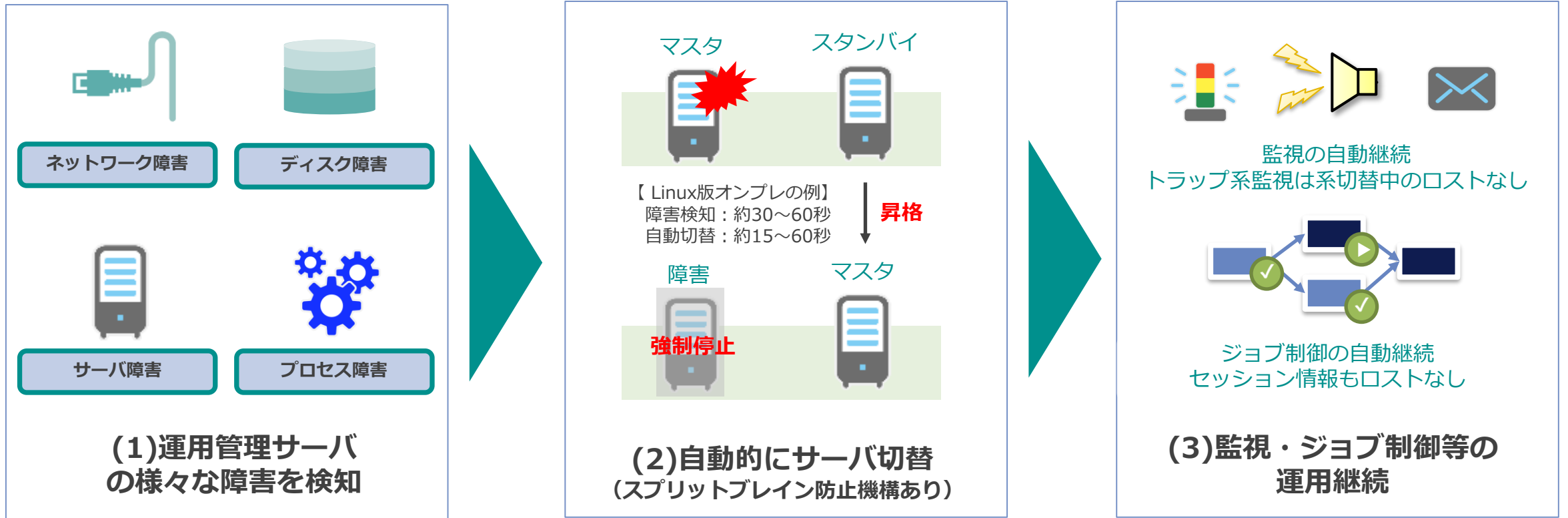
環境が変わっても、Hinemos自身がソフトウェアで死活監視とDB同期を実施するだけ

プラットフォームが変わっても、**同じスキルセットで導入可能**

### ③障害検知と自動系切替による運用継続

運用管理機能の可用性

環境に合った様々な障害を検知し、自動切替を実現



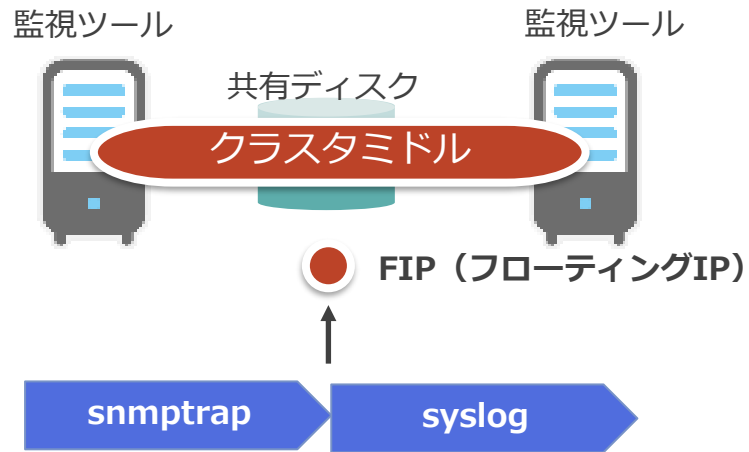
素早い自動切替により、監視やジョブ制御を止めることなく運用継続

## ④系切替中のメッセージロスト防止機構

運用管理機能の可用性

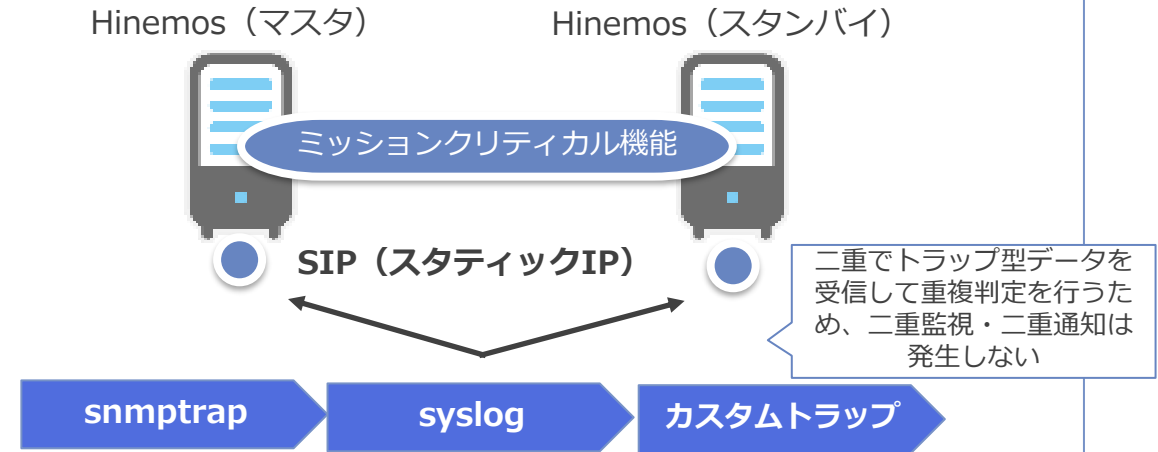
両系でトラップ型データを受信するトラップ型監視のロスト防止機構

### 一般的な監視ツールのクラスタ構成



FIPでトラップ型データを受信するため  
系切替中のトラップ型データは受信できない

### Hinemosミッションクリティカル構成



マスタ・スタンバイの両サーバの  
SIPでトラップ型データを受信をして  
系切替中の受信担保をする

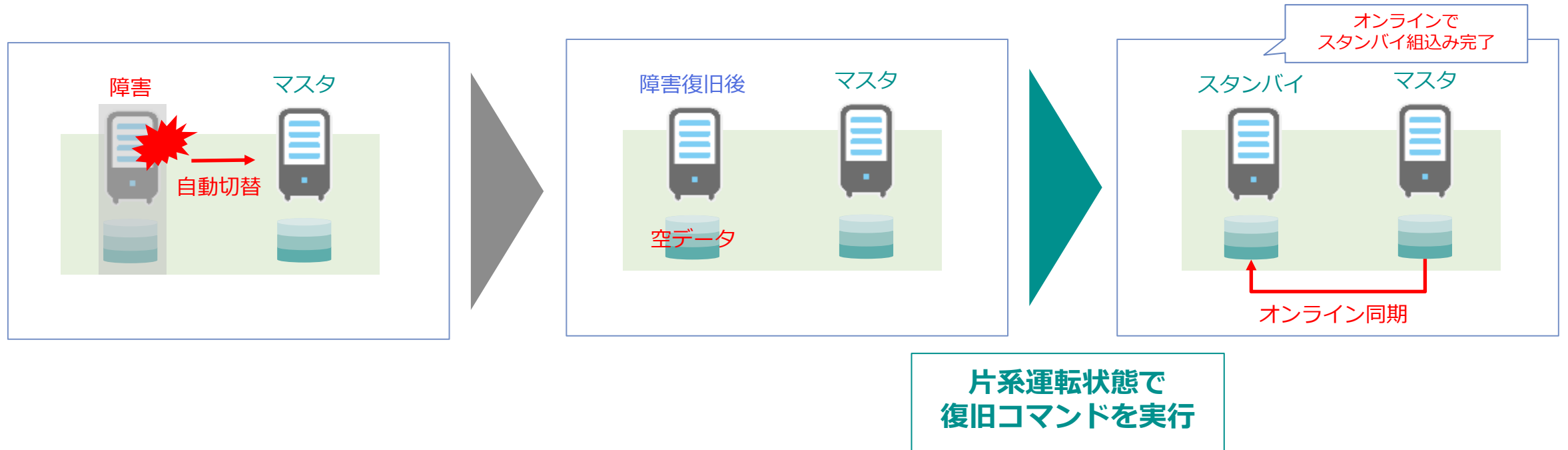
**Hinemosマネージャの系切替中でも、重要なトラップをロストせずに監視**

## ⑤ オンライン系切戻しによる障害復旧

簡易運用・オンライン復旧

片系運転からの両系運転への切戻しも、オンラインで実現

障害時の自動切替が出来ても、両系運転に切戻す際に停止が発生しては意味がない



障害発生から障害復旧まで、全てをシステムに影響なく無停止で対処！

## ⑥1コマンドによる系切戻し

簡易運用・オンライン復旧

複数の障害パターンに対して、提供する復旧コマンドは1つのみ

### 一般的な運用管理製品のクラスタ構成



- ・ 障害ポイントが多い
- ・ 対処する操作は障害ポイントによる
- ・ 複数のエンジニアが必要



パターンごとの復旧手順書が必要

### Hinemosミッションクリティカル構成



- ・ Hinemos視点で障害切り分け可
- ・ 復旧コマンドは1つだけ
- ・ Hinemosエンジニアのみで十分



復旧手順書のパターンは不要

シンプルな手順書かつオペレータでも実施できる簡易操作で復旧が可能

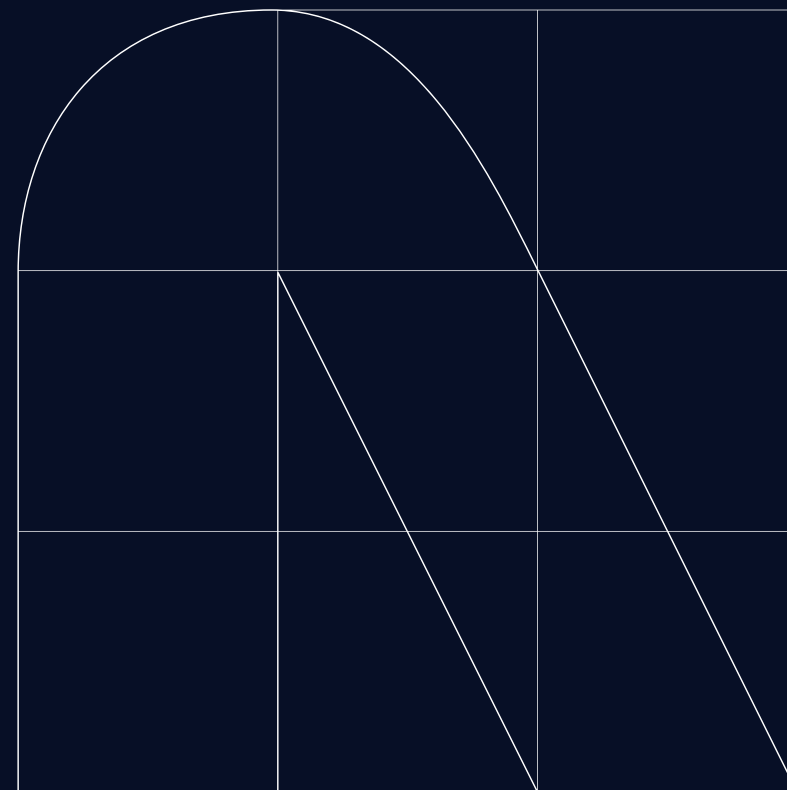
# クラスタ方式からみたHinemosの優位性

	Hinemos		一般的なクラスタ方式の製品	
費用	◎	Hinemosのソフトウェア費用のみ	△	クラスタミドル、共有ディスク、構築費用など
切替時間	○	素早い切替（1～2分）	△	数分～数十分かかるものも
監視	◎	トラップ型監視のロスト防止機構	×	トラップ型監視はロスト
ジョブ	○	ジョブ制御の自動継続	○	ジョブ制御の自動継続
復旧	◎	1つの復旧コマンド	×	障害パターンごとの復旧方法
クラウド	◎	プラットフォーム非依存で全機能を高可用化	×	クラウドにより実現可否・制約がある
保守性	◎	ワンストップ保守サポート	×	障害ポイントごとに異なる保守サポート

設計・導入・運用そして費用の全てにおいて優位性がある

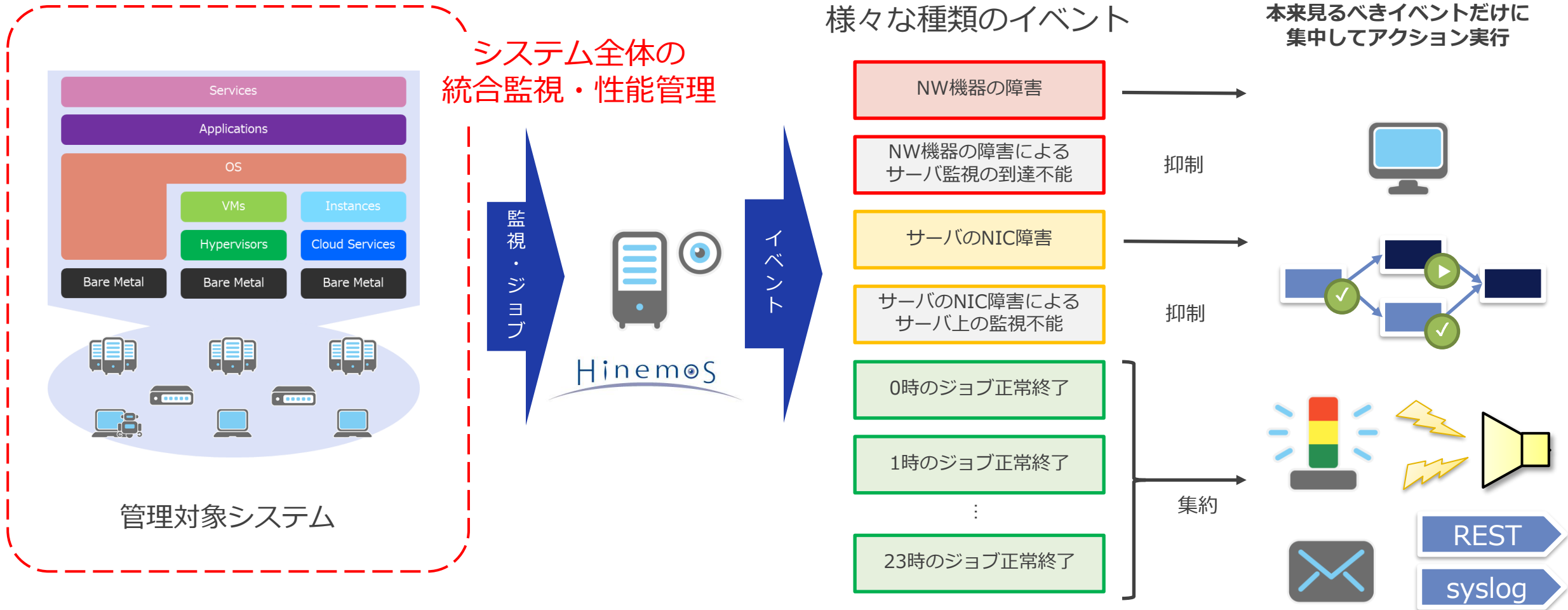
# 06

## Hinemos メッセージフィルタ



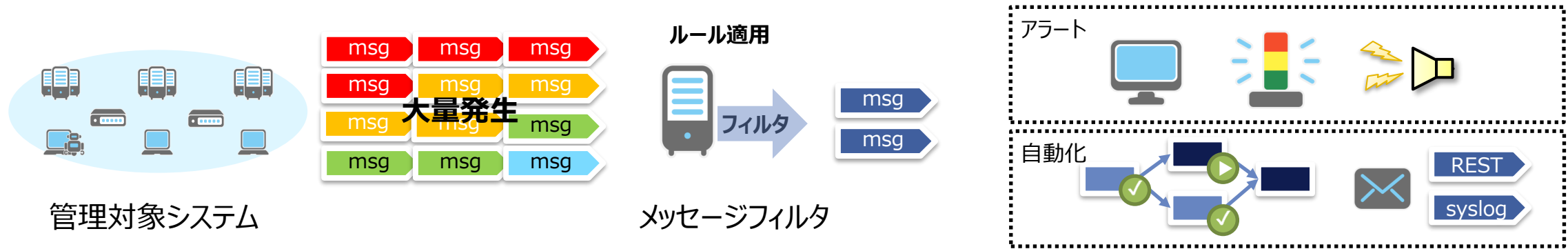
# 背景 本質的なイベント

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

ルールエンジンを活用したインテリジェントなアラートと自動化



## メッセージフィルタの4つの特徴

①インテリジェントなアラート

②インテリジェントな自動化

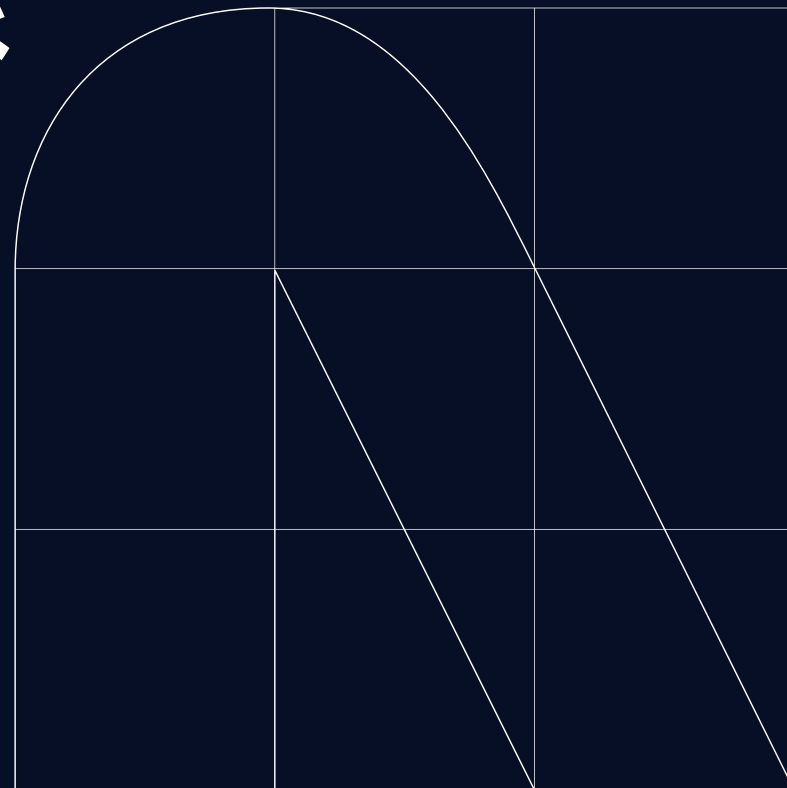
③ルールベースの条件指定

④Hinemosからのシームレスな導入

「本質的なイベント」の発見を実現

# 07

## Hinemos インシデントダッシュボード



# Hinemos インシデントダッシュボードの概要

インシデントの起票から調査、対応、クローズまでの管理を、軽量かつ使い慣れたExcel-likeなUIにて実現



「記録」と「共有」をスムーズに実現し、組織的なインシデント管理をトータルサポート

# Hinemos インシデントダッシュボードの機能 (1/2)

見やすく操作しやすい表示形式と、組織での管理をサポートする機能を提供

## Excelライクでシンプルな画面

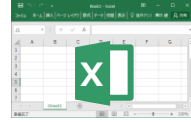
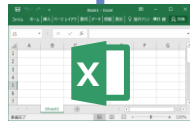
管理項目を容易にカスタマイズ可能  
CSV連携で、Excel運用からの移行も簡単

ID	発生時刻	担当者	インシデントの種類	状態	発生日時	検出日時	解決日時	発生場所	発生原因	発生種別
9	ping01	manager01	ping01	発生中	2022/10/26 17:35	2022/10/26 17:35	2022/10/27 17:35	14 時間前	Pinging 127.0.0.1 (127.0.0.1). Ping statistics for 127.0.0.1: Packets: Sent = 1, Received = 1, Lost = 0 (0% loss), Approximate round trip time in milliseconds: Minimum = 0.03ms, Maximum = 0.03ms, Average = 0.03ms	Administrator

インポート      CSV形式      エクスポート



既存ナレッジの活用・集約

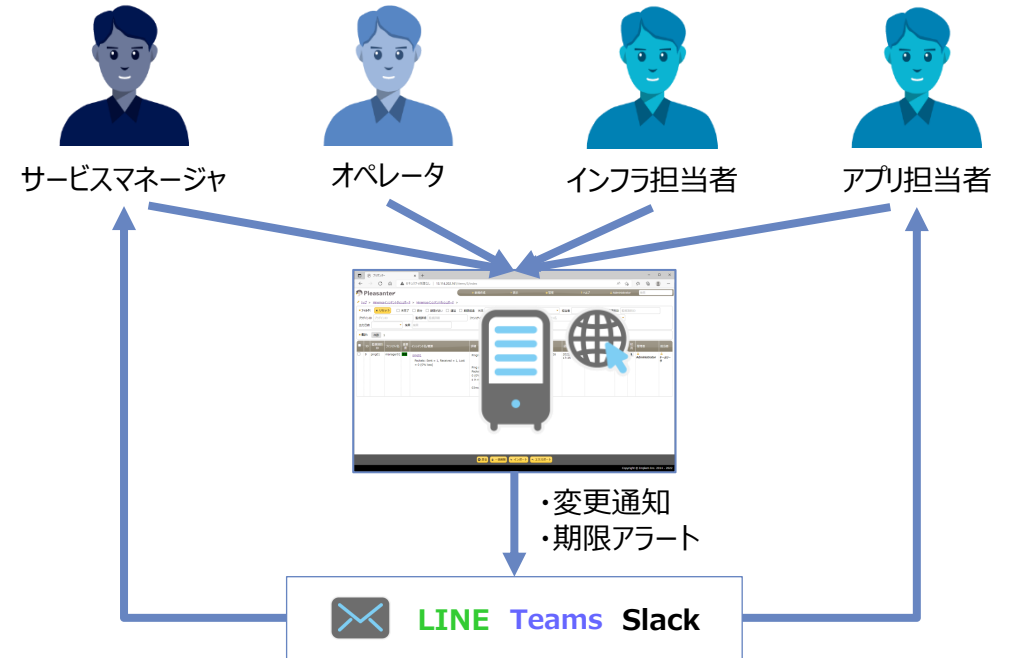


既存運用も踏襲可能



## 組織での管理をサポート

Webベースで情報を共有し  
様々なコミュニケーションツールに連携可能



▶ 初めてインシデント管理する方も安心して導入可能

▶ 複数メンバーでの効率的なインシデント管理を実現

# Hinemos インシデントダッシュボードの機能 (2/2)

目的に合わせた表示ビューを具備し、Hinemosからの自動起票によりインシデント管理の効率化を実現

## 目的に合わせた表示ビュー

インシデント起票

インシデント一覧

ID	監視項目	ファンクション名	重要度	インシデント名/概要	詳細	対応開始	出力日時	経過日数	期限	進捗率	状況	管理者	担当
2				リモート接続が頻りに切断する	踏み台サーバへの接続を切断して、接続しなおせば動くようになる。	2020/07/23 11:37		2020/07/24 4日後	60%	🟡	🚩	th01	mw01
4				システムにログインできない	午後からシステムにアクセスできない状態となっている。	2020/07/20 13:12		2020/07/21 1日後	75%	🟢	🚩	th01	th01
8				システムのリソースが不足		2020/07/18 13:53		2020/08/21 2ヶ月後	15%	🔴	🚩	th01	ap01
3				データ更新処理でエラー	データ更新処理でDBがエラーを吐く	2020/07/22 11:46		2020/07/23 3日後	50%	🟢	🚩	th01	th01

インシデント起票

インシデント一覧

様々なビューで表示可能

カレンダー

ガントチャート

かんばん方式

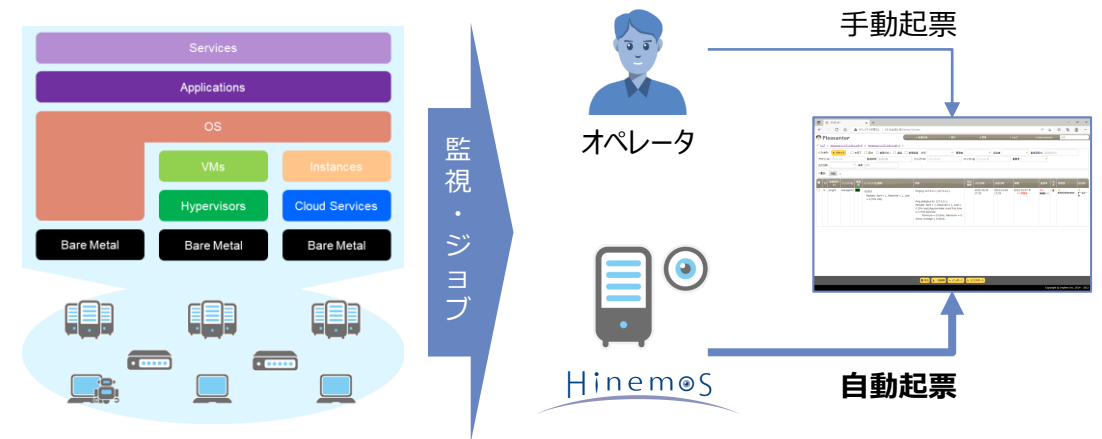
合計: (0h \* 0%)

- システムのレスポンスが遅い (0h \* 25%) ap01: 準備
- システムにログインできない (0h \* 75%) th01: 実施中
- データ更新処理でエラー (0h \* 50%) th01: 実施
- リモート接続が頻りに切断する (0h \* 60%)

▶ 用途に合わせた表示で管理を効率化

## Hinemosからの自動起票

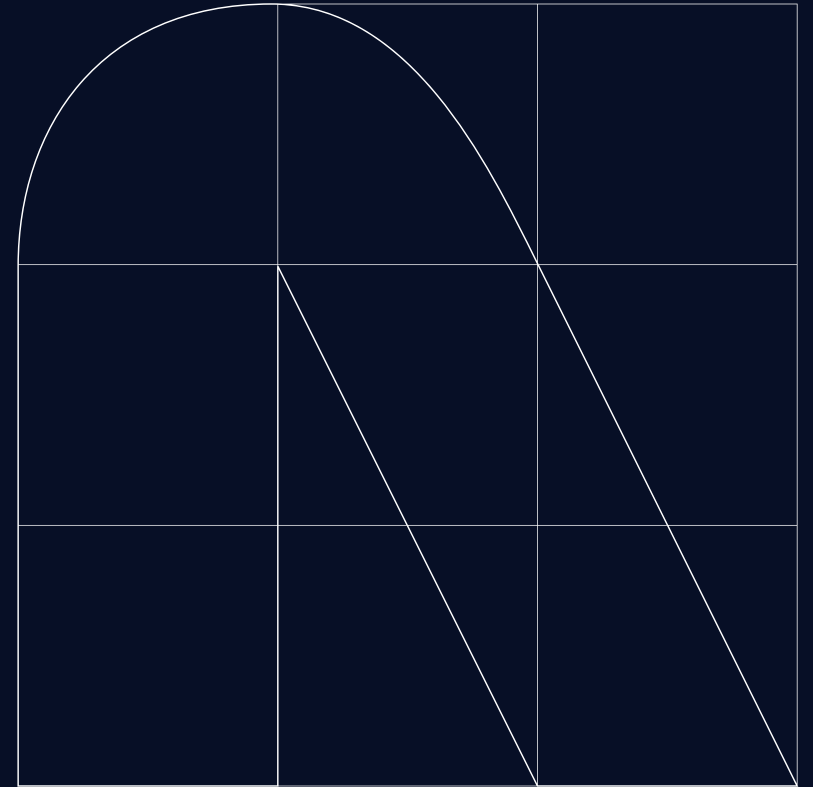
Hinemosで検知したイベントを自動でインシデントダッシュボードに起票が可能



▶ 自動起票により記票ロスの防止と工数削減

# 08

## appendix



Hinemosではじめる

## 実践ジョブ管理・自動化入門 技術評論社

2023.3.9 発売開始

最新バージョン ver.7.0に対応

Amazonから購入可能



# お問い合わせはこちら

## Hinemosに関するお問合せ

お気軽にお問合せください。

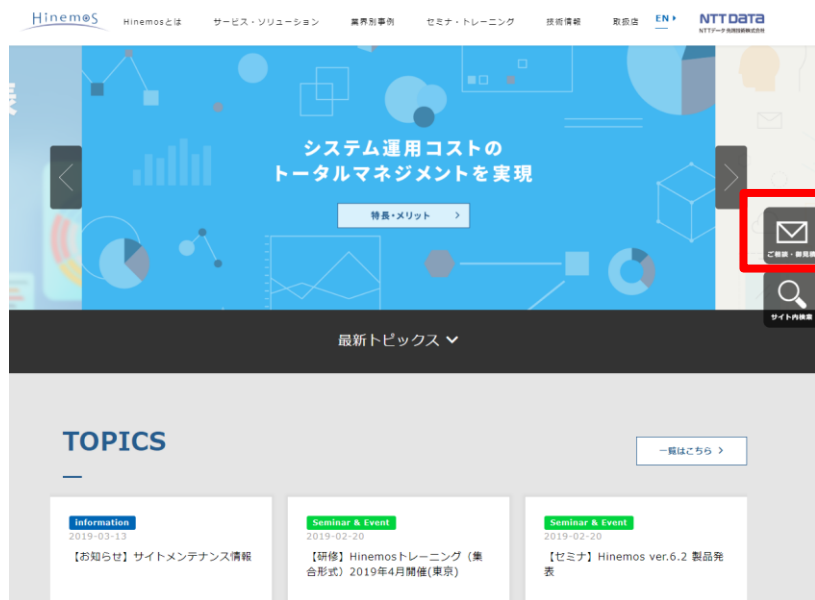
Hinemosポータルサイト

URL : <https://www.hinemos.info/contact>

Hinemos



お待ちしているもに！



ご相談フォーム



