

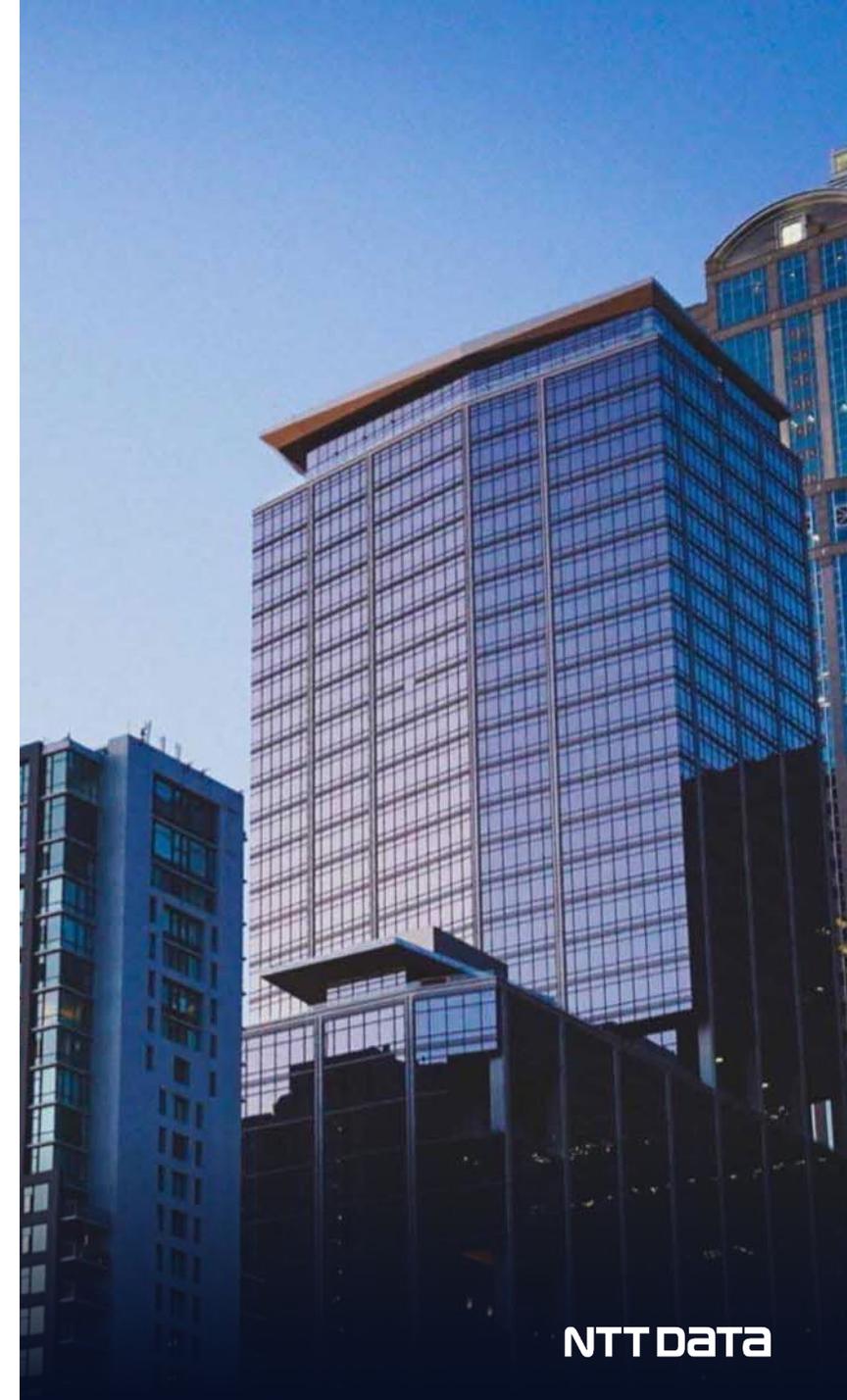
生成AIを用いた運用自動化・効率化の実現



Hinemos

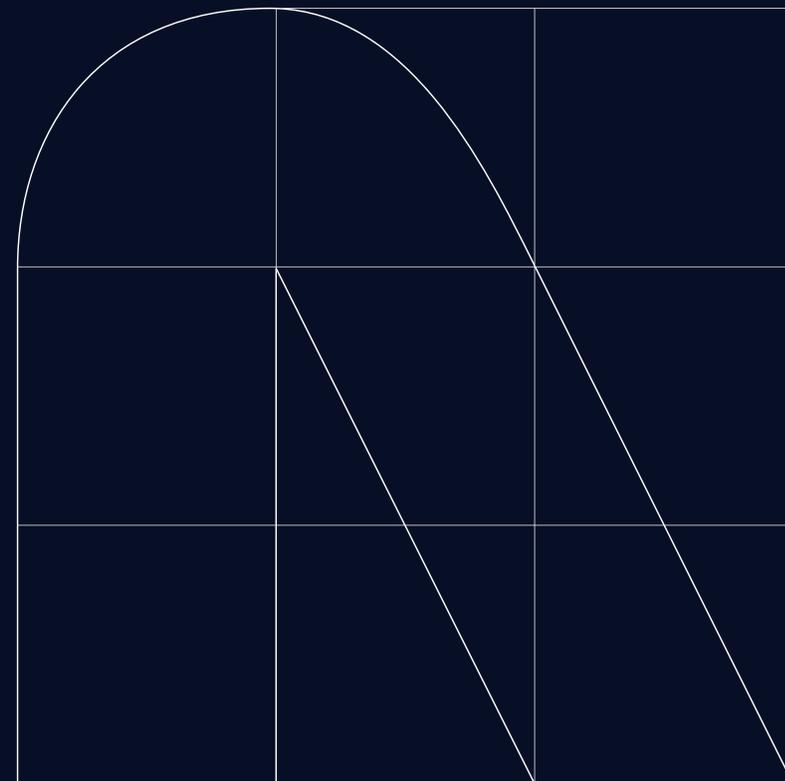
目次

1. 生成AIを使ったAIドリブン運用の世界観
2. Hinemosを使った運用自動化
3. AIドリブン運用のデモンストレーション
4. まとめ



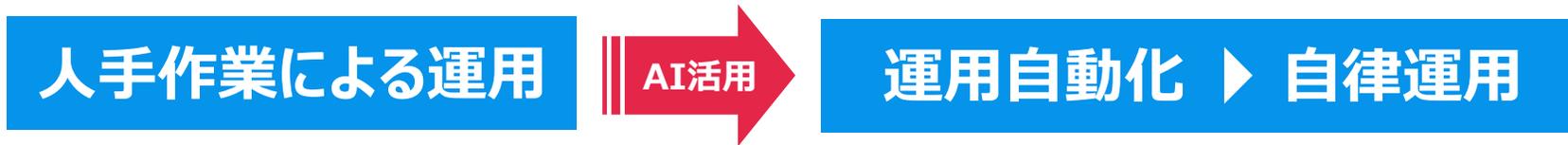
01

生成AIを使ったAIドリブン運用の世界観



AIドリブン運用とは

これまで人手作業で実施していたものをAIベースにすることで
飛躍的に運用を効率化し、生産性の向上を実現



得られる効果

- ① システム運用品質の向上
- ② 故障対応・運用改善のアジリティの向上
- ③ システム運用コストの低減

AIドリブン運用による運用課題の解決

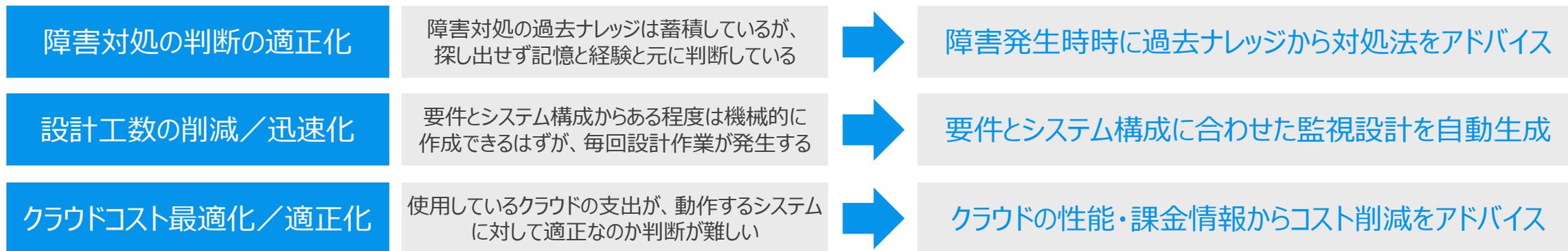
AIを運用に活用するAIドリブン運用により ITシステム運用の飛躍的な「品質向上」「アジリティ」「コスト低減」を実現

解決すべき課題：

- DX推進によるシステムの増加、複雑さに伴い運用品質の確保が難しい
- 人手に頼った運用では新たなシステムの受入れ、運用改善に迅速に対応できない
- クラウド活用等により不要なコスト原因の特定が困難

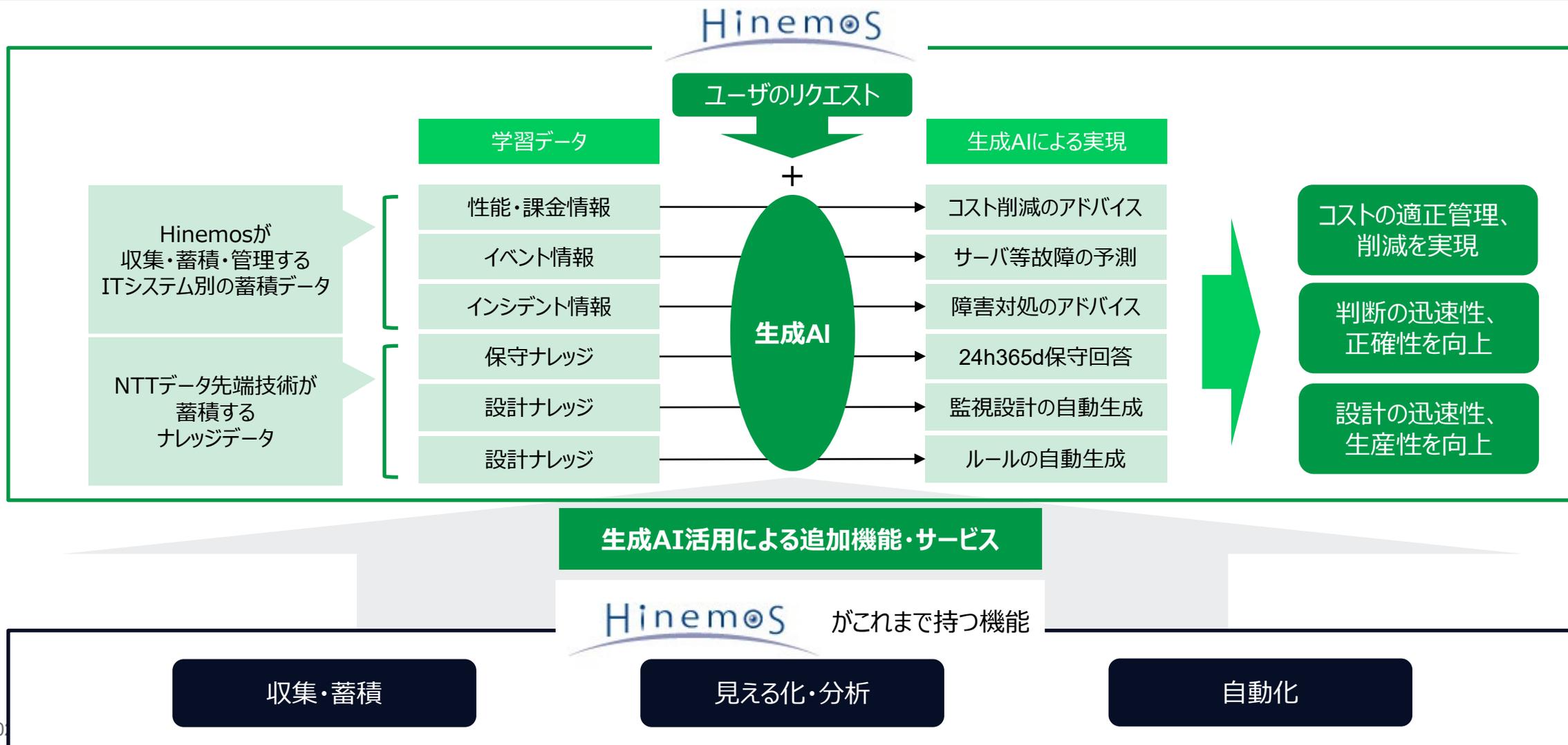
具体的な運用現場の課題例

生成AI活用による解決



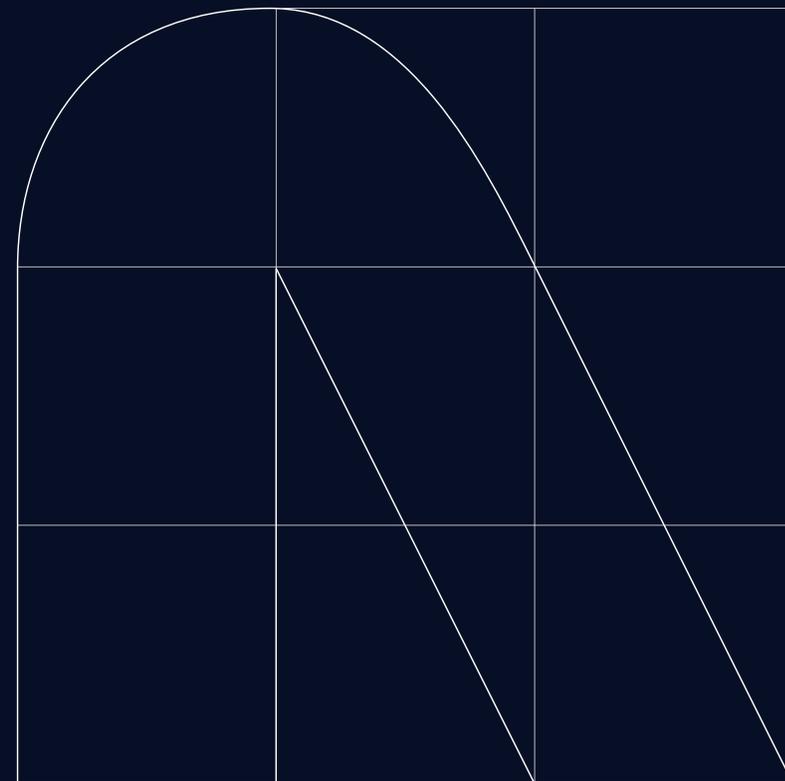
HinemosのAIドリブン運用の推進

HinemosがITシステムから「収集・蓄積」したデータと
NTTデータ先端技術社が過去に培ったナレッジを活用しAIドリブン運用を実現



02

Hinemosを使った運用自動化



Hinemosが提供する機能・サービス

Hinemos

Collection

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など、多種多様な情報を収集・管理します。

Monitoring

監視・性能

システム運用状況の効率的な監視・視覚化、異常検知をします。

Automation

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、RPAの様々な自動化を集約管理します。

Common/Basic

リポジトリ

システム構成をノードとスコープで定義し、各ノードのデバイス情報、パッケージ等の取得・履歴管理が可能です。

アカウント

各アカウントごとにロール（役割）を指定し、オブジェクト権限とシステム権限で参照・操作は範囲を制御します。

通知

監視やジョブの実行結果に応じて通知し、また瞬間的な異常やメッセージラッシュにおける通知抑制も可能です。

カレンダー

監視やジョブを実行する日、実行しない日や、GW等における振替実行の定義可能です。

エンタープライズ

マップやレポート、設定エクセル編集機能など、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します。

Cloud

クラウド管理

クラウドの特徴をカバーする専用機能を提供します。

High Availability

ミッション クリティカル

通常のOS2台だけでHinemos マネージャを多重化をソフトウェアの機能として提供します。

IT Service Management

Hinemos インシデントダッシュボード

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ライクなUIにて実現します。

Intelligent

Hinemos 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します。

Security

Hinemos セキュリティ ネットワーク診断オプション

ネットワークシステム（ネットワーク機器、サーバー）を検査し、セキュリティの脆弱性を洗い出します。

Hinemos セキュリティ アプリケーション診断オプション

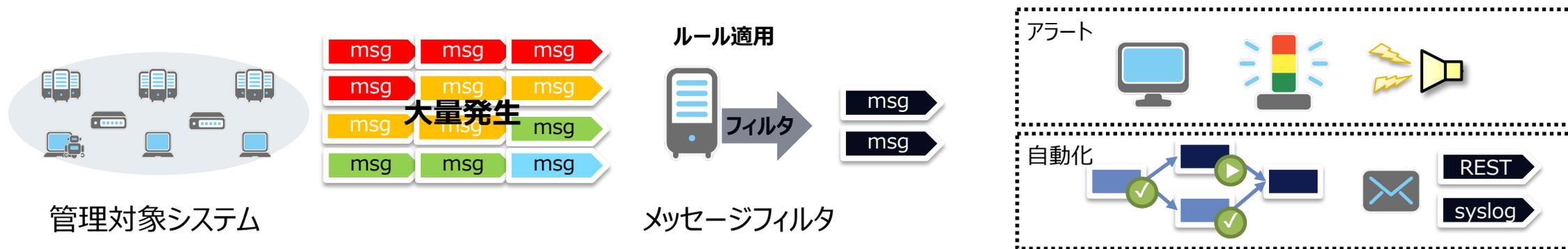
HTTP/HTTPS通信が可能なインターネットに公開されているWebサイトを検査し、セキュリティの脆弱性を洗い出します。

Hinemos セキュリティ 情報配信オプション

脆弱性情報に関する情報収集業務を支援します。

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

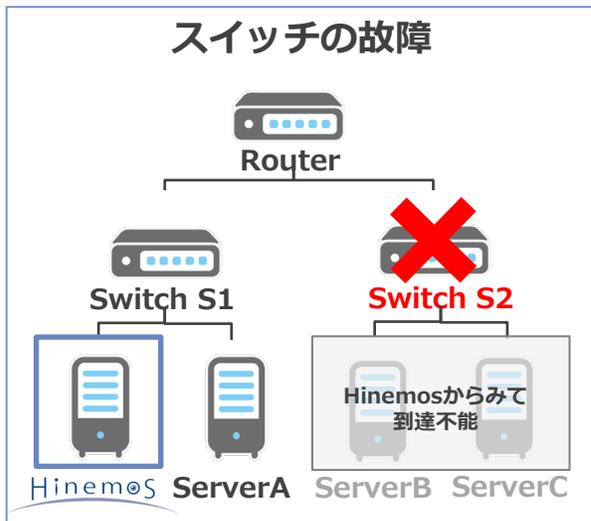
④Hinemosからのシームレスな導入

「本質的なイベント」の発見を実現

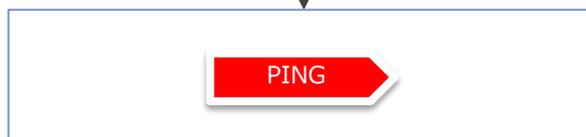
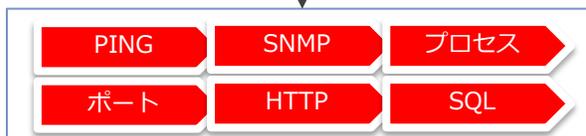
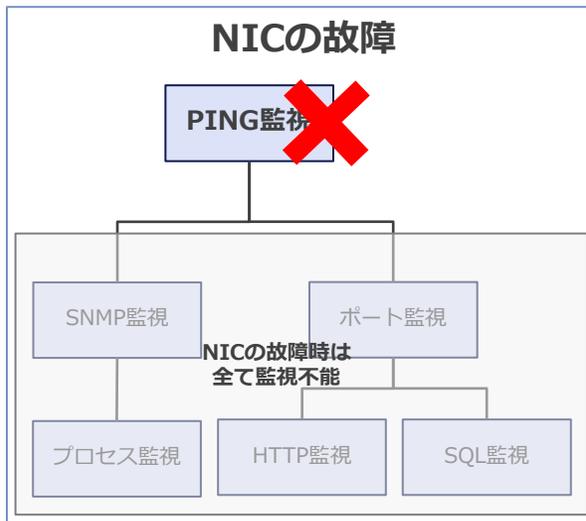
①インテリジェントなアラート

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

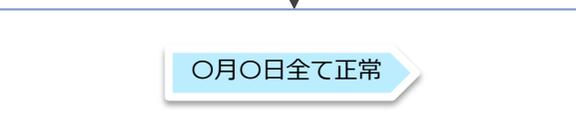
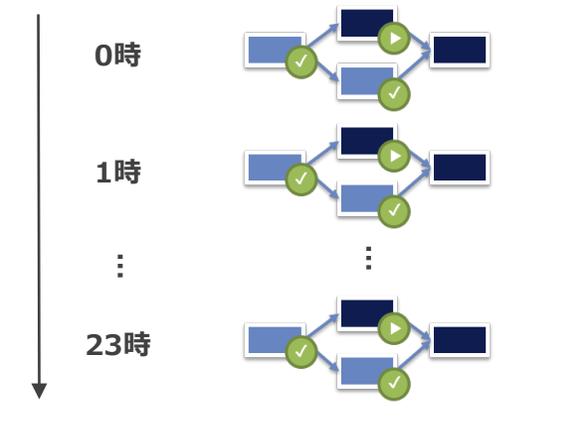


NICの故障

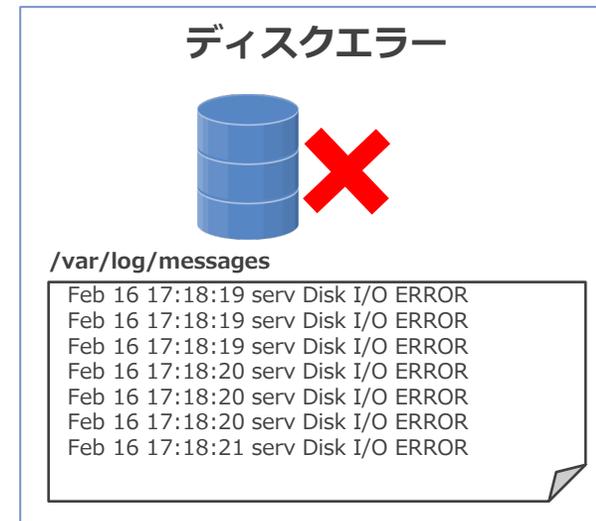


関連メッセージの集約

毎時発生するジョブ終了イベント



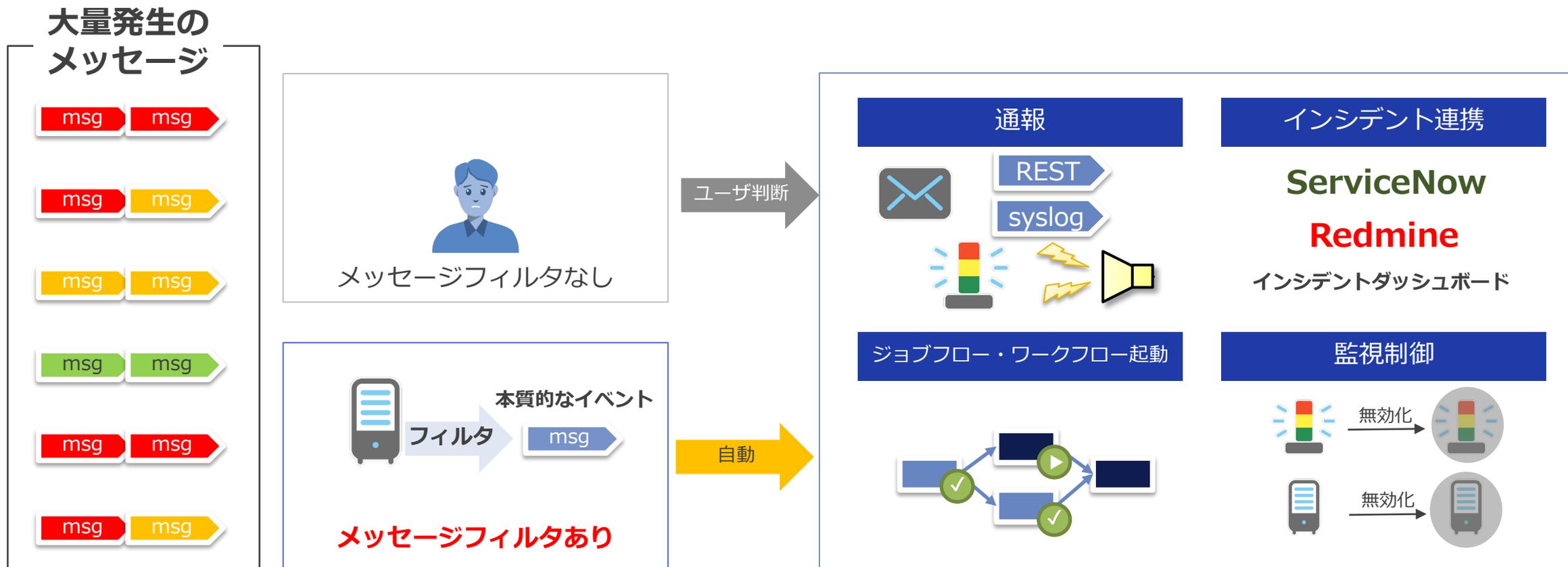
ディスクエラー



本質的なイベントへのフィルタリング (抑制・集約)

②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

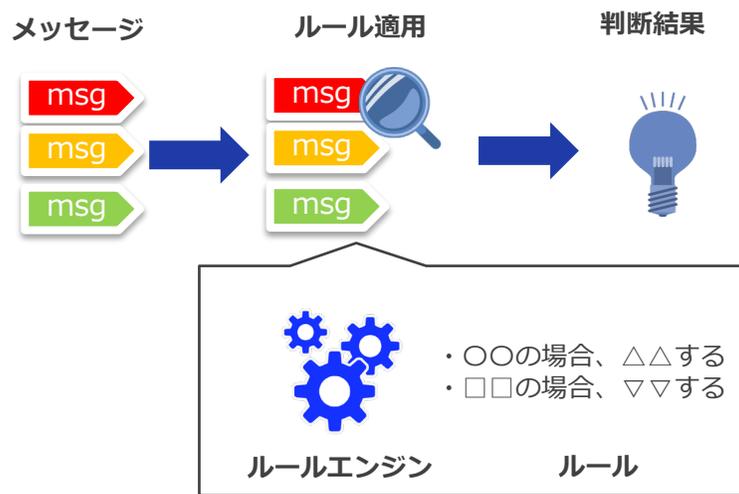


③ルールベースの条件指定

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

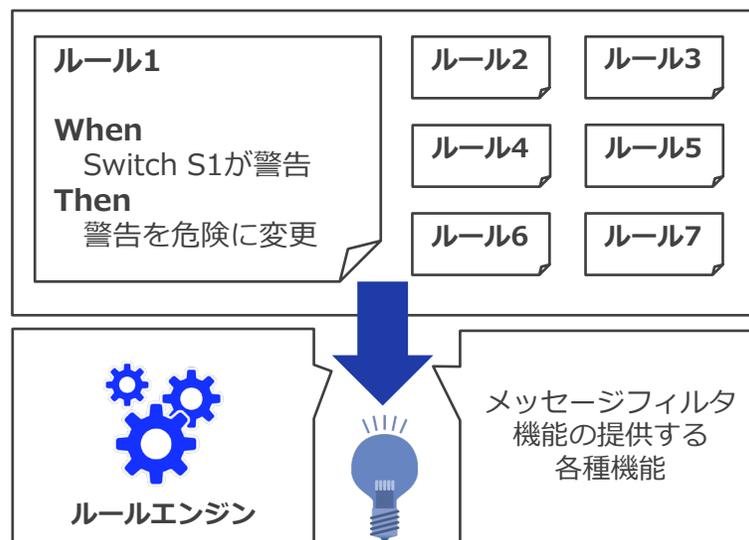
ルールエンジンとは、与えられたルールに従って判断を行う機能です



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

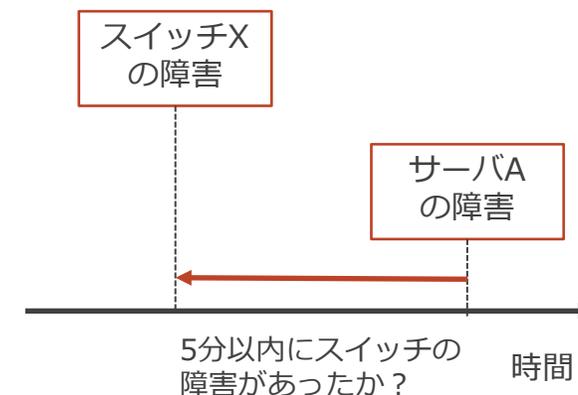
When/Thenからなるルールを記載すると、メッセージフィルタが適切に判断します



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

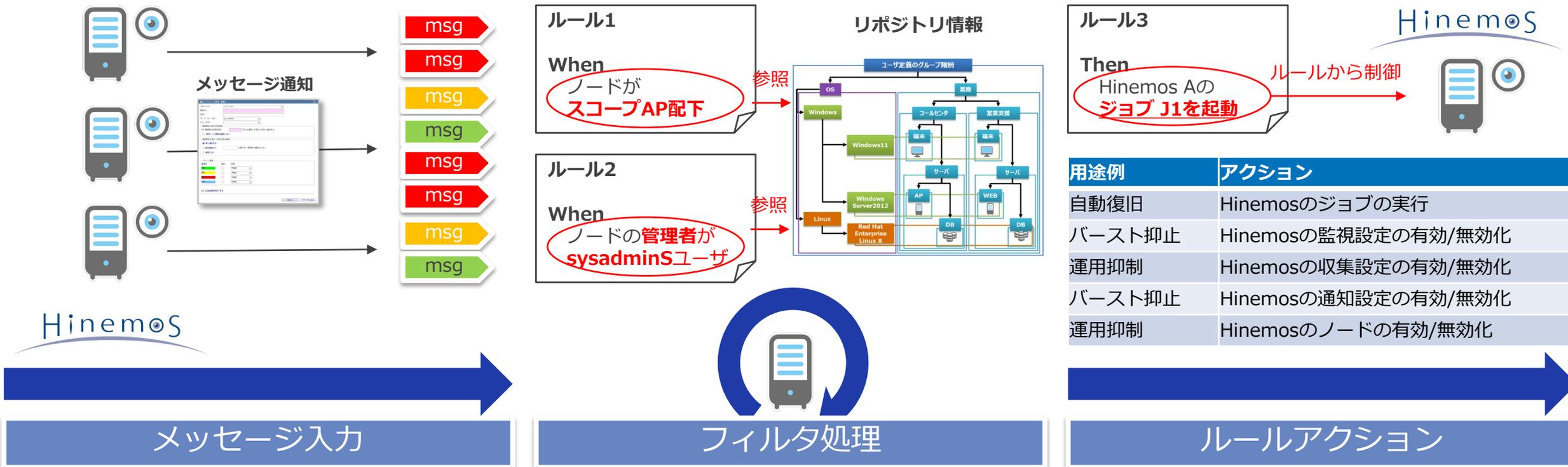
④Hinemosからのシームレスな導入

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

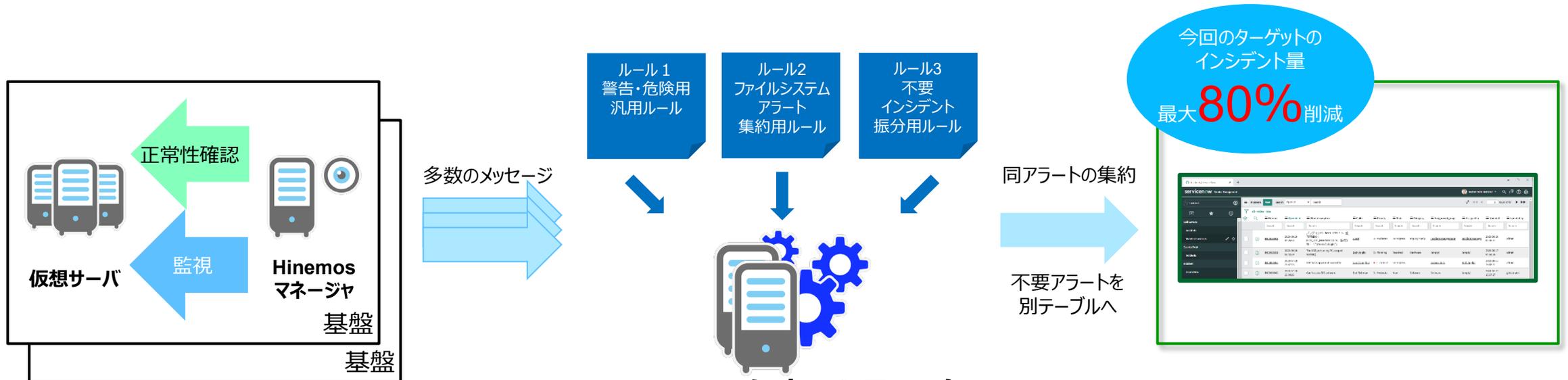
ルールの判定の結果のアクション(ルールアクション)でHinemosの各種操作が可能です



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

※ver.6.xのHinemosではHinemosメッセージ送信スクリプトをコマンド通知から呼び出す形式となります
 本資料ではそれを含めてメッセージ通知という表現で統一します
 © 2023 NTT DATA INTELLILINK Corporation

Hinemosメッセージフィルタ導入により、インシデント量の削減を実現



現行システムには一切影響なし
メッセージを転送する設定を
「追加」するのみ

たった3ルールのみで実現
簡易な3ルールで
フィルタリング・振り分けを実現

確認インシデント数を削減
確認する必要がある
インシデント数を削減

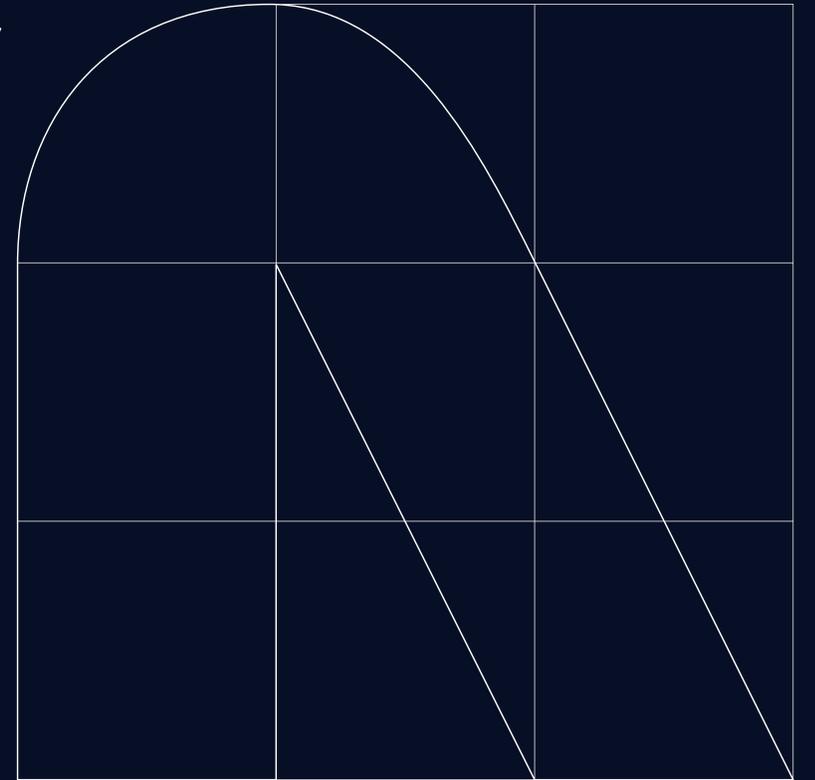
お客様の声

- 開発キットのサンプルルールによりルールの開発が容易でシミュレーション機能で試験が簡単だった
- 今後はServiceNowからHinemosメッセージフィルタの制御やより複雑なフィルタ処理を実現したい
- GUIやExcelライクでルール開発がもっと簡易に出来るなどの製品の発展を期待している

- 導入企業・団体様
株式会社NTTデータグループ様
- サーバ台数
145台 (Windows128台、Linux17台)
- Hinemosマネージャ導入台数
2台

03

AIドリブン運用のデモンストレーション



HinemosのAIドリブン運用 第1弾：運用自動化を実現するルール自動生成

ChatGPTを活用した運用自動化のインターフェースにより 専門知識が必要だった自動化の設定・ルール生成を、自然言語での指示で可能に

専門知識が不要

自動化の設定・ルールを自動生成



生成AIベースのインターフェース

イベント管理

自動化アクションの判断

運用自動化



生成AIによるルール生成

① 自然言語指示によるルール生成

生成AI

指示

NW機器#1でメッセージにLinkDownが含まれるイベントは通知しない

ルール生成

```
WHEN
  ○○○
THEN
  ○○○
```

デモにて
当ルールの作成をご紹介します。



Hinemos

② ルールのシミュレーション・試験

メッセージフィルタ開発キット

ルール①
ルール②
ルール③

```
WHEN
  ○○○
THEN
  ○○○
```

実行結果

様々な種類のイベント

ルール

アクション

NW機器の障害

LinkDown,LinkUp瞬断
時の重要度変更

NW機器の障害による
サーバ監視の到達不能

× 抑制

サーバのNIC障害

サーバのNIC障害による
サーバ上の監視不能

× 抑制

0時のジョブ正常終了

1時のジョブ正常終了

⋮

23時のジョブ正常終了

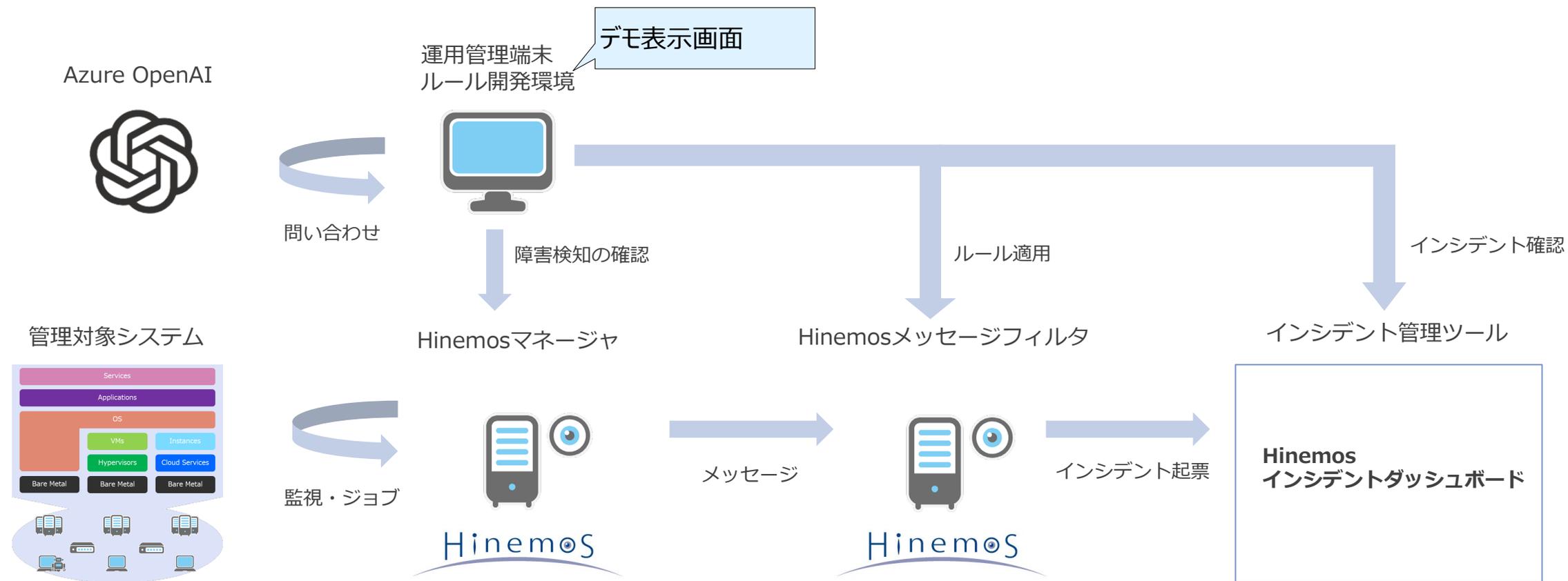
集約

0時~23時まで
全て正常終了

③ ルールの適用

デモ環境の対象システム構成

シンプルなデモの説明のため、インシデント起票するシナリオを対象にします



各コンポーネントは1つのサーバに導入が可能です。
デモでは分かりやすくするため、サーバを分けております。

シナリオ NW機器のLinkDown/Upの重要度の変更

NW機器のLinkDownの検知は重要。

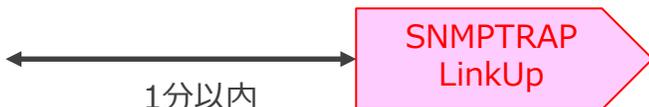
イベント



インシデント



ただし瞬断の場合は、インシデントとして起票するが重要度は低くしたい



1分以内



重要度を警告で起票する



1分以上



重要度を危険で起票する



デモの流れ

以下の流れでデモを行う。

- ①変更前の環境
- ②ルールの自動生成
- ③ルールのデプロイ
- ④変更後の環境

デモ環境の制約により実環境をお見せすることができません。
画像ベースでデモの投影となります。

①変更前の環境

LinkDown , LinkUpのイベントが検知されると、重要度"危険"でインシデント起票される

Hinemosマネージャ

監視履歴 リポジトリ メンテナンス 監視設定 スタートアップ

重要度	受信日時	アプリケーション	ファミリーID	スコープ	メッセージ	確認	監視詳細	確認ユー
危険	2023/10/12 08:37:09	LinkUp	NW-LB02	ロードバランサ#2	Agent Interface Up (linkUp Trap)interface 10	未確認	.1.3.6.1.6.3.1...	
危険	2023/10/12 08:36:57	LinkDown	NW-LB02	ロードバランサ#2	Agent Interface Down (linkDown Trap)interface 10	未確認	.1.3.6.1.6.3.1...	
危険	2023/10/12 08:34:51	LinkUp	NW-LB02	ロードバランサ#2	Agent Interface Up (linkUp Trap)interface 10	未確認	.1.3.6.1.6.3.1...	
危険	2023/10/12 08:32:05	LinkDown	NW-LB02	ロードバランサ#2	Agent Interface Down (linkDown Trap)interface 10	未確認	.1.3.6.1.6.3.1...	
危険	2023/10/12 08:29:05	BACH-A-Log	HINE-MGR01	HINE-MGR01	Oct 12 08:28:59 HINE-MGR01 root[44811]: [BCH-A] 定期ログの転送に失敗	未確認	.*BCH-A.*	
危険	2023/10/12 08:29:05	BACH-A-Log	HINE-MGR01	HINE-MGR01	Oct 12 08:28:54 HINE-MGR01 root[44811]: [BCH-A] tomcatサービスの起動に失敗	未確認	.*BCH-A.*	

Hinemosメッセージフィルタ

Hinemosインシデントダッシュボード

Pleasantier

新規作成 表示 管理 ヘルプ Administrator 検索

トップ > Hinemosインシデントダッシュボード >

フィルタ: リセット 未完了 自分 期限が近い 遅延 期限超過 状況 管理者 担当者

監視項目ID プラグインID 監視詳細 ファミリティID ファシリティ名

重要度 出力日時 検索

集計: 件数 12

ID	監視項目ID	ファミリーID	重要度	インシデント名/概要	出力日時	期限	状況
15	NW-TRAP02	NW-LB02	危険	LinkUp Agent Interface Up (linkUp Trap) interface 10	2023/10/12 08:37	23/10/13 金 1日後	未
14	NW-TRAP01	NW-LB02	危険	LinkDown Agent Interface Down (linkDown Trap) interface 10	2023/10/12 08:36	23/10/13 金 1日後	未
11	NW-TRAP02	NW-LB02	危険	LinkUp Agent Interface Up (linkUp Trap) interface 10	2023/10/12 08:34	23/10/13 金 1日後	未
10	NW-TRAP01	NW-LB02	危険	LinkDown Agent Interface Down (linkDown Trap) interface 10	2023/10/12 08:32	23/10/13 金 1日後	未

① 変更前の環境

○実現したいこと

LinkDownが検知後、LinkUpが60秒以内に検知した場合は重要度を“警告”のインシデント起票したい。

監視履歴

スコープ: , 検索開始時刻:ALL(マネージャ1)

重要度	受信日時	アプリ
危険	2023/10/12 08:37:09	Link
危険	2023/10/12 08:36:57	Link
危険	2023/10/12 08:34:51	Link
危険	2023/10/12 08:32:05	Link
危険	2023/10/12 08:29:05	BAC
危険	2023/10/12 08:29:05	BAC

以下のルールの追加を行う。

LinkDownのイベントが検知してから

- ① 60秒間待機し、LinkUpのイベントが検知されない場合
- LinkDownのインシデントを起票
- ② 60秒以内にLinkUpが検知された場合は
- LinkDownの重要度を警告
- LinkUpの重要度を警告 に変更しインシデントを起票
- ③ 60秒以上経過し、Linkupイベントが出力された場合
- LinkUpのインシデントを起票

Hinemosメッセージフィルタ

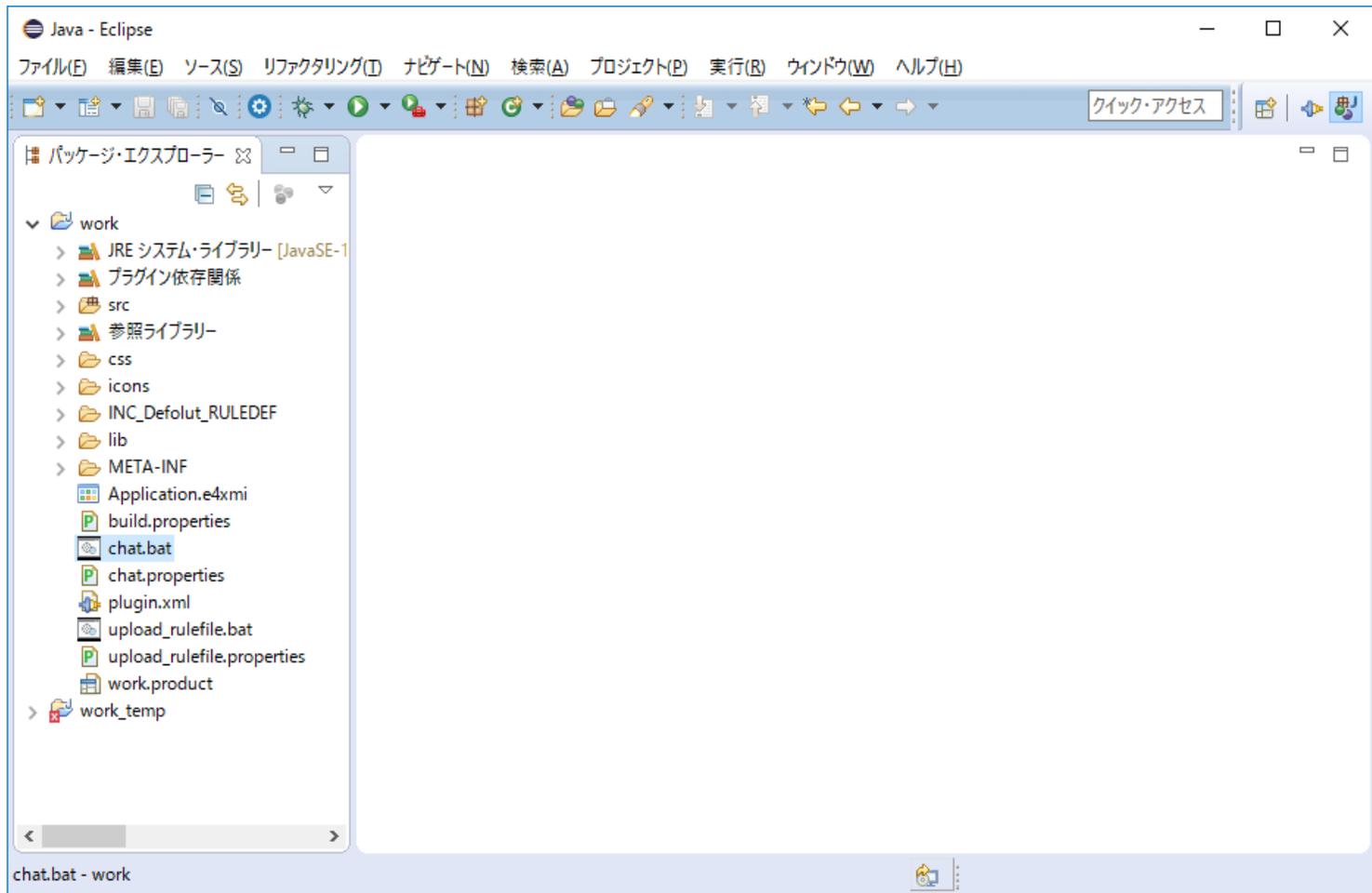
検索結果: 件数 12

ID	監視項目ID	ファンクティID	重要度	インシデント名/概要	出力日時	期限	状況
15	NW-TRAP02	NW-LB02	危険	LinkUp Agent Interface Up (linkUp Trap) interface 10	2023/10/12 08:37	2023/10/13 金 1日後	未
14	NW-TRAP01	NW-LB02	危険	LinkDown Agent Interface Down (linkDown Trap) interface 10	2023/10/12 08:36	2023/10/13 金 1日後	未
11	NW-TRAP02	NW-LB02	危険	LinkUp Agent Interface Up (linkUp Trap) interface 10	2023/10/12 08:34	2023/10/13 金 1日後	未
10	NW-TRAP01	NW-LB02	危険	LinkDown Agent Interface Down (linkDown Trap) interface 10	2023/10/12 08:32	2023/10/13 金 1日後	未
9	BATCH-LOG001	HINE-MGR01	危険	BACH-A-Log	2023/10/12 08:29	2023/10/13 金 1日後	未

② ルールの自動作成

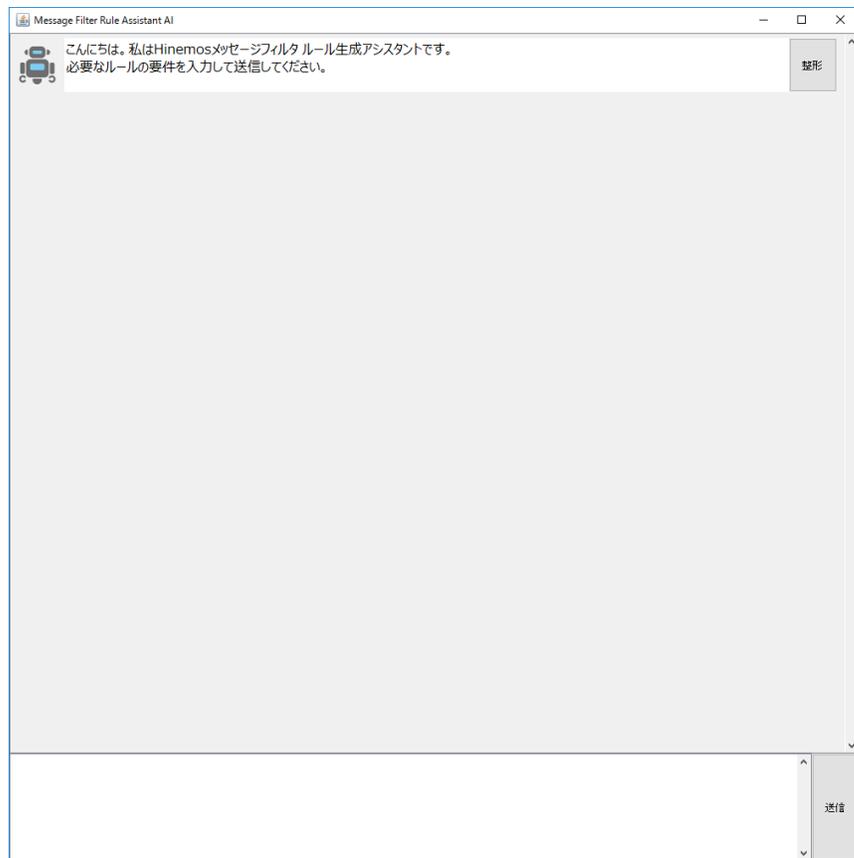
生成AIを利用しルール（driプログラム）の作成を行う。

Hinemosメッセージフィルタ 開発キット（運用端末）

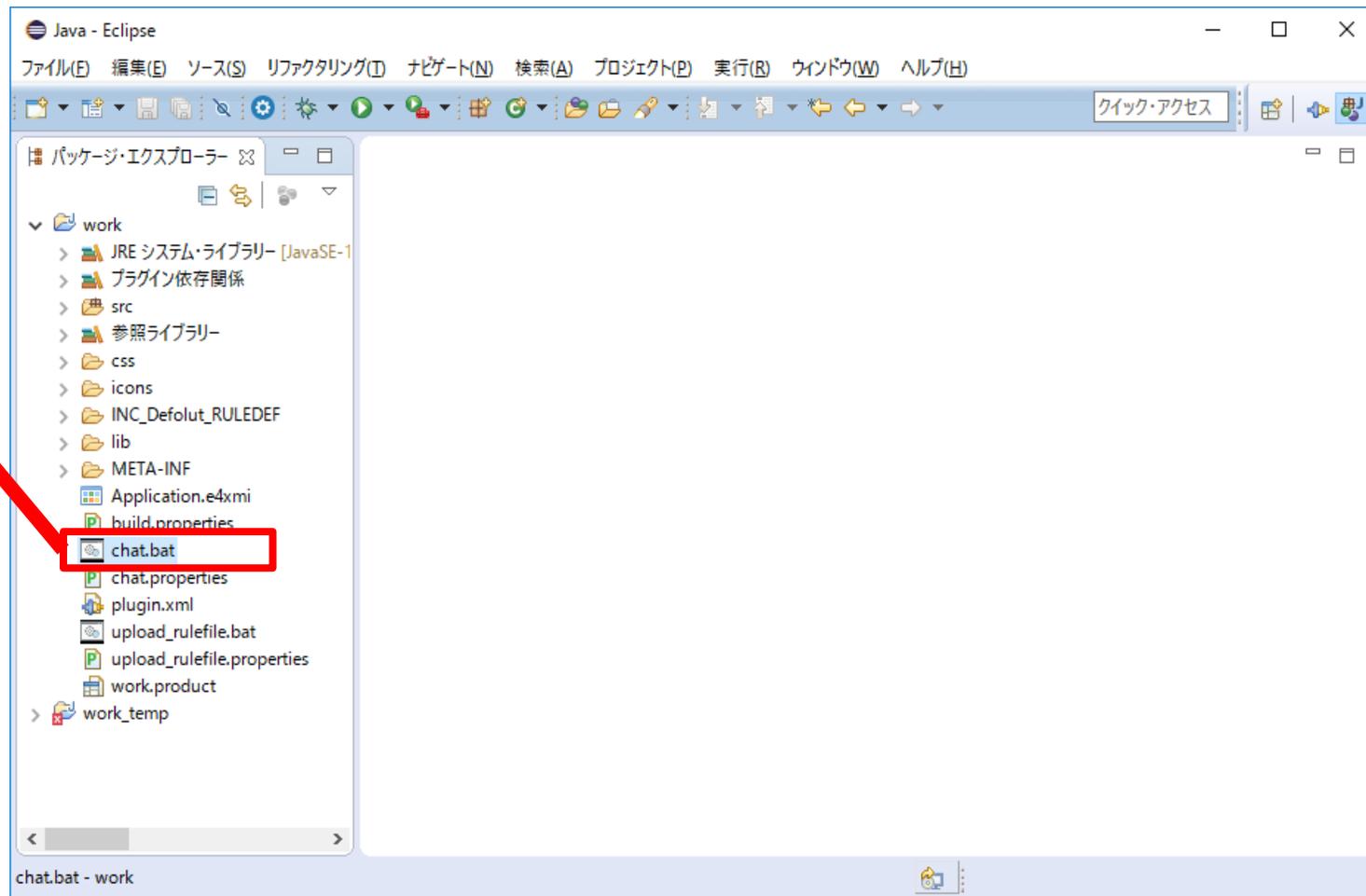


②ルール of 自動作成

開発キットより、生成AIに問い合わせるフォームを開く。



Hinemosメッセージフィルタ 開発キット (運用端末)



②ルールの自動作成

問い合わせフォームより、ルール要件を記入し生成AIに送信する。

Hinemosメッセージフィルタ 開発キット (運用端末)

The screenshot shows a web-based interface for creating message filter rules. It is divided into two main sections: 'Rule Requirements' (ルール要件) and 'Rules to be Implemented' (実現したいルール). A red arrow points from the implementation steps back to the requirements, indicating that the implementation is based on the specified requirements. A red box highlights the 'Rule Requirements' section with the text '※ルール要件の記入' (Please enter the rule requirements).

Message Filter Rule Assistant AI
こんにちは。私はHinemosメッセージフィルタルール生成に必要なルールの要件を入力して送信してください。

○ルール要件

- ・前提条件
 - メッセージにLinkDownを含むイベントを\$m1とする。
 - メッセージにLinkUpを含むイベントを\$m2とする。
- ・要件
 - 処理優先度3
\$m1を受信後、60秒間待機し"インシデント連携\$m1"コマンドを実行する。
 - 処理優先度2
\$m2を受信後、60秒前まで遡り\$m1を受信しているかを確認し、受信している場合は以下を行う。
 - \$m1の重要度に"警告"を設定する。
 - \$m2の重要度に"警告"を設定する。
 - "インシデント連携\$m1"コマンドを実行する。
 - "インシデント連携\$m2"コマンドを実行する。
 - 処理優先度1
イベント受信後、"インシデント連携\$m2"コマンドを実行する。

○実現したいルール

LinkDownのイベントが検知してから

- ① 60秒間待機し、LinkUpのイベントが検知されない場合
 - LinkDownのインシデントを起票
- ② 60秒以内にLinkUpが検知された場合は
 - LinkDownの重要度を警告
 - LinkUpの重要度を警告 に変更しインシデントを起票
- ③ 60秒以上経過し、Linkupイベントが出力された場合
 - LinkUpのインシデントを起票

※ルール要件の記入

work_temp

chat.bat - work

② ルールの自動作成

生成AIからのルール回答

Hinemosメッセージフィルタ 開発キット (運用端末)

Message Filter Rule Assistant AI

こんにちは。私はHinemosメッセージフィルタルール生成アシスタントです。
必要なルールの要件を入力して送信してください。

・前提条件
メッセージにLinkDownを含むイベントを\$m1とする。
メッセージにLinkUpを含むイベントを\$m2とする。

・要件

- 処理優先度3
\$m1を受信後、60秒間待機し"インシデント連携\$m1"コマンドを実行する。
- 処理優先度2
\$m2を受信後、60秒前まで遡り\$m1を受信しているかを確認し、受信している場合は以下を行う。
 - \$m1の重要度に"警告"を設定する。
 - \$m2の重要度に"警告"を設定する。
 - "インシデント連携\$m1"コマンドを実行する。
 - "インシデント連携\$m2"コマンドを実行する。
- 処理優先度1
イベント受信後、"インシデント連携\$m2"コマンドを実行する。

```
rule "Priority 3: Single Event Notification"
salience 3
timer (int: 60s)
when
  $m1: Message(text contains "linkDown")
then
  action.cmd("インシデント連携$m1");
  delete($m1);
end

rule "Priority 2: Notification within 1 minute"
salience 2
When
  $m2: Message(text contains "linkUp")
  $m1: Message(text contains "linkDown" ,this before[0s, 60s] $m2)
then
  $m1.setPriority("警告");
  $m2.setPriority("警告");
  action.cmd("インシデント連携$m1");
  action.cmd("インシデント連携$m2");
  delete($m1);
  delete($m2);
end

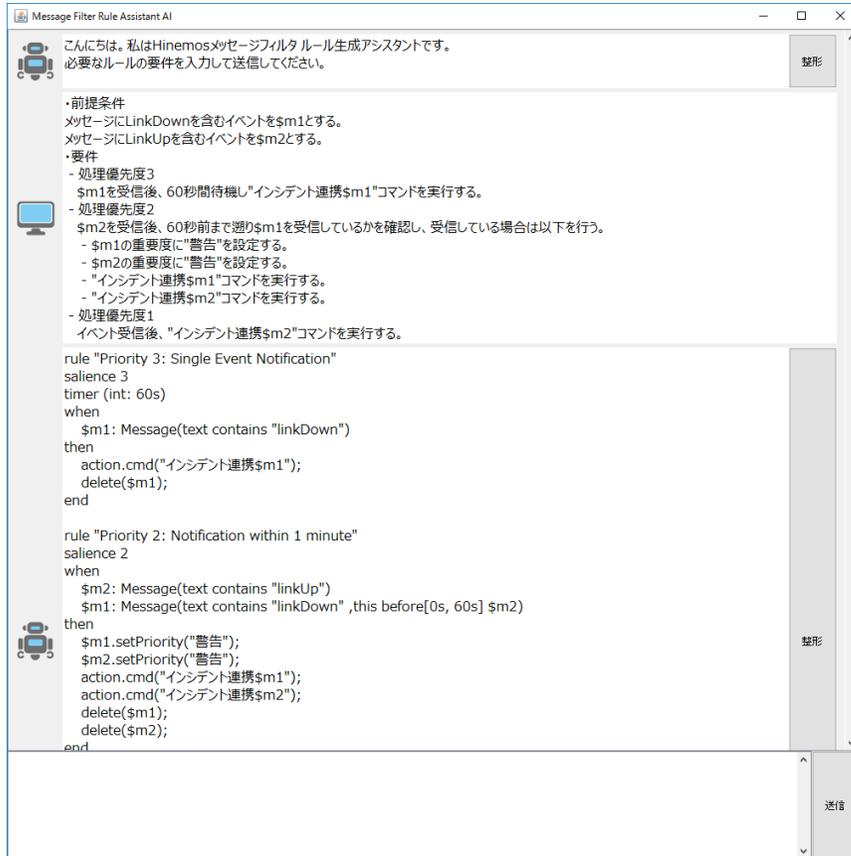
rule "Priority 1: Single Event Notification"
salience 1
when
  $m2: Message(text contains "linkUp")
then
  action.cmd("インシデント連携$m2");
  delete($m2);
end
```

○生成AI問い合わせ内容

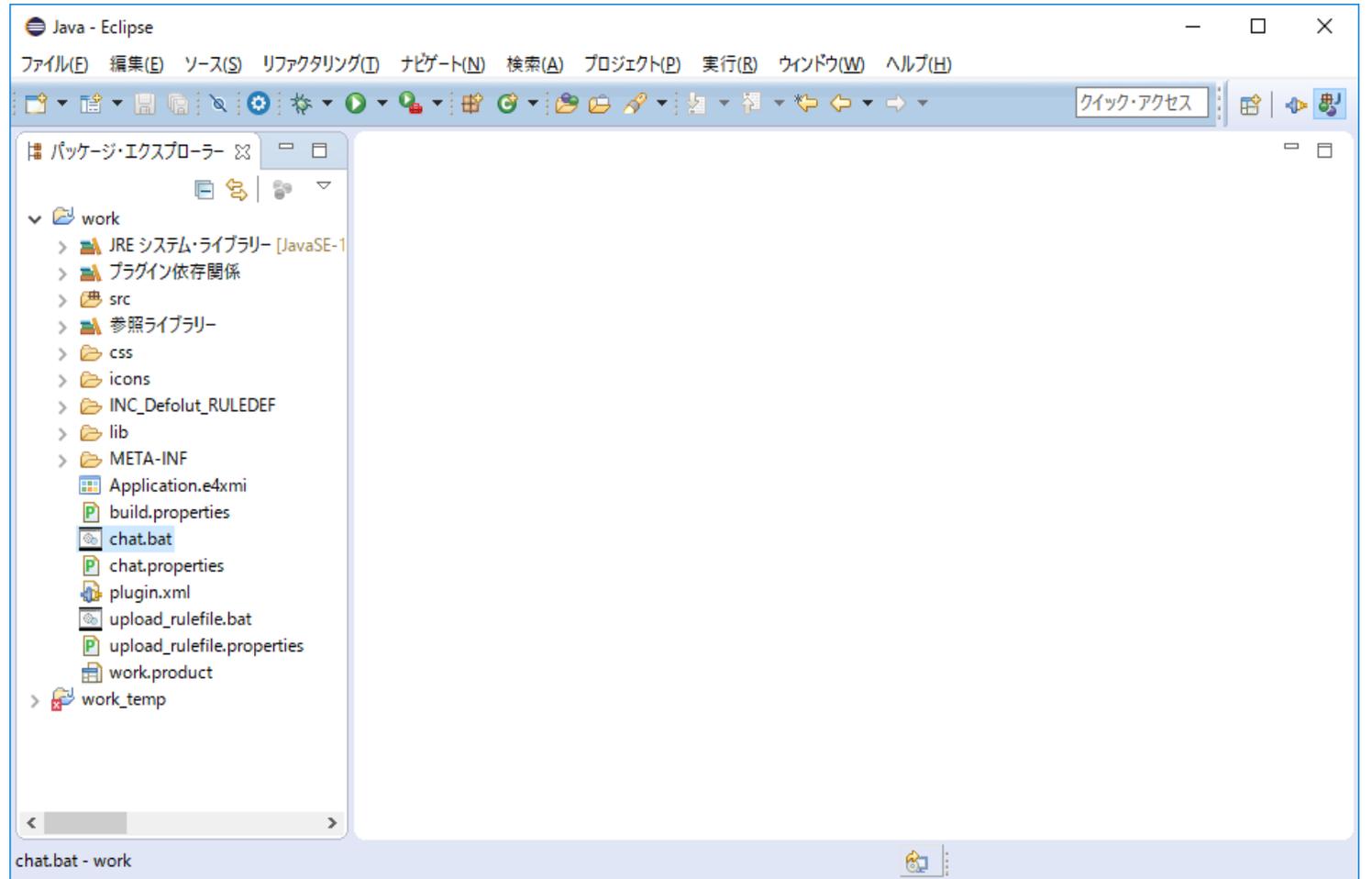
- 処理優先度3
\$m1を受信後、60秒間待機し"インシデント連携\$m1"コマンドを実行する。
- 処理優先度2
\$m2を受信後、60秒前まで遡り\$m1を受信しているかを確認し、受信している場合は以下を行う。
 - \$m1の重要度に"警告"を設定する。
 - \$m2の重要度に"警告"を設定する。
 - "インシデント連携\$m1"コマンドを実行する。
 - "インシデント連携\$m2"コマンドを実行する。
- 処理優先度1
イベント受信後、"インシデント連携\$m2"コマンドを実行する。

③ ルールのデプロイ

回答されたルールを.drl拡張子ファイル化し、Hinemosメッセージフィルタにアップロードを行う。

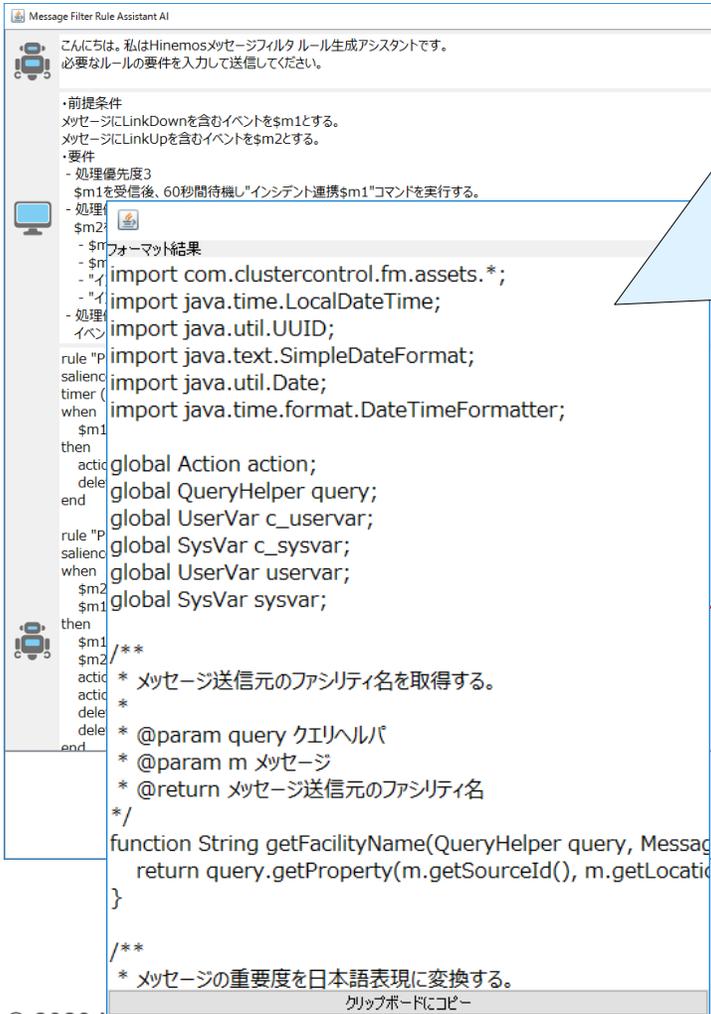


Hinemosメッセージフィルタ 開発キット (運用端末)



③ ルールのデプロイ

回答されたルールの補正を行う。



```
import com.clustercontrol.fm.assets.*;
import java.time.LocalDateTime;
import java.util.UUID;
import java.text.SimpleDateFormat;
import java.util.Date;
import java.time.format.DateTimeFormatter;
```

```
global Action action;
global QueryHelper query;
global UserVar c_uservar;
global SysVar c_sysvar;
global UserVar uservar;
global SysVar sysvar;
```

```
/**
 * メッセージ送信元のファシリティ名を取得する。
 */
```

```
@param query クエリヘルパ
@param m メッセージ
@return メッセージ送信元のファシリティ名
```

```
function String getFacilityName(QueryHelper query, Message m) {
    return query.getProperty(m.getSourceId(), m.getLocation(),
    "FACILITY_NAME");
}
```

```
/**
 * メッセージの重要度を日本語表現に変換する。
 * この日本語表現は、通知用の置換文字列「#[PRIORITY_JP]」で取得できるテキストに
 * 相当する。
 */
```

```
@param m メッセージ
@return 重要度の日本語表現
```

```
function String getPriorityText(Message m) {
    String priorityText = m.getPriority();
    if (priorityText == null) {
        return "不明";
    } else if (priorityText.equals("critical")||priorityText.equals("危険")) {
        return "危険";
    } else if (priorityText.equals("warning")||priorityText.equals("警告")) {
        return "警告";
    } else if (priorityText.equals("info")||priorityText.equals("情報")) {
        return "情報";
    } else if (priorityText.equals("unknown")||priorityText.equals("不明")) {
        return "不明";
    } else {
        return "不明";
    }
}
```

```
rule "Priority 3: Single Event Notification"
salience 3
timer (int: 60s)
when
    $m1: Message(text contains "linkDown")
then
    action.cmd(
        "issues_id_m1",
        "issues_code_m1",
        "python3.6 /opt/hinemos_itil/integration-pleasanter/hinemos-
        setting/scripts/issue.py "
        + "$m1.getAttribute("monitorId") + "$m1 "
        + "$m1.getAttribute("pluginId") + "$m1 "
        + "$m1.getAttribute("monitorDetail") + "$m1 "
        + "$m1.getLocation() + "$m1 "
        + "$m1.getFacilityName(query, $m1) + "$m1 "
        + "$m1.getPriorityText($m1) + "$m1 "
        + "$m1.getAttribute("application") + "$m1 "
        + "$m1.getText() + "$m1 "
        + "$m1.getAttribute("messageOrg") + "$m1 "
        + "$m1 "
        + "$m1.getGeneratedTime().format(DateTimeFormatter.ofPattern("yyyy/MM/dd
        HH:mm:ss")) + "$m1 " ,
        10000);

    delete($m1);
end

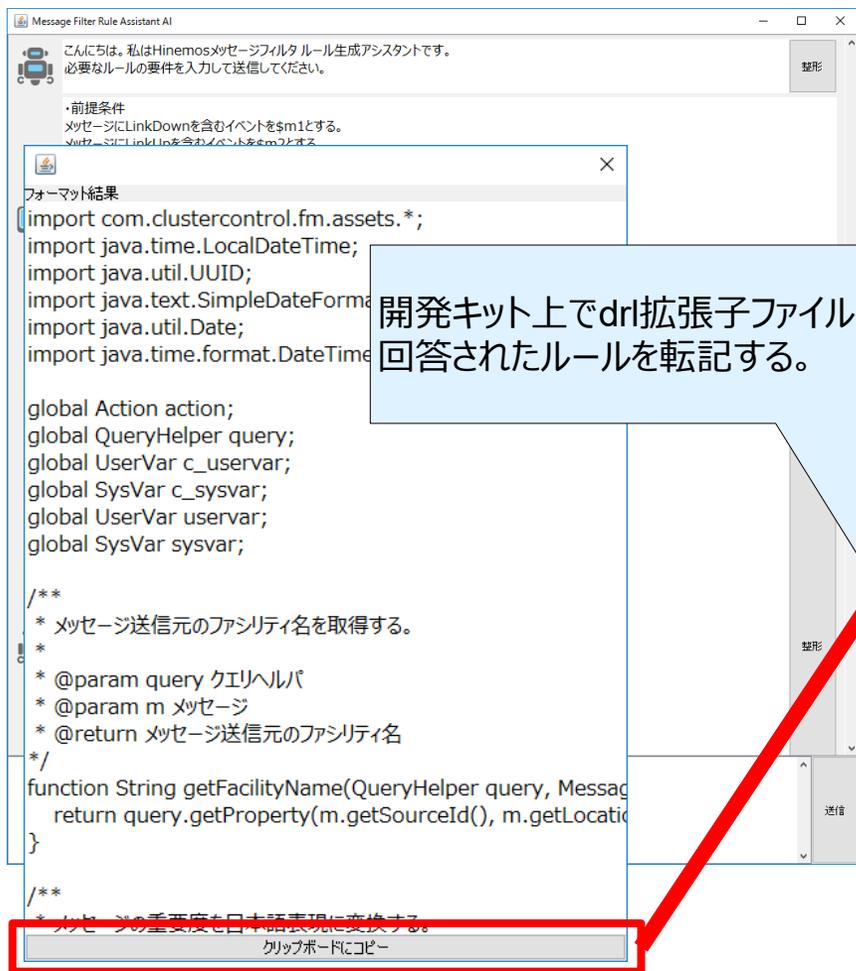
rule "Priority 2: Notification within 1 minute"
salience 2
when
    $m2: Message(text contains "linkUp")
    $m1: Message(text contains "linkDown" ,this before[0s, 60s] $m2)
then
    $m1.setPriority("警告");
    $m2.setPriority("警告");
    action.cmd(
        "issues_id_m1",
        "issues_code_m1",
        "python3.6 /opt/hinemos_itil/integration-pleasanter/hinemos-
        setting/scripts/issue.py "
        + "$m1.getAttribute("monitorId") + "$m1 "
        + "$m1.getAttribute("pluginId") + "$m1 "
        + "$m1.getAttribute("monitorDetail") + "$m1 "
        + "$m1.getLocation() + "$m1 "
        + "$m1.getFacilityName(query, $m1) + "$m1 "
        + "$m1.getPriorityText($m1) + "$m1 "
        + "$m1.getText() + "$m1 "
        + "$m1.getAttribute("messageOrg") + "$m1 "
        + "$m1 "
        + "$m1.getGeneratedTime().format(DateTimeFormatter.ofPattern("yyyy/MM/dd
        HH:mm:ss")) + "$m1 " ,
        10000);

    delete($m1);
end

~ (省略) ~
```

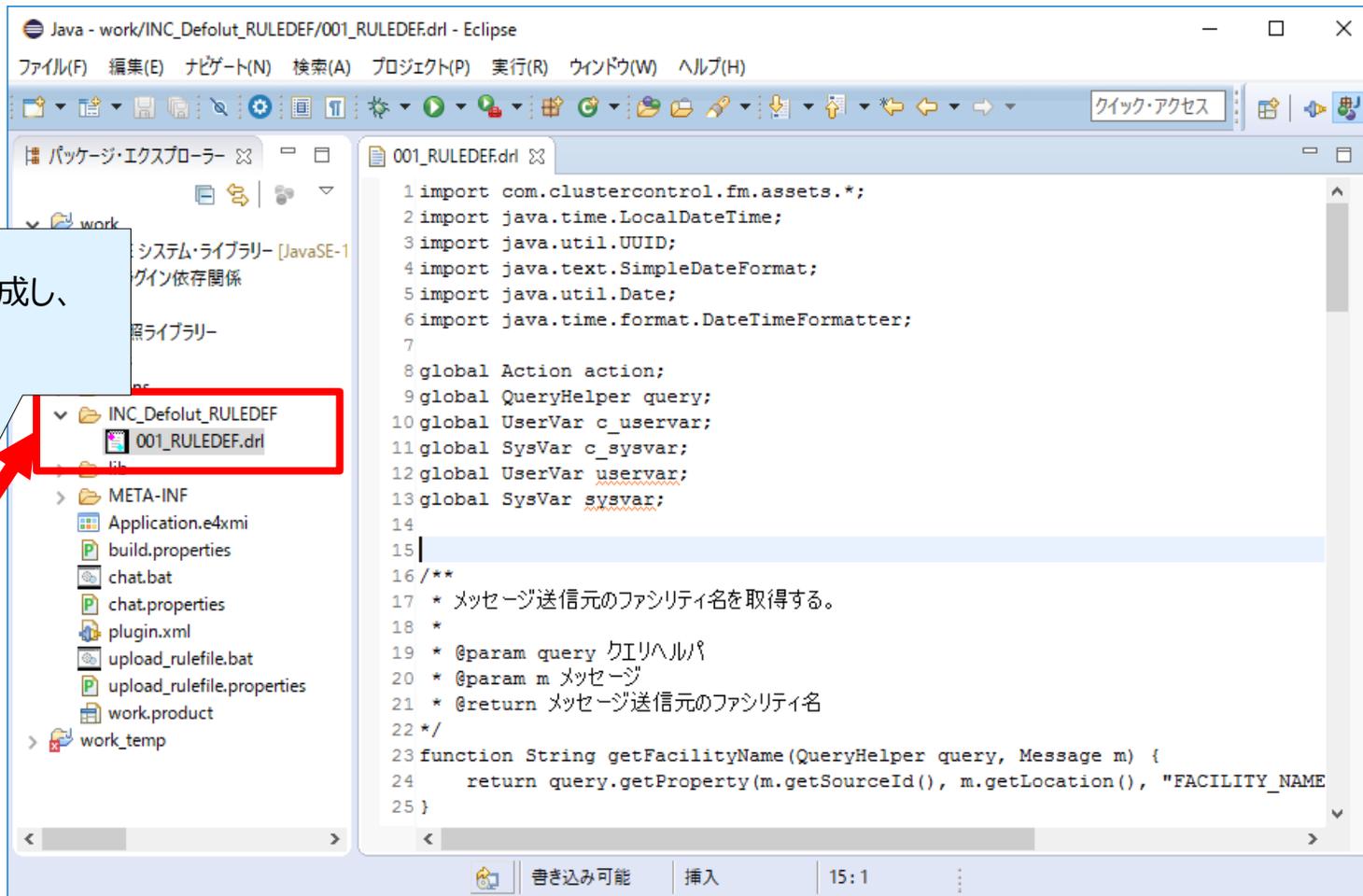
③ ルールのデプロイ

回答されたルールを開発キットへ転記



開発キット上でdrl拡張子ファイルを作成し、
回答されたルールを転記する。

Hinemosメッセージフィルタ 開発キット (運用端末)



③ ルールのデプロイ

Hinemosメッセージフィルタに作成したルールをアップロードする。

Hinemosメッセージフィルタ 開発キット (運用端末)

The screenshot shows the Eclipse IDE interface. On the left, the Package Explorer displays a project structure with a folder named 'INC_Defolut_RULEDEF' containing a file '001_RULEDEF.drl'. Below it, a folder 'work' contains various files, including 'upload_rulefile.bat' which is highlighted with a red box. On the right, the editor shows the content of '001_RULEDEF.drl', which is a Drools rule definition. The code includes imports for classes like 'com.clustercontrol.fm.assets.*', 'java.time.LocalDateTime', 'java.util.UUID', 'java.text.SimpleDateFormat', 'java.util.Date', and 'java.time.format.DateTimeFormatter'. It also defines global variables for 'Action', 'QueryHelper', 'UserVar', and 'SysVar', and a function 'getFacilityName' that returns a facility name from a message.

```
1 import com.clustercontrol.fm.assets.*;
2 import java.time.LocalDateTime;
3 import java.util.UUID;
4 import java.text.SimpleDateFormat;
5 import java.util.Date;
6 import java.time.format.DateTimeFormatter;
7
8 global Action action;
9 global QueryHelper query;
10 global UserVar c_uservar;
11 global SysVar c_sysvar;
12 global UserVar uservar;
13 global SysVar sysvar;
14
15
16 /**
17  * メッセージ送信元のファシリティ名を取得する。
18  *
19  * @param query クエリヘルパ
20  * @param m メッセージ
21  * @return メッセージ送信元のファシリティ名
22  */
23 function String getFacilityName(QueryHelper query, Message m) {
24     return query.getProperty(m.getSourceId(), m.getLocation(), "FACILITY_NAME");
25 }
```

.batファイルの実行で、アップロードが完了

④ 変更後の環境

Hinemosマネージャ

監視履歴 スタートアップ

監視履歴[イベント] ×

スコープ: , 検索開始時刻:ALL(マネージャ1)

重要度	受信日時	アプリケーション	ファシリティID	スコープ	メッセージ	監視
危険	2023/10/12 11:04:57	LinkUp	NW-LB02	ロードバランサ#2	Agent Interface Up (linkUp Trap)interface 10	NW
危険	2023/10/12 11:04:42	LinkDown	NW-LB02	ロードバランサ#2	Agent Interface Down (linkDown Trap)interface 10	NW
危険	2023/10/12 08:28:59	SLA-LOG	HINE-MGR01	HINE-MGR01	Oct 12 08:28:59 HINE-MGR01 root[44814]: [BCH-A] 定期ログの転送に失敗	BAT
危険	2023/10/12 08:28:54	SLA-LOG	HINE-MGR01	HINE-MGR01	Oct 12 08:28:54 HINE-MGR01 root[44811]: [BCH-A] tomcatサービスの起動に失敗	BAT

LinkDownのイベント検知後
60秒以内に
LinkUpのイベントを検知している。

Hinemosインシデントダッシュボード

Pleasantier

+ 新規作成

▶ 表示

✳ 管理

? ヘルプ

Administrator

検索

トップ > Hinemosインシデントダッシュボード >

フィルタ: リセット 未完了 自分 期限が近い

プラグインID 監視詳細

検索

集計: 件数 23

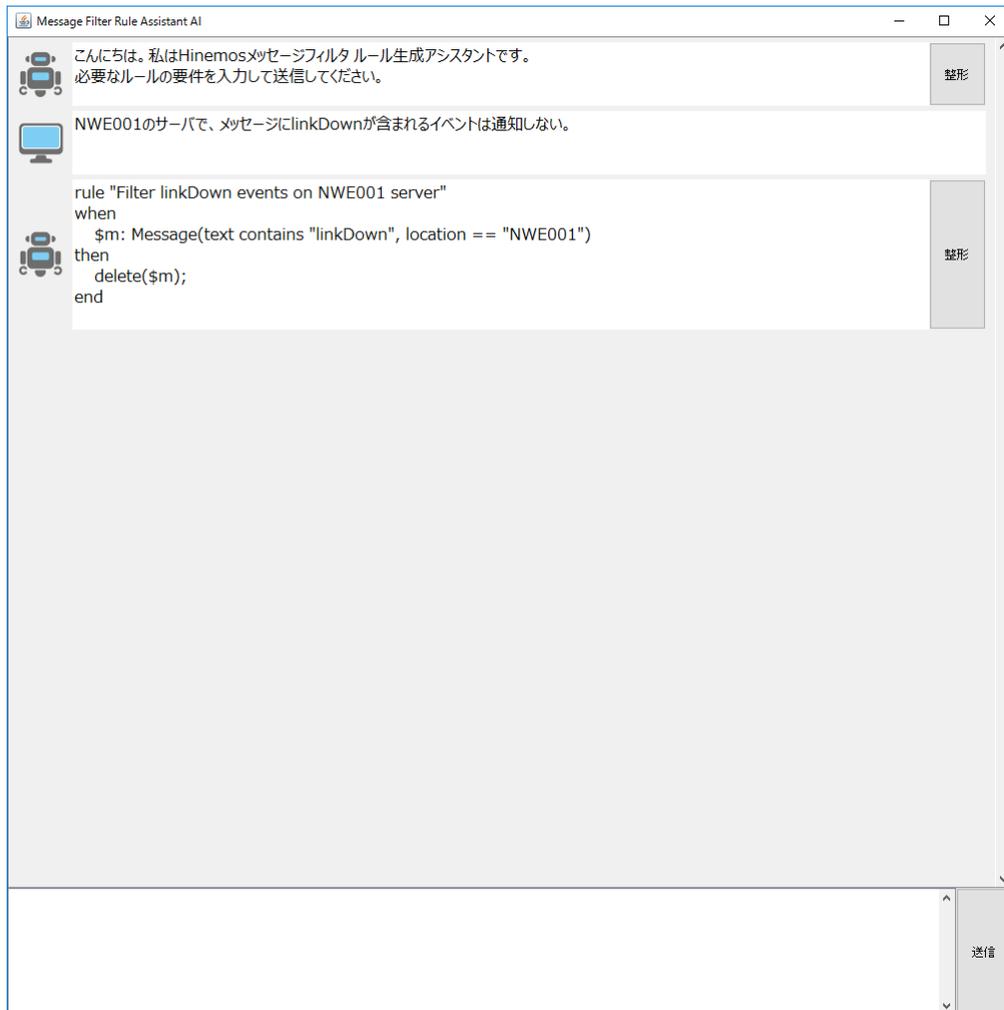
ID	監視項目ID	ファシリティID	重要度	インシデント名/概要	出力日時	期限	状況
29	NW-TRAP02	NW-LB02	警告	LinkUp Agent Interface Up (linkUp Trap) interface 10	2023/10/12 11:04	2023/10/13 金 1日後	未
28	NW-TRAP01	NW-LB02	警告	LinkDown Agent Interface Down (linkDown Trap) interface 10	2023/10/12 11:04	2023/10/13 金 1日後	未

重要度「警告」でインシデント起票されている。

Hinemosメッセージフィルタ

ルール作成の紹介

1つのルール生成の場合、より自然言語に近い形式での問い合わせが可能



○想定するユースケース

NWE001が故障発生し、1週間後に機器交換が予定。
機器交換までの間はLinkDownのイベントを通知しない。



○生成AIへの問い合わせ

NWE001のサーバで、メッセージにlinkDownが含まれるイベントは通知しない。

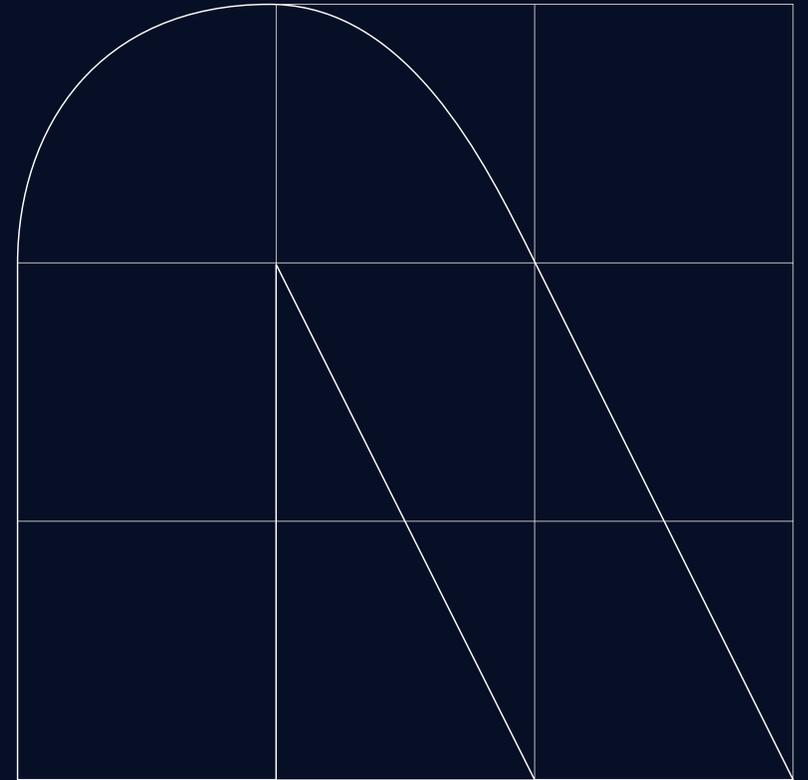


○生成AIからの回答（ルール）

```
rule "Filter linkDown events on NWE001 server"
when
  $m: Message(text contains "linkDown", location == "NWE001")
then
  delete($m)
end
```

04

まとめ



1. 生成AIを使ったAIドリブン運用の世界観

AIドリブン運用により、人手作業では成し得なかった、**飛躍的な生産性の向上**が見込める

2. Hinemosを使った運用自動化

- イベント連携による運用自動化を実現する、Hinemosメッセージフィルタをご紹介
- Hinemosメッセージフィルタを活用することでインシデントを**"80%"**を削減した事例のご紹介

3. AIドリブン運用のデモンストレーション

- Hinemosが取り込むAIドリブン運用により、**運用自動化を推進**することが見込める。
- 生成AI（ChatGPT）活用した運用自動化のインターフェースを**2024年4月目途に製品化予定**

- Hinemosメッセージフィルタを利用することで、不要なインシデントの削減が可能。
- 更に生成AIを活用することでルールの生成効率化を行い、運用自動化を推進。

AIドリブン運用の取り組みを行いたいお客様は、ご連絡お待ちしております。



NTT DATA