202206版

NTT DATA
Trusted Global Innovator

Hinemosメッセージフィルタのご紹介

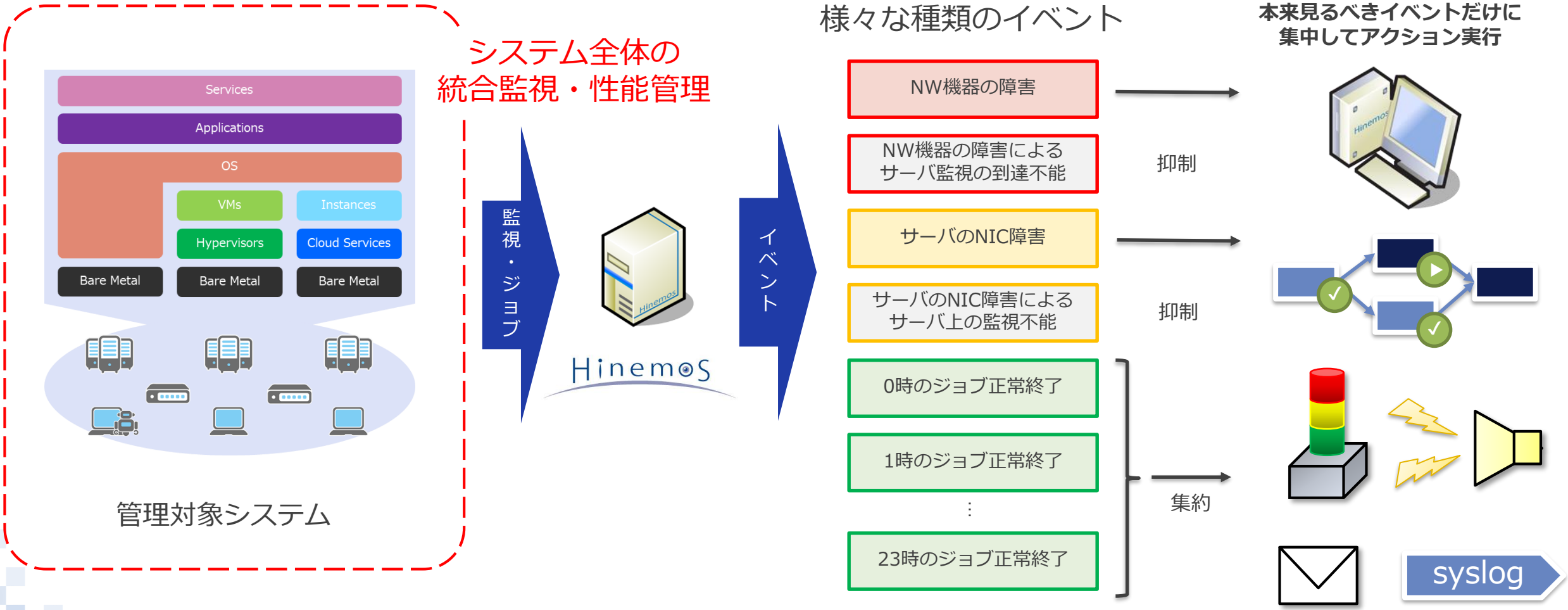
NTTデータ先端技術株式会社

1. 背景
2. 機能概要
3. 機能詳細
4. ユースケース
5. 動作要件・サポートサイクル
6. ご相談・お問い合わせ

背景

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

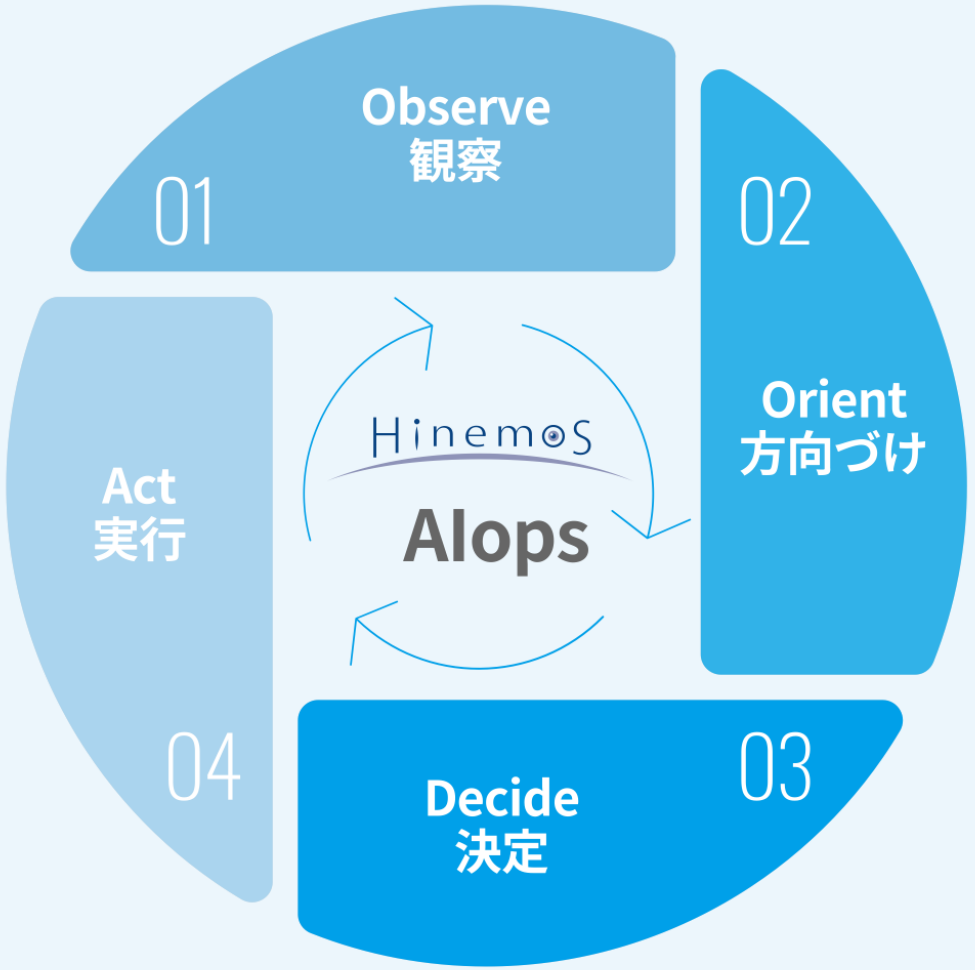


HinemosのAIOpsの取り組み

運用ログを分析活用
- 予兆検知によるシステム障害未然対応
- 故障原因の分析
- 性能分析とキャパプラ

監視自動設定/オートチューニング
- サーバ負荷分析による閾値自動設定
- ログ監視の自動キーワード設定

障害対応の迅速化 / 自動化
- システム障害の自動復旧
- リコメンドによる未知の障害対応



インテリジェントなアラート
- 不要なイベントを抑制
- 関連メッセージを集約

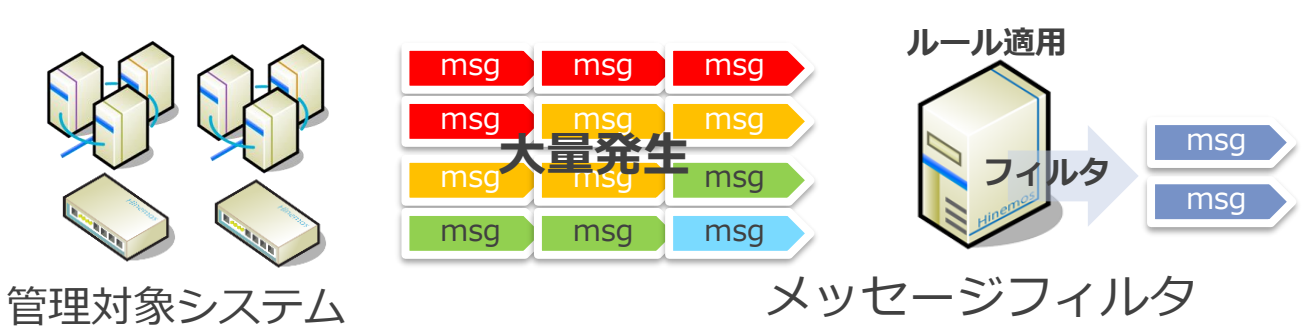
ナレッジ活用
- インシデント分析による傾向把握
- フィードバックによるナレッジの充実

対応判断の支援
- 過去類似インシデントをオペレータに提示
- エンジニアの呼び出し要否を判断

機能概要

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

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

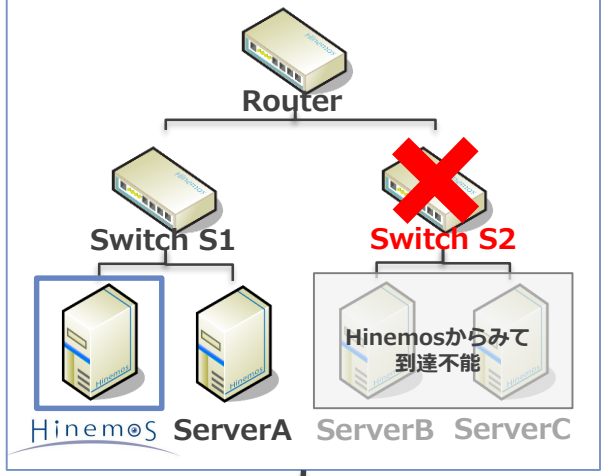
④Hinemosからのシームレスな導入

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

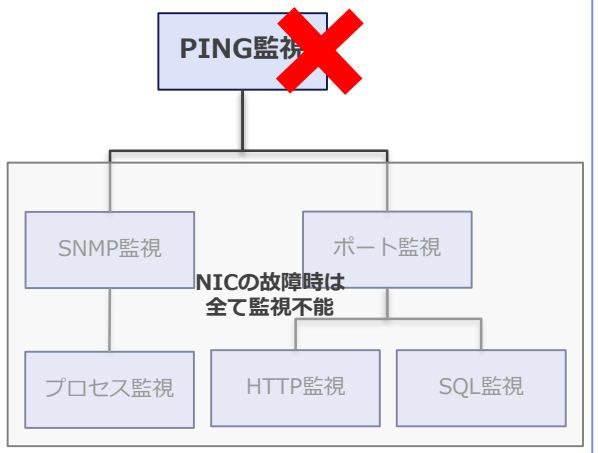
不要なメッセージの抑制

スイッチの故障



Switch S2 障害 / Server B 不明 / Server C 不明

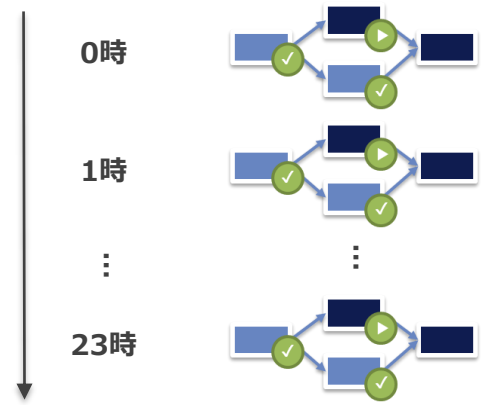
NICの故障



PING / SNMP / プロセス / ポート / HTTP / SQL

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



0時正常 / 1時正常 … 11時正常
12時正常 / 12時正常 … 23時正常

ディスクエラー

/var/log/messages

```
Feb 16 17:18:19 serv Disk I/O ERROR
Feb 16 17:18:19 serv Disk I/O ERROR
Feb 16 17:18:19 serv Disk I/O ERROR
Feb 16 17:18:20 serv Disk I/O ERROR
Feb 16 17:18:20 serv Disk I/O ERROR
Feb 16 17:18:20 serv Disk I/O ERROR
Feb 16 17:18:21 serv Disk I/O ERROR
```

I/O ERROR / I/O ERROR … I/O ERROR
I/O ERROR / I/O ERROR … I/O ERROR

本質的なイベントへのフィルタリング（抑制・集約）

Switch S2 障害

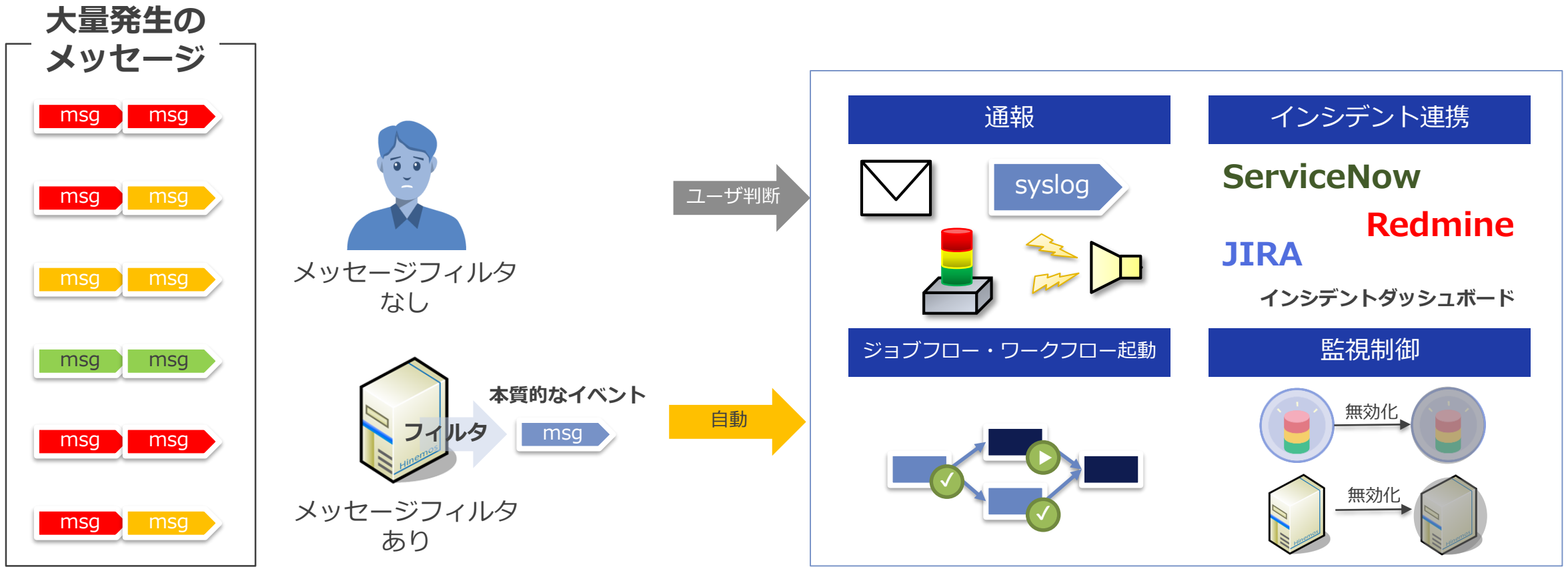
PING

〇月〇日全て正常

I/O ERROR
〇〇時から〇〇時まで
1203回発生

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

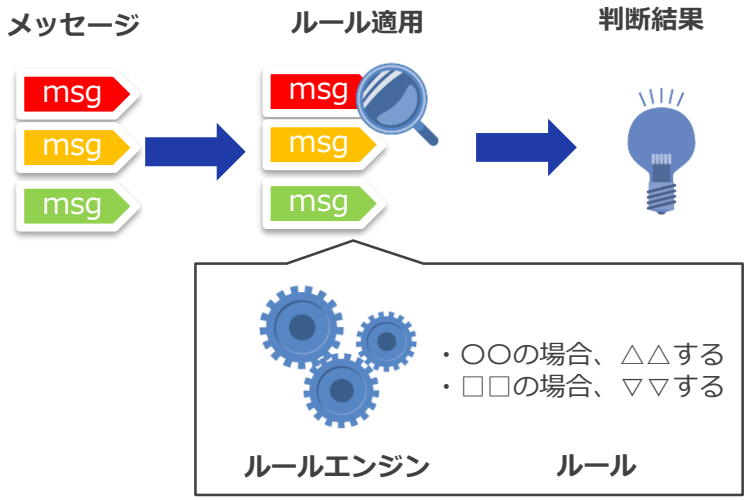


③ルールベースの条件指定

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

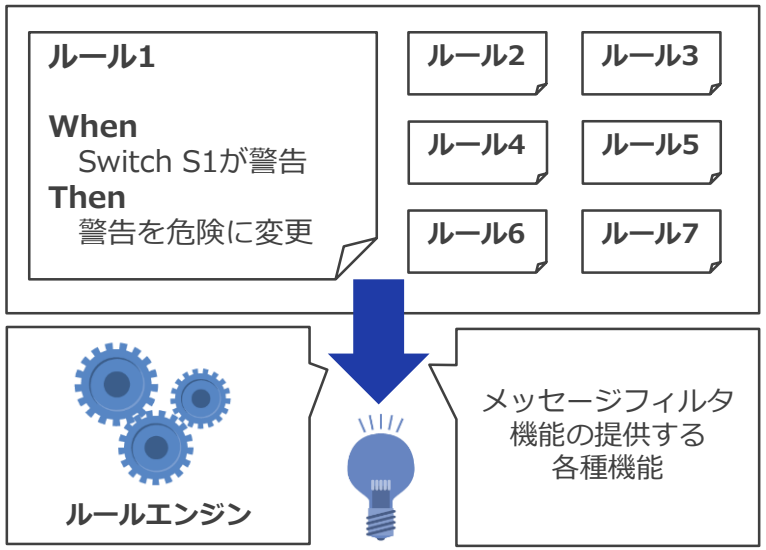
ルールエンジンとは、与えられたルールに従って判断を行う機能です



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

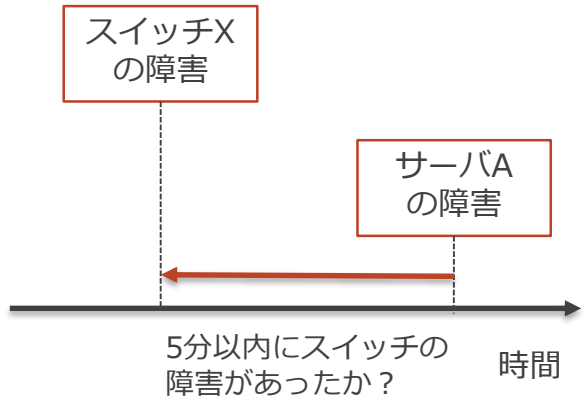
When/Thenからなるルールを記載すると、メッセージフィルタが適切に判断します



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

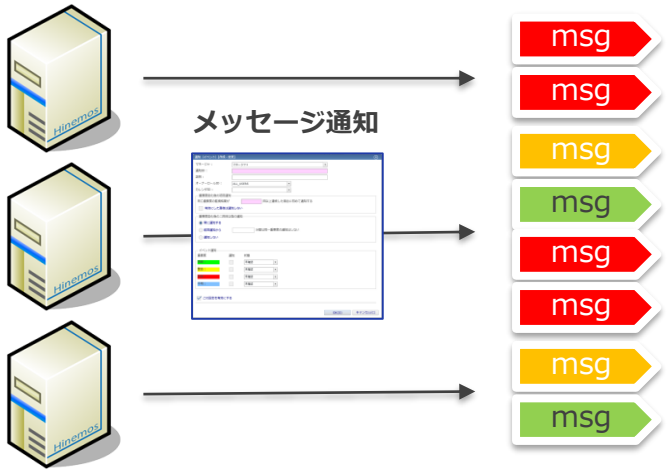


Droolsが提供するCEP機能をそのまま使用できます

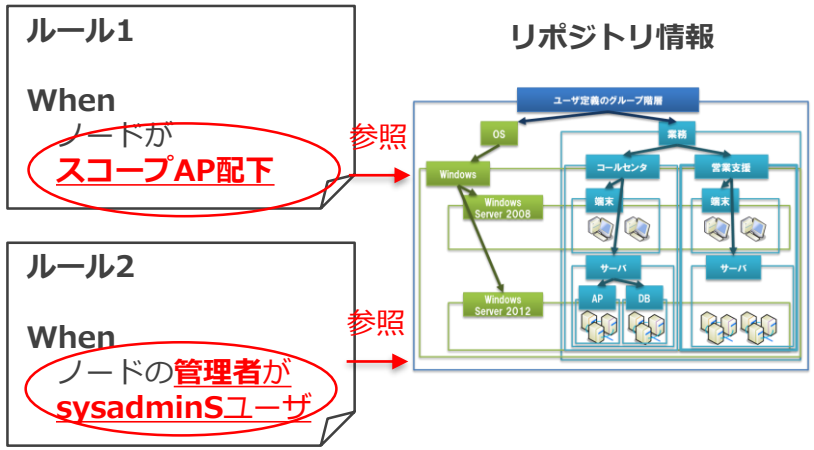
④Hinemosからのシームレスな導入

Hinemosメッセージを受信し、リポジトリ情報をルールの条件内で参照可、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

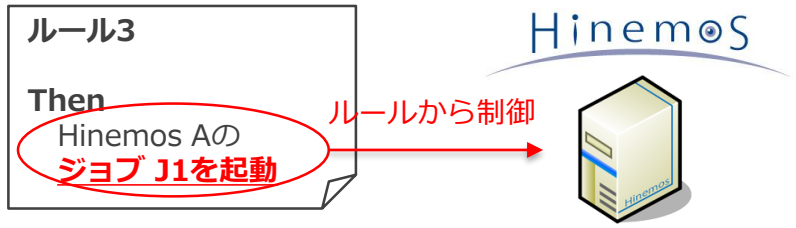
Hinemosの監視やジョブ実行結果をメッセージ通知を使ってメッセージフィルタに簡単に連携できます



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

メッセージ入力

フィルタ処理

ルールアクション

機能詳細

機能一覧

◆フィルタマネージャ

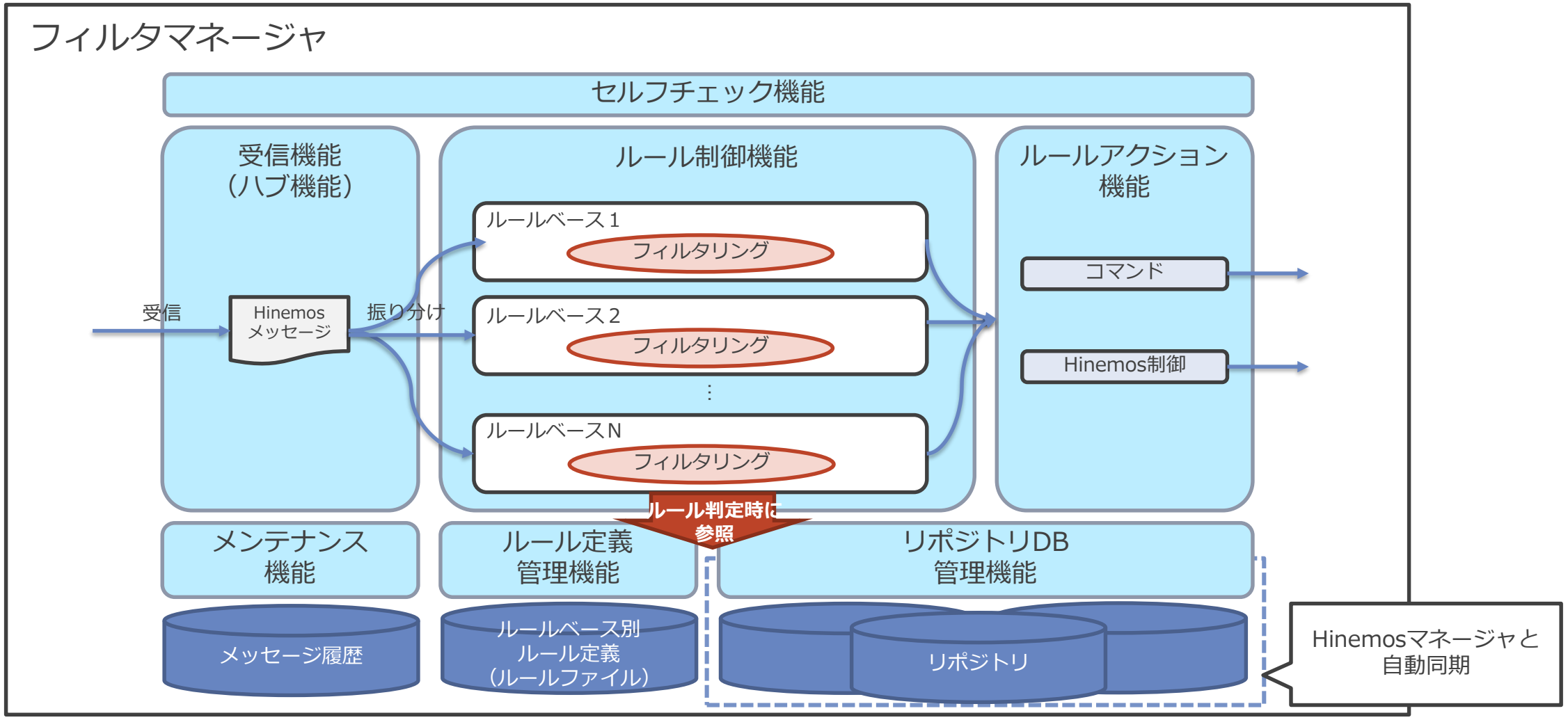
機能分類	機能名	説明
基本機能	フィルタマネージャの入力・ルール処理・出力になる基本的な機能	
	受信機能（ハブ機能）	各種メッセージを受信します
	ルール制御機能	メッセージにルールを適用します
	ルールアクション機能	ルール適用後の各種アクションを管理します
ルール定義機能	ルール定義を簡易化するための定義・情報を参照するための補助的な管理機能	
	ルール定義管理機能	ルールファイルを管理します
	リポジトリDB管理機能	ルール内で参照する情報を管理します
管理機能	フィルタマネージャ自体を安定運用するための管理する機能	
	メンテナンス機能	フィルタマネージャのメンテナンスをします
	セルフチェック機能	フィルタマネージャ自身のセルフチェック機構です

◆Hinemosメッセージフィルタ開発キット

機能分類	機能名	説明
ルール開発機能	フィルタマネージャに適用するルールを定義するためのルール定義開発の支援機能	
	ルール開発支援機能	ルール開発を支援するツールキットです
	ルールシミュレーション機能	ルール実行、メッセージ送信をシミュレーションします

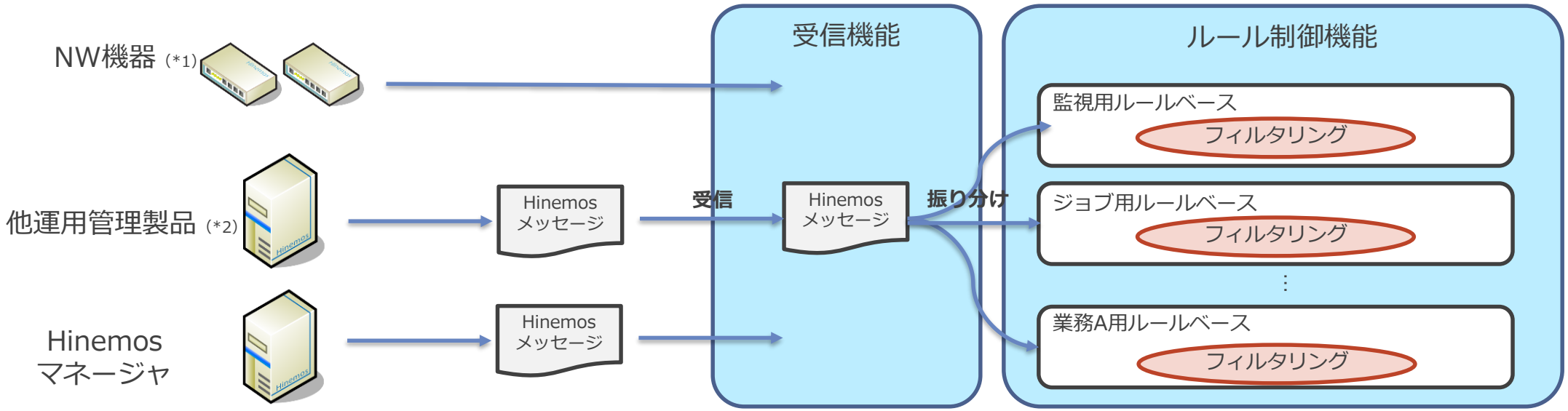
フィルタマネージャ

Hinemosメッセージを受信し、フィルタリングを行い、必要に応じてアクションを実行します



基本機能　　受信機能

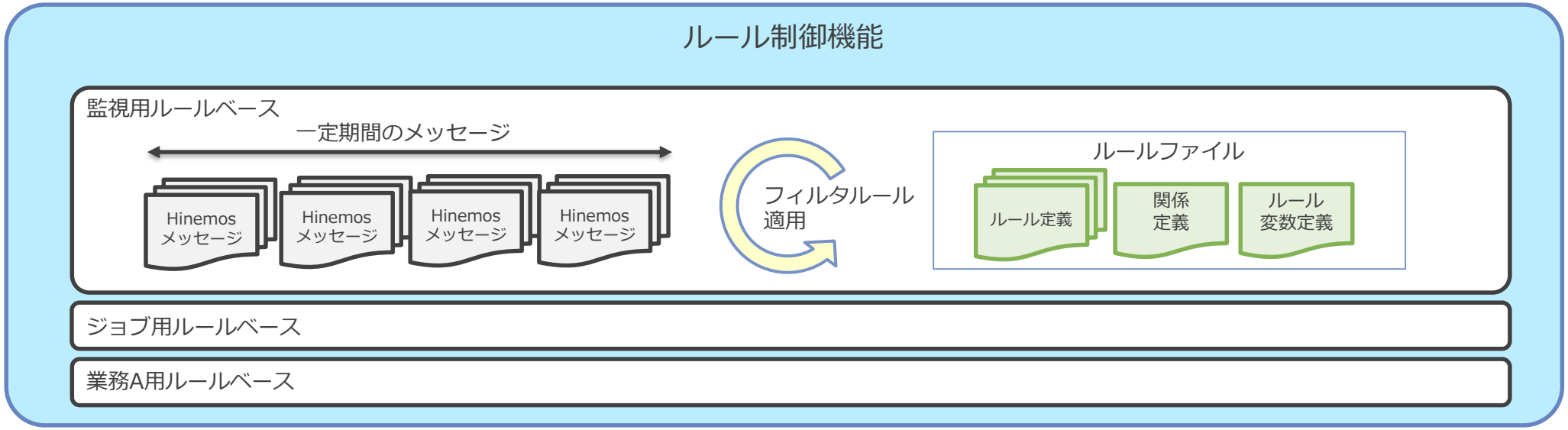
Hinemosメッセージを受信し、フィルタリングの定義の単位（ルールベース）にメッセージを振り分けをします



※1）syslog、SNMPTRAPの様な汎用的なプロトコルへは今後、順次拡張予定です
※2）Hinemosメッセージの形式の電文であれば各種の運用管理製品に対応可能です

基本機能　　ルール制御機能

ルールベース単位で定義したルール（ルールファイル）に従い、メッセージをCEP（complex event processing）にも対応したフィルタリングをします



ルール定義

DRLに従ってWhen/Then形式で表記します。複数のファイル、ルールを適用する事が可能です。

ルールA
When
　NW機器XのPINGがERROR
Then
　sshコマンドを実行

ルールB
When
　JOB 001が失敗
Then
　復旧ジョブを起動

CEP（complex event processing）

複数のイベント（メッセージ）の関係をルールで指定できます。例えば、このイベント発生の5分以内に●●のイベントが発生したら、をルール化できます。

ルールC
When
　サーバ SのPINGがERROR & 5分以内にNW機器の障害
Then
　サーバ SのPINGのERRORは何もしない

基本機能　　ルールアクション機能

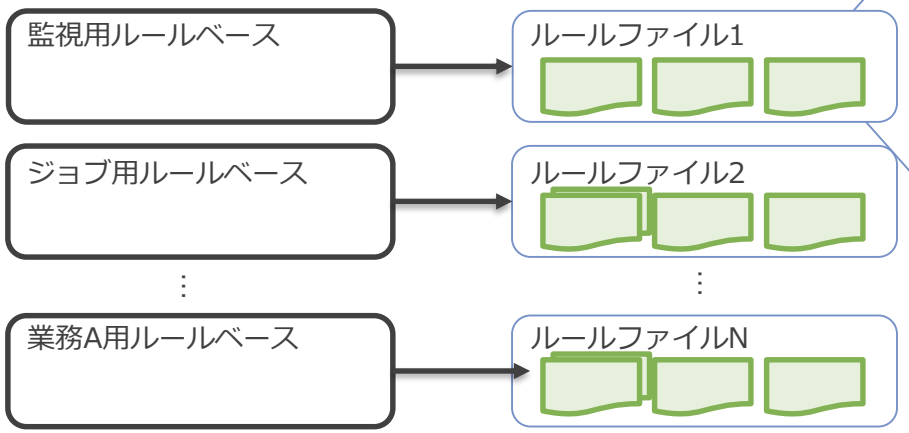
ルールによって指定されたアクションの実行を管理します

アクション分類	概要	アクション詳細
メッセージの制御	ルールの判定の結果、 メッセージを生成、更新、削除等を行う	メッセージの判定終了
		メッセージの生成
		メッセージの更新
インシデントの送信	ルールの判定の結果、 外部システムへインシデントを送信する	コマンド実行
Hinemosとの連動	ルールの判定の結果、 送信元のHinemosに対してアクションを 実行する	Hinemosのジョブの実行
		Hinemosの監視設定の有効/無効化
		Hinemosの収集設定の有効/無効化
		Hinemosの通知設定の有効/無効化
		Hinemosのノードの有効/無効化

ルール定義機能　ルール定義管理機能

フィルタしたいルールの定義一式をルールファイルとして管理します。また、ルール定義から関係定義やHinemosのリポジトリ・カレンダを参照するクエリヘルパー関数を提供します。

用途別で作成したルールベースに、どのルールを適用するかを指定する



ルールファイル

次の3種類のファイルをアーカイブしたものをルールファイルと呼ぶ

ルール定義

- 必須
- DRLに従ってWhen/Thenで表記するテキストファイル
- 複数のファイル可
- ルール定義内でクエリーヘルパー関数を使用可

関係定義

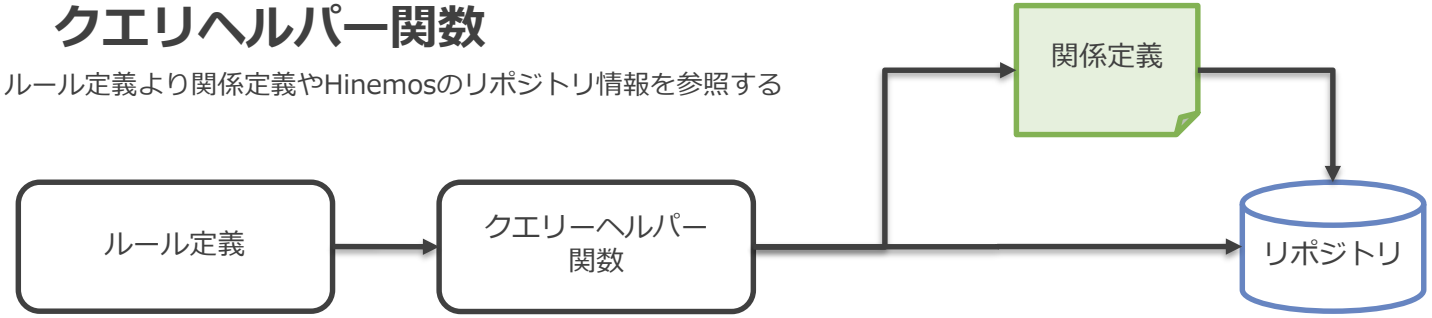
- 任意
- 重点ノードや依存関係を定義するエクセルファイル
- 1ファイルのみ
- クエリーヘルパー関数より使用可能なデータ

ルール変数定義

- 任意
- CSVで表記するテキストファイル
- 1ファイルのみ
- ルール内で使用可能なデータ

クエリヘルパー関数

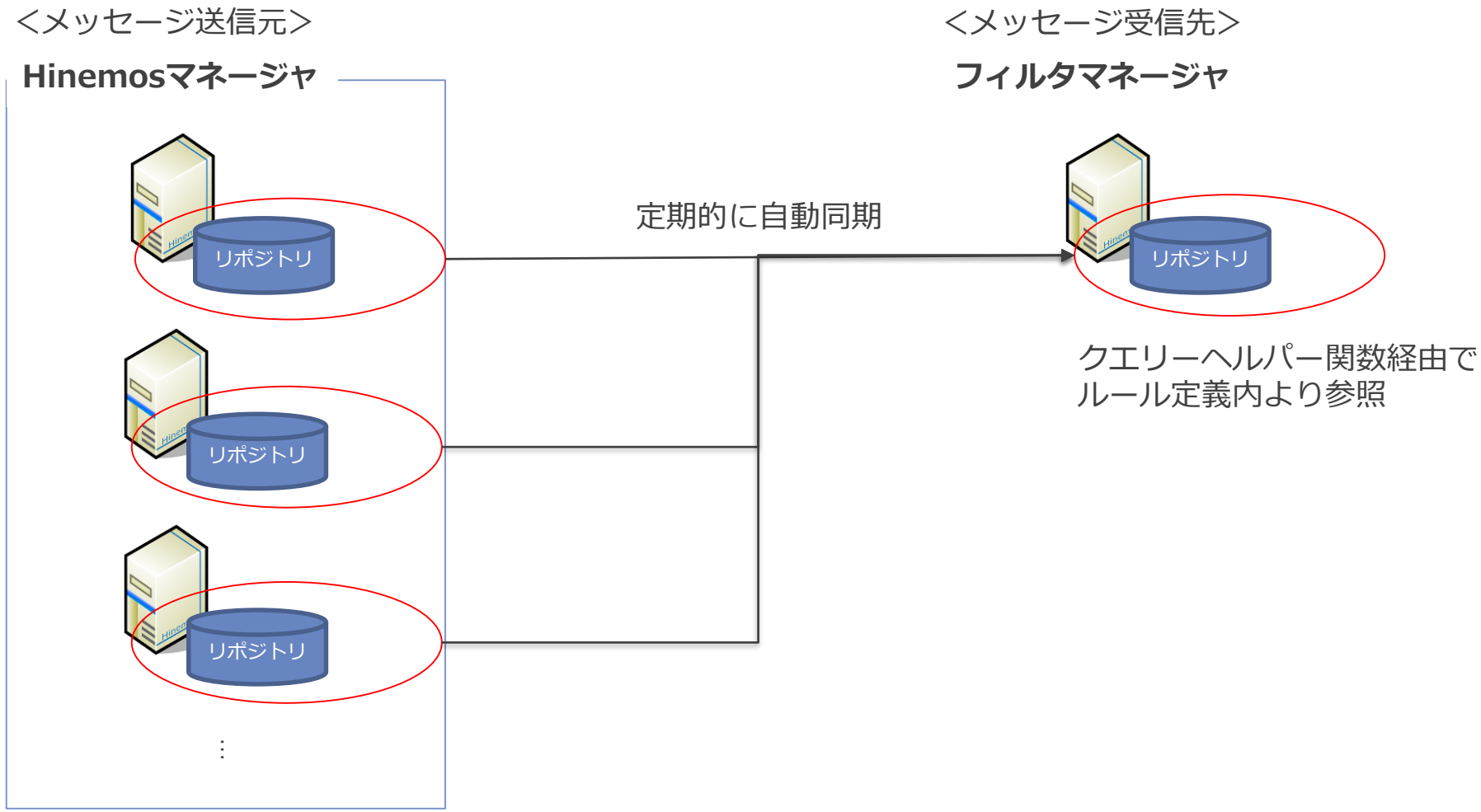
ルール定義より関係定義やHinemosのリポジトリ情報を参照する



例）対象のメッセージがHogeスコープ所属するノードかの判定に使用

ルール定義機能　リポジトリDB管理機能

ルール定義を簡易に記載するためのHinemosのリポジトリ情報の同期管理をします



管理機能

Hinemosマネージャと同様に自身の正常性を監視するセルフチェック機能とメンテナンス機能があります

セルフチェック機能

分類	チェック項目
パフォーマンス低下	遅延タスク監視
	長期実行スレッド監視
	滞留タスク監視
	メッセージバースト監視
	ファクト数監視

メンテナンス機能

運用	スクリプト名
データベースのバックアップ	hinemos_fm_backup.sh
データベースのリストア	hinemos_fm_restore.sh
データベースの再構成	hinemos_fm_cluster_db.sh
ルール変数のエクスポート	hinemos_fm_export.sh
メッセージのエクスポート	hinemos_fm_export.sh
ルールベースの有効・無効化	hinemos_fm_set_rulebase.sh
一時データの削除	hinemos_fm_clear_tmp.sh
履歴データの即時削除	hinemos_fm_delete.sh
メッセージListen有効・無効化	hinemos_fm_set_listen.sh
アクションの有効・無効化	hinemos_fm_set_action.sh
リポジトリDBの即時削除	hinemos_fm_delete_hinemosdb.sh
環境サマリ	hinemos_fm_summary.sh

Hinemosメッセージフィルタ開発キットとサンプルルール集

ルール開発や動作確認を簡単行う開発キットと、サンプルルール集を提供しています。PC上でルール開発やシミュレーションが簡易に行えます。

Hinemosメッセージフィルタ開発キット

提供物	説明
DRL開発用IDE	・**DRL開発用のプロジェクト**（eclipseベース）を提供 Codeready studioの機能を使用したProjectとしても利用可能
ルールファイル雛形	・ルールファイルを構成する次の**雛形**を提供 - ルール定義ファイル - 関係定義ファイル - ルール変数定義ファイル
シミュレーション機能	・次の2つのシミュレーションを行う機能を提供 **- Hinemosメッセージ送信** Hinemosメッセージの一覧（エクセル）を指定の フィルタマネージャに送信 **- ルール適用** ルールファイル、リポジトリ、関係定義に対して Hinemosメッセージ適用をIDE上で実行
ユーザーガイド	・次のマニュアルを提供 **- Hinemosメッセージフィルタ マニュアル** インストール方法、使い方の基本的なユーザガイド **- 開発キットマニュアル** 開発キット上でルールテンプレートをベースに ルール開発する手順 シミュレーション機能の使い方

サンプルルール集

サンプル	説明
001_SAMPLE	ファシリティIDがSwitchS1で警告のメッセージを危険に変更するサンプル
002_SAMPLE	ファシリティIDがServerA[0-9][0-9]で情報のメッセージは一度受信後3分間抑止するサンプル
003_SAMPLE	ファシリティIDがSwitchS1のメッセージはコマンドを実行するサンプル
004_SAMPLE	ファシリティIDがSwitchS31のSwitchS32どちらかから警告でメッセージが来ても前後30秒でもう片方から情報のメッセージが来ていた場合は抑止するサンプル
005_SAMPLE	30秒以内にファシリティIDがSwitchS31のSwitchS32の両方から警告でメッセージが来た場合に両メッセージを抑止し、新たな危険メッセージ（ファシリティID：SwitchS3X）を発行するサンプル
006_SAMPLE	リポジトリデータ上でSwitchS1_Scopeスコープの配下のノードから来たメッセージを出力するサンプル
007_SAMPLE	関係定義上でSwitchS1ノードの子ノードから来たメッセージを出力するサンプル
008_SAMPLE	ファシリティIDがServerBから始まり危険のメッセージの数をカウントするサンプル
009_SAMPLE	有効期限外のルールは実行されないことを確認するサンプル
010_SAMPLE	salienceを用いたルールの優先度を確認するサンプル

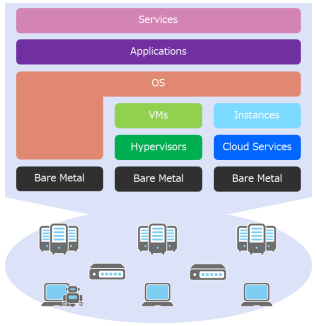
ユースケース

ユースケースの対象システム構成

シンプルなユースケースの説明のため、**インシデント起票するシナリオ**を対象にします

Before

管理対象システム



監視・ジョブ

Hinemosマネージャ

インシデント起票

インシデント管理ツール

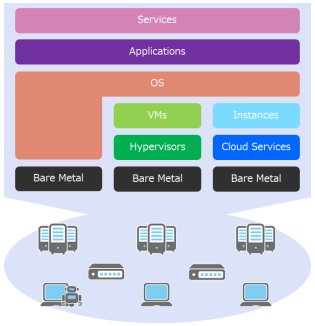
ServiceNow

JIRA **Redmine**

インシデントダッシュボード

After

管理対象システム



監視・ジョブ

Hinemosマネージャ

メッセージ

Hinemosフィルタマネージャ

インシデント起票

インシデント管理ツール

ServiceNow

JIRA **Redmine**

インシデントダッシュボード

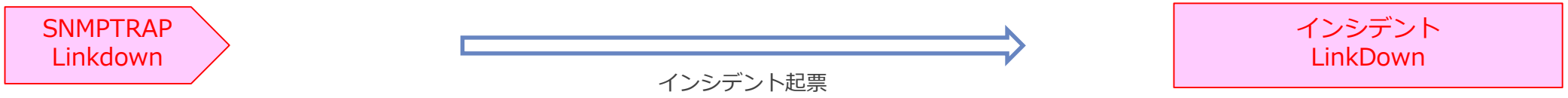
NTT DATA

シナリオ例①　NW機器のLinkDown/Upの集約

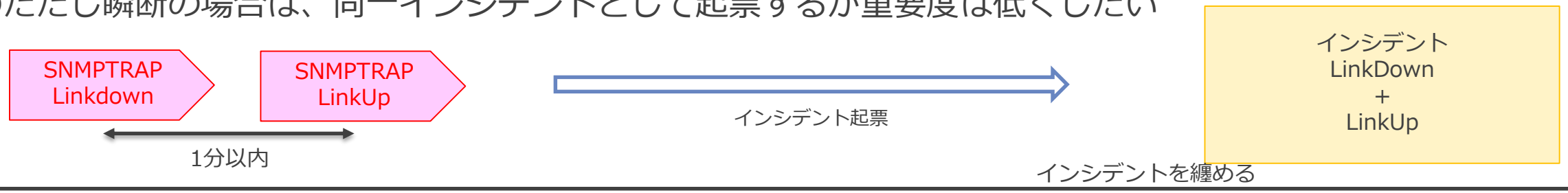
NW機器のLinkDownの検知は重要、だけど…

イベント　　**インシデント**

①LinkDownは直ちに重大なインシデントとして起票したい



②ただし瞬断の場合は、同一インシデントとして起票するが重要度は低くしたい



③どこまでが瞬断かは、経験則的に1分程度とする

シナリオ例②　　毎時ジョブの障害復旧までの通知

毎時ジョブの異常の通知、一度発生すると何回か繰り返す…

イベント　　**インシデント**

①毎時ジョブの異常は、インシデントとして起票したい

毎時ジョブA
12日7時 危険

インシデント起票

インシデント
毎時ジョブA　12日7時 危険

②一度発生すると、2回目以降は警告扱いでインシデント記録

毎時ジョブA
12日7時 危険

毎時ジョブA
12日8時 危険

毎時ジョブA
12日9時 危険

インシデント起票

インシデント
毎時ジョブA　12日7時 危険

インシデント
毎時ジョブA　12日8時 警告

インシデント
毎時ジョブA　12日9時 警告

③ただし、１日経っても改善されない場合は、再び危険扱いでインシデント記録

毎時ジョブA
12日23時 危険

毎時ジョブA
13日0時 危険

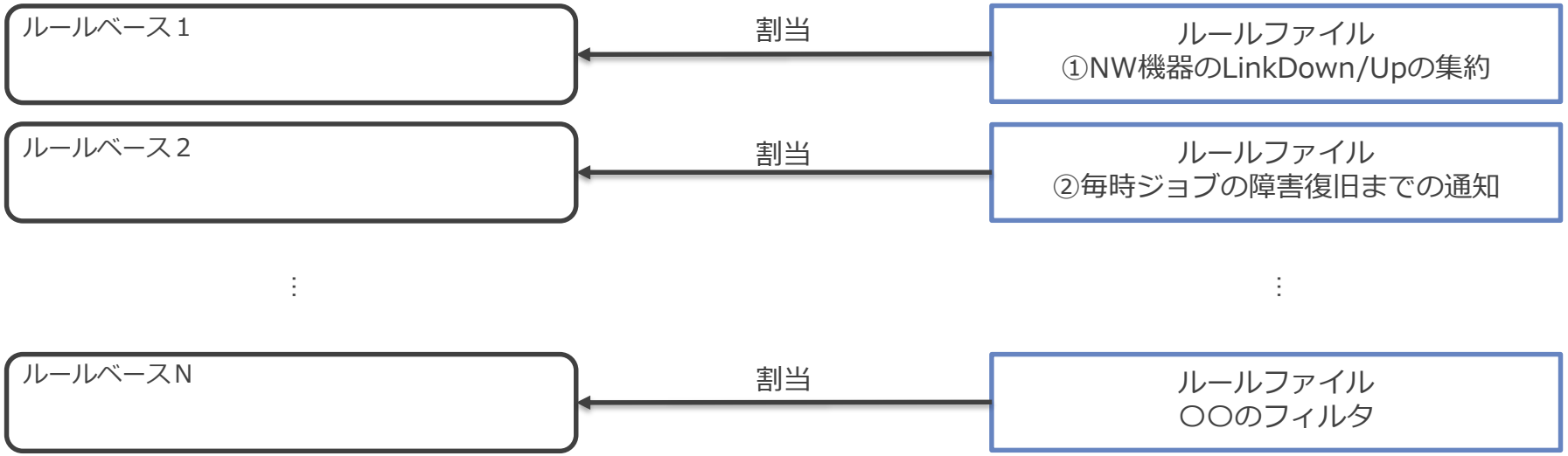
インシデント起票

インシデント
毎時ジョブA　12日7時 危険

インシデント
毎時ジョブA　13日7時 危険

設定　ルールベースとシナリオ

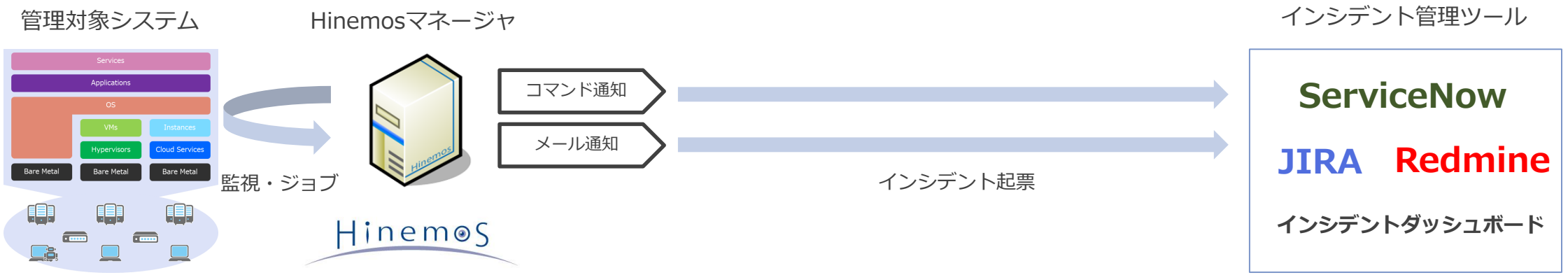
ルールベースという箱に指定のシナリオを処理（フィルタ）するルールファイルを割り当てます



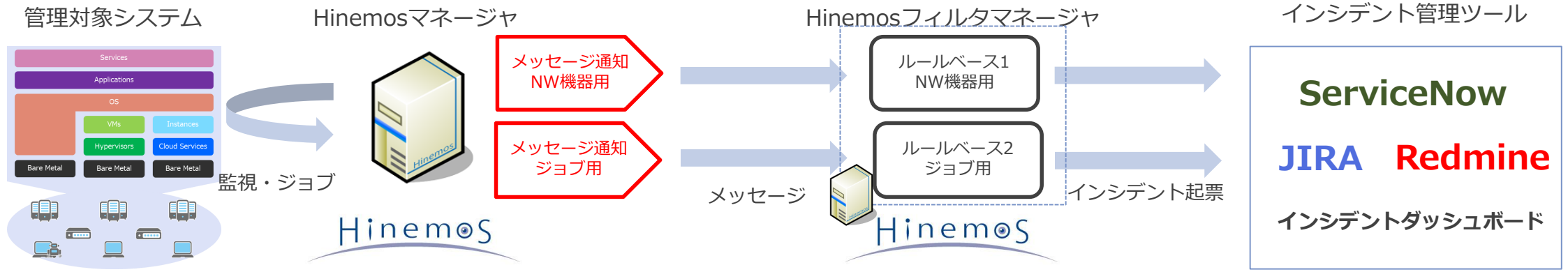
設定　メッセージ通知への置き換え

Hinemosの外部連携用の通知（メール、コマンド通知等）をメッセージ通知に置き換えます

Before



After



設定　ルールとルールアクション

ルールはWhen/Thenの形式で定義し、Thenの中のルールアクションにて外部連携をします

ルールファイル（複数のルール定義ファイルを纏めたもの）

ルールアクション

アクション分類	アクション詳細
インシデントの送信	コマンド実行
Hinemosとの連動	Hinemosのジョブの実行
	Hinemosの監視設定の有効/無効化
	Hinemosの収集設定の有効/無効化
	Hinemosの通知設定の有効/無効化
	Hinemosのノードの有効/無効化

順次拡張予定

設定の全体像のまとめ

次の３ステップで、インテリジェントなアラートを実現します

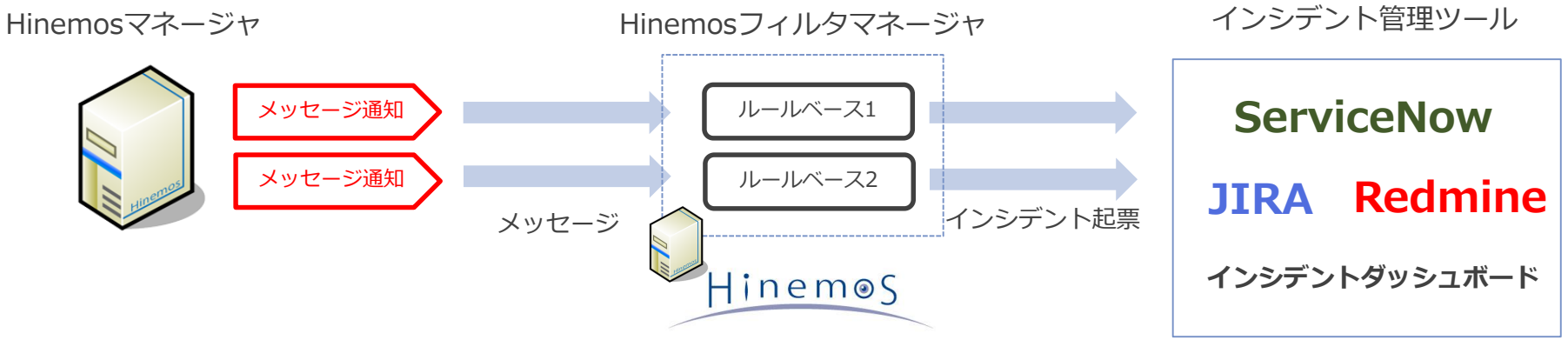
設定・導入作業のフロー

Hinemosマネージャに
メッセージ通知を設定

Hinemosフィルタマネージャの
ルールベースに
ルールファイルを割当

ルールの中の
ルールアクションで
外部連携を実現

運用時の処理フロー



ルール例①　NW機器のLinkDown/Upの集約

NW機器のLinkDown/Upの集約のルールは、次の２つのルールで実現できます

When
メッセージM1がリンクダウン、かつ
メッセージM2がリンクアップ、かつ
M1とM2が1分以内

Then
M1とM2を1つの警告インシデントに起票
M1とM2の削除

1分以内なら
1つのインシデントで起票

When（1分遅延タイマー）
メッセージMがリンクダウン

Then
Mを危険インシデントに起票
Mの削除

1分経過しても残ってるなら
単独のインシデントで起票

ルール例②　毎時ジョブの障害復旧までの通知

毎時ジョブの障害復旧までの通知のルールは、次の２つのルールで実現できます

When
メッセージMがジョブ、かつ、危険
24時間フラグが立っていない

Then
Mを危険インシデントに起票
24時間フラグを立てる
Mの削除

24時間フラグがなければ
危険インシデント

When
メッセージMがジョブ、かつ、危険
24時間フラグが立っている

Then
Mを警告インシデントに起票
Mの削除

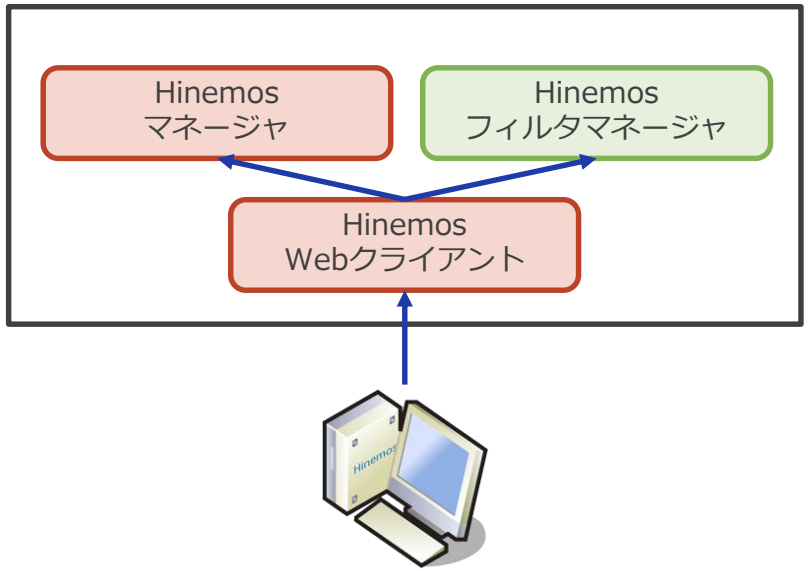
24時間フラグがあれば
警告インシデント

動作要件・サポートサイクル

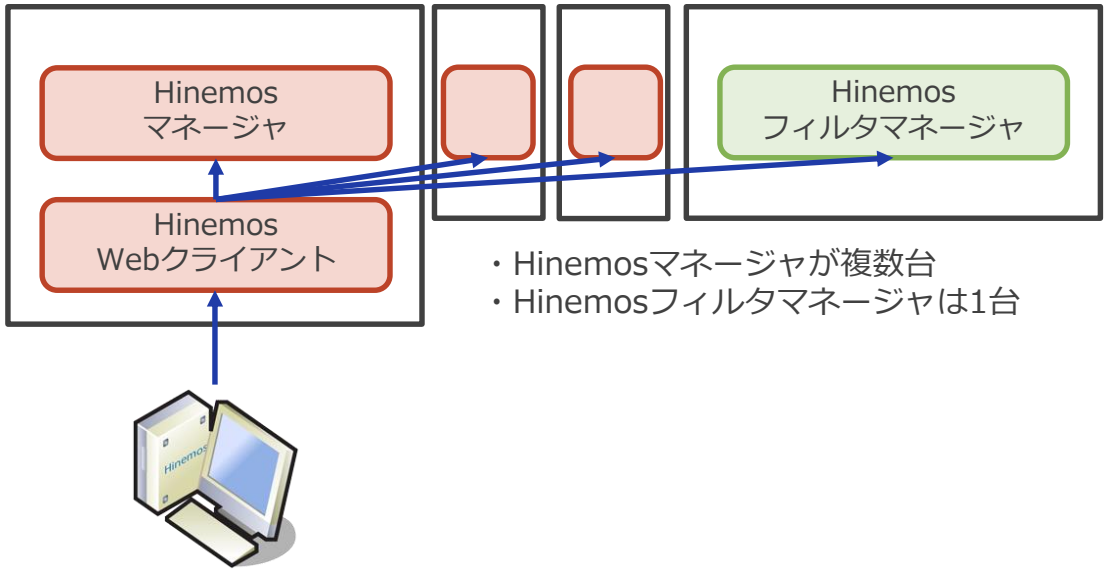
アーキテクチャ

コンポーネント名	説明	備考
Hinemosフィルタマネージャ	・**Hinemosメッセージ**を**フィルタ**するメッセージフィルタのコアコンポーネント ・最小構成はHinemosマネージャとサーバ同居が可能、拡張性としてHinemosフィルタマネージャのみ別サーバに配置可能 ・HinemosマネージャとHinemosフィルタマネージャの関係は、N：1（指標としては最大10台（シングル・MC共通）） ・画面機能はHinemos Webクライアントにより対応	

最小構成（同一サーバ配置）



拡張構成（別サーバ配置）



・Hinemosマネージャが複数台
・Hinemosフィルタマネージャは1台

動作要件

コンポーネント名	項目	詳細	備考
Hinemosフィルタマネージャ	推奨スペック	・スペック（本製品単独で導入する場合） CPU：2GHz, 4コア以上 メモリ：8GB以上 HDD：50GB以上 ネットワークインタフェース：1個以上	
	対応OS	・RHEL7, RHEL8, Amazon Linux 2	
	その他要件	・HinemosJRE対応 ・Hinemosマネージャシングル（Linux版）と同居が可能 ・Hinemosマネージャシングル（Windows版）と同居が不可 ・HinemosマネージャMC機能と同居が不可	
Hinemosフィルタ 開発キット	推奨スペック	・スペック CPU：2GHz, 4コア以上 メモリ：8GB以上 HDD：50GB以上 ネットワークインタフェース：1個以上	
	対応OS	・RHEL7, CentOS7, RHEL8, Windows10	
	その他要件	・Eclipse/Javaの導入が必須 ・HinemosJRE対応	

注意事項）

・Hinemosフィルターマネージャはシングル構成のみ対応（ver.1.0）

・Hinemosのインシデント管理連携ツール（インシデント連携）、コマンドラインツールに対応

サポートサイクル

基本的なライフサイクルの考え方

バージョン	通常保守サービス	延長保守サービス	特別延長保守サービス*2
期間	最大5年*1	最大3年	最大2年
マイナーバージョンリリース	○	×	×
新規パッチ作成	○	×	×
技術問い合わせ	○	○	○
異常対処対応	○	○ *3	○*3

*1：期間は各ソフトウェアのメジャーバージョンリリース日付を起点とします。
*2：期間・内容・費用はご利用案件毎の個別調整となります。
*3：ご利用バージョンでの発生した事象の解決を保証するものではございません。

各サービスの終了日

製品名	バージョン	通常保守サービス	延長保守サービス	特別延長保守サービス
Hinemos メッセージフィルタ	1.0	2027年3月31日	2030年3月31日	2032年3月31日
Hinemos メッセージフィルタ 開発キット	1.0	2027年3月31日	2030年3月31日	2032年3月31日

ご相談・お問い合わせ

お問い合わせはこちら

製品・サービスに関するお問い合わせはこちら

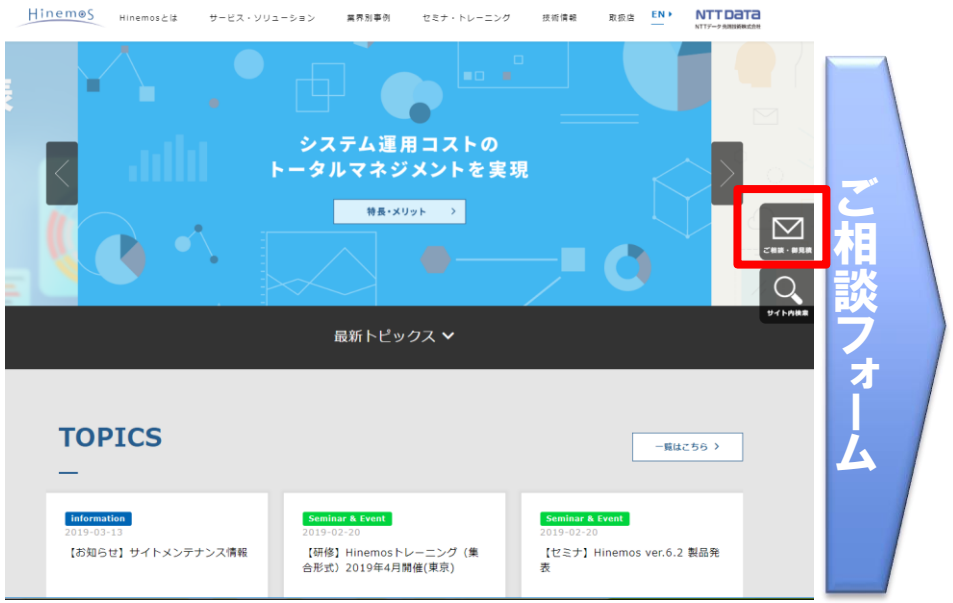
お待ちしているもに！

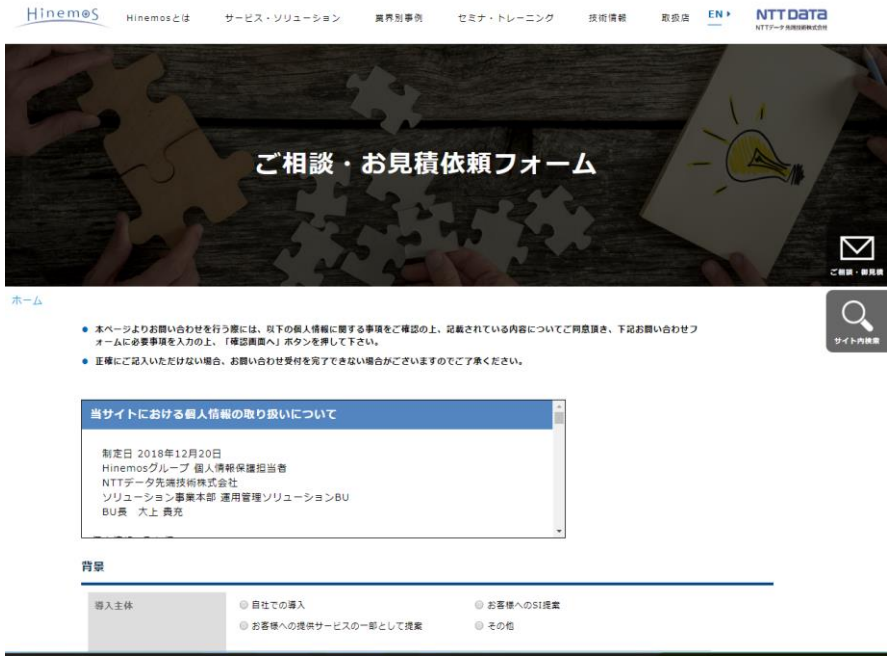
Hinemosに関するお問合せ

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





NTT DATA

Trusted Global Innovator