



## **Zabbixのアラートを効率良く インシデント管理ツールと連携する方法**

---

## ITIL Ver3

Service Strategy

Service Design

Service Operation

Continual Service Improvement

ITILは“**運用管理の標準**”として、  
一時爆発的に取り上げられて  
今では“**あたり前の事**”となっています

ITILに準拠し機能を有した商用製品を利用



**運用改善に成功する企業**

# 運用課題の整理に関して

一方で・・・  
運用改善を勧めていく上での課題

1. “プロセス整理の壁”
2. 商用ツールの“コストの壁”



## ゴールを前に断念してしまう企業。

断念してしまう理由

“ITIL” 呪文に掛かって運用改善に取り生み出すと・・・

1. 全てを一度に取り組まないとなかなか成果が出ないのではないか？
2. “ITIL”で書かれていることを、具体的に何をすればよいのか？



## small steps/quick wins

「できるところから小さく始めて早く結果を出す」

運用現場の改善には大変重要な事

1. 課題解決に順位をつけて
2. 中間ゴールを設けて

運用課題順位付け  
1. 監視システムの課題  
2. インシデント管理の課題  
3. 障害対応自動化の課題  
4. 変更管理の課題  
5. 構成管理の課題  
6. . . . .



## 運用課題克服へのベストウェイ



## 1. 業務の効率化



## 2. 人的ミス削減

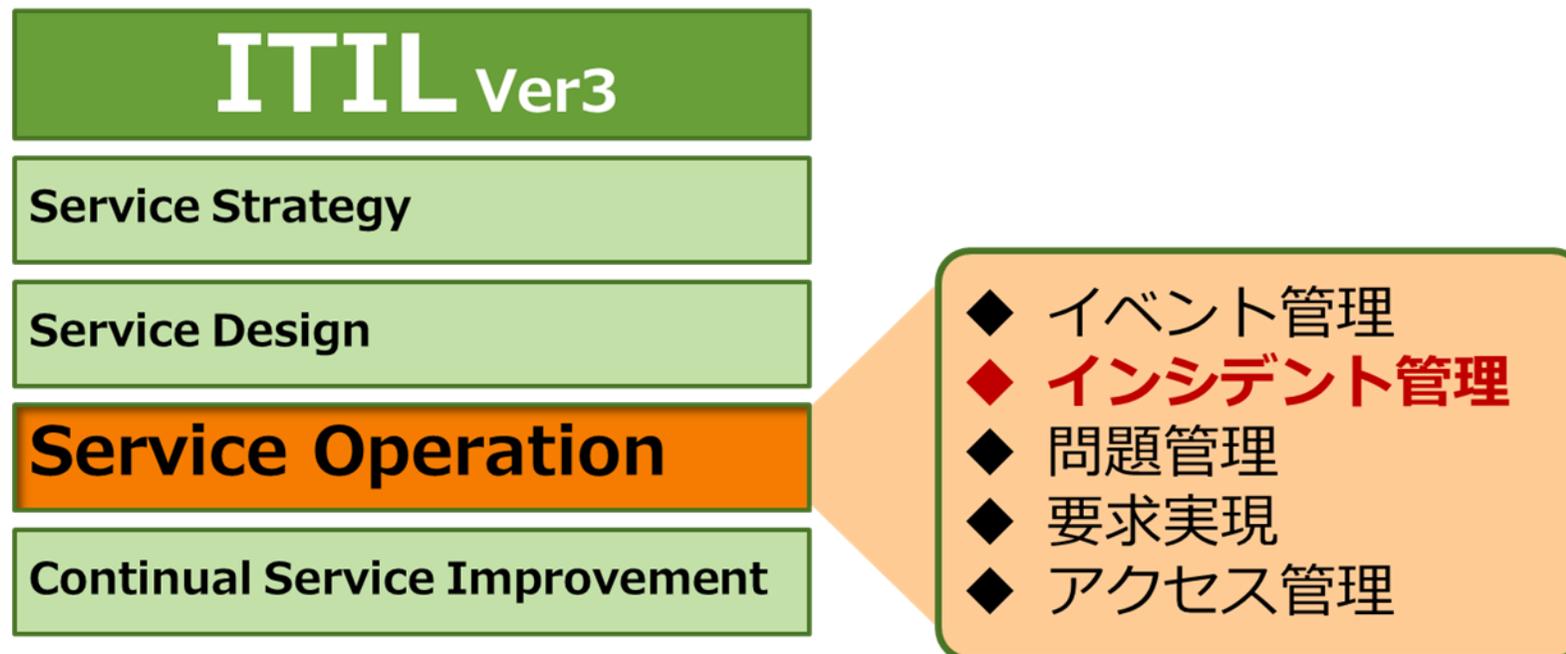


## 3. 運用プロセスの定着



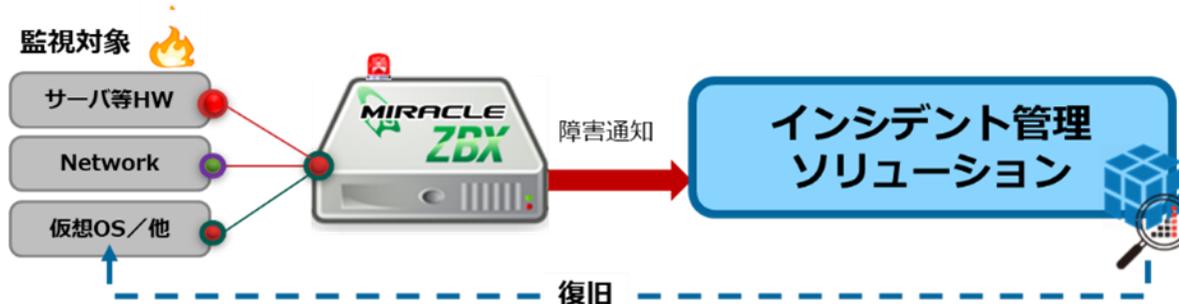
## 4. サービス拡大に対応

サービスオペレーションのプロセスの一つ“インシデント管理”





インシデントにより中断されたITサービスを早急に復旧させ、  
ビジネスの負のインパクトを最小限にすること



## インシデント管理のプロセス

- 1 検知と記録
- 2 分類と初期サポート
- 3 調査と診断
- 4 解決と復旧
- 5 インシデントのクローズ

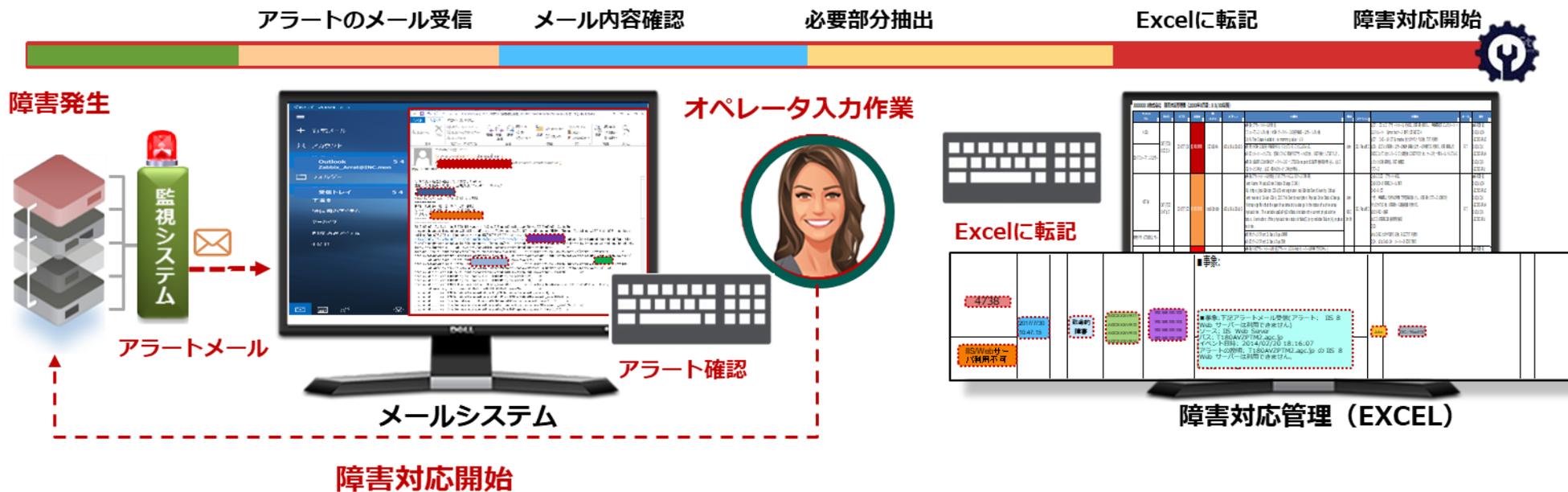
**シンプルではありますが、確実に行うことは大変です。**

しかし、ITサービスの運用を円滑に回す為の重要なキーとなるのでしっかりと行う必要があります。

その為には・・・

**専用のツールを導入することも決解方法の一つ**

## 監視ツールからのイベントをメーラーで受け取り、Excelで管理する場合



### 【課題】

- ✓メール電文を見て障害対応の必要性を判断 → 遅延
- ✓メール電文からチケットに必要な項目を転記 → 記述ミス
- ✓チケット起票を優先すると対応着手が遅れる → SLA違反へ
- ✓障害対応の優先で対応状況がわからない → 管理に支障

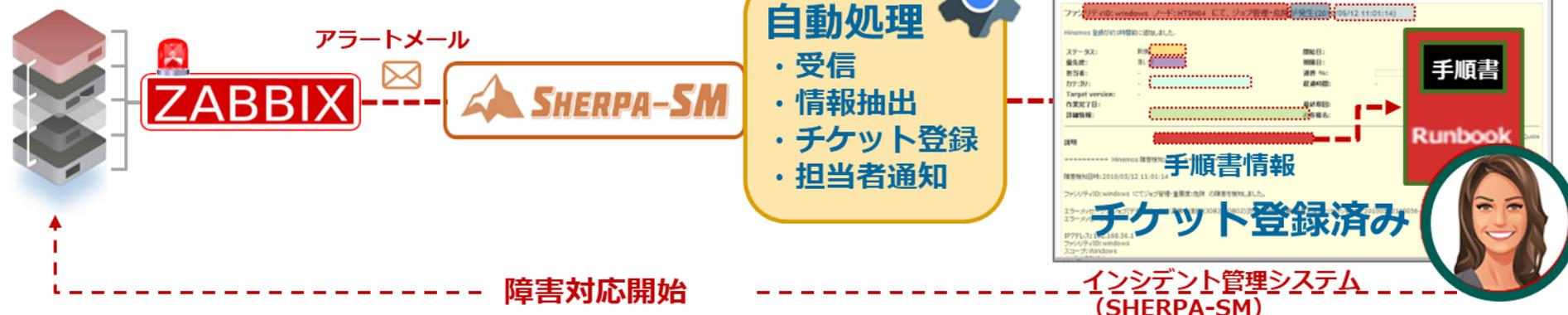
# インシデント管理ツールを導入することで

## 監視ツールからのイベントをインシデント管理ツールで管理する場合

アラートメール受信／情報抽出／チケット登録／担当者通知

障害対応開始

障害発生



### 【効果】

- ✓ 全ての通知を自動取込み（管理漏れが無くなる）
- ✓ 必要な項目を自動転記（起票ミス／記載漏れ無し）
- ✓ 該当担当者通知（譲り合っでの対応遅延防止）
- ✓ 全ての対応状況把握と進捗状況の把握

## ZabbixからのアラートをSHERPA – SMのチケットテンプレートの機能で連携



**障害発生** → **ZABBIX** → **SHERPA-SM**

**インシデント自動登録**

**【Zabbixからの通知メールサンプル】**

名前：アラートメール送信  
デフォルトの件名：【障害】 {TRIGGER.NAME}:  
{ITEM.LASTVALUE}: zabbix  
デフォルトのメッセージ：  
Original event ID: {EVENT.ID}  
障害発生時刻: {DATE} {TIME}  
ホスト名: {HOST.HOST}  
IPアドレス: {HOST.IP}  
設置場所: {INVENTORY}  
深刻度: {TRIGGER.SEVERITY}  
障害内容: {TRIGGER.NAME}  
最新値: {ITEM.LASTVALUE}

**必要な情報を自動取り込み**

**フィールドも増やせます**

**メール原文も表示できます**

**システム監視 #69**

[sherpa-ir] [\*\* PROBLEM Service Alert: man-mx02/System Load is CRITICAL \*\*][2017-09-11\_16-45-231]

Admin Redmine が [2017/09/11 16:46] に追加. [2017/09/11 16:49] に更新.

ステータス:	電話通知連携済み	開始日:	2017/09/11
優先度:	低	期日:	
担当者:	オペレーター1	進捗率:	0%
お客様名:	mx02.man.aws.inb	監視サーバ:	
ホスト:	man-mx02	エラー内容:	System Load
トリガー名:	PROBLEM	障害レベル:	
通知区分:	CRITICAL	対象URL:	https://cheeer.net/eredmine/projects/cloud_common/issues/new
連携種別:	都度	復旧方法:	
影響判定:		原因:	

From: redmine.z@snerpairsm310.local  
To: redmine@sherpairsm310.local  
Cc:  
Date: 2017-09-11T16:45:36+09:00  
Subject: \* PROBLEM Service Alert: man-mx02/System Load is CRITICAL \*  
Zabbix\*\*\*  
Notification Type: PROBLEM  
Host: man-mx02  
Alias: mx02.man.aws.inb  
Address: 10.16.1.10  
Service: System Load  
State: CRITICAL  
Date/Time: Tue May 9 10:06:27 JST 2017  
Additional Info:  
CHECK\_NRPE: Socket timeout after 30 seconds.

# 記入漏れや情報不足などのミスを防止

## マイページ



Home マイページ プロジェクト ヘルプ ログイン: 検索:

**担当分 インシデント**

### マイページ

このページをパーソナライズ

#### 担当しているチケット (13)

#	プロジェクト	トラッカー	題名
17	開発管理	ミドルウェア障害	DBコネクション数の設定が更新されない (新規)
16	開発管理	不具合報告	ログインできません (対応中)
1	インシデント管理	問合せ	ログインできません (対応中)
11	インシデント管理	バージョンアップ	処理高速化対応 (進行中)
10	インシデント管理	バージョンアップ	新機能開発 (進行中)
9	インシデント管理	バージョンアップ	v2.0.0バージョンアップ対応 (進行中)
13	インシデント管理	不具合報告	ユーザガイドの誤字 (対応中)
12	インシデント管理	定例作業	2015年上期欄卸作業 (進行中)
8	インシデント管理	問合せ	HTTPSで接続出来ない (対応中)
7	インシデント管理	障害通知	[Warn]Unjust connection (対応中)

#### 報告したチケット (7)

#	プロジェクト	トラッカー	題名
17	開発管理	ミドルウェア障害	DBコネクション数の設定が更新されない (新規)
16	開発管理	不具合報告	ログインできません (対応中)
11	インシデント管理	バージョンアップ	処理高速化対応 (進行中)
10	インシデント管理	バージョンアップ	新機能開発 (進行中)
9	インシデント管理	バージョンアップ	v2.0.0バージョンアップ対応 (進行中)
13	インシデント管理	不具合報告	ユーザガイドの誤字 (対応中)
12	インシデント管理	定例作業	2015年上期欄卸作業 (進行中)

# 自分の担当案件がひと目で把握でき 対応漏れも防止

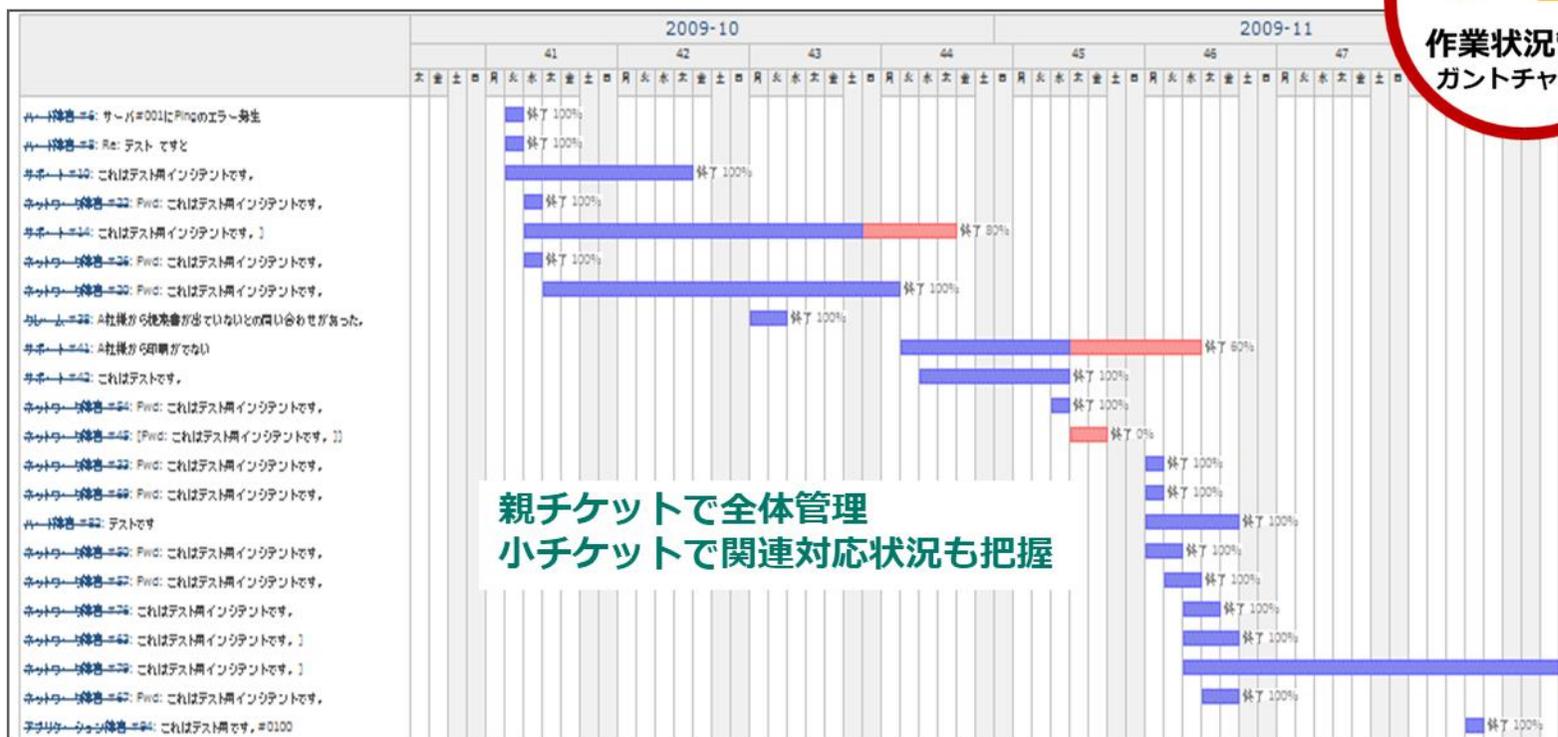
## 優先度表示



✓ # ▼	トラッカー	ステータス	優先度	題名	起票者	担当者	更新日	
<input type="checkbox"/> 383	アプリケーション障害	新規	急いで	ファシリティID: windows ノード: HTSN04 にて、ジョブ管理・危険が発生(2010/05/12 11:01:14)	Hinemos 登録		2010年5月12日 11:01 AM	
<input type="checkbox"/> 382	アプリケーション障害	新規	急いで	ファシリティID: windows ノード: HTSN04 にて、ジョブ管理・危険が発生(2010/05/12 10:49:45)	Hinemos 登録		2010年5月12日 10:49 AM	
<input type="checkbox"/> 381	アプリケーション障害	新規	急いで	ファシリティID: windows ノード: HTSN04 にて、ジョブ管理・危険が発生(2010/05/12 10:38:43)	Hinemos 登録		2010年5月12日 10:38 AM	
<input type="checkbox"/> 379	ハード障害	担当	高め	ファシリティID: redmine ノード: redmine にて、DISK監視「/」警告が発生(2010/05/11 16:04:00)	Hinemos 登録	管理 運用	2010年5月11日 18:19 PM	
<input type="checkbox"/> 379	ハード障害	担当	高め	ファシリティID: redmine ノード: redmine にて、DISK監視「/」警告が発生(2010/05/11 16:04:00)	Hinemos 登録	管理 運用	2010年5月11日 18:19 PM	
<input type="checkbox"/>	監視ツールから障害の“優先度”をもとに背景色帯の変えて表示						管理 運用	2010年5月11日 18:19 PM
<input type="checkbox"/> 379	ハード障害	担当	高め	ファシリティID: redmine ノード: redmine にて、DISK監視「/」警告が発生(2010/05/11 16:04:00)	Hinemos 登録	管理 運用	2010年5月11日 18:19 PM	
<input type="checkbox"/> 379	ハード障害	担当	高め	ファシリティID: redmine ノード: redmine にて、DISK監視「/」警告が発生(2010/05/11 16:04:00)	Hinemos 登録	管理 運用	2010年5月11日 18:19 PM	

各障害対応が、どの程度急を要するものか見極めも必要

# ガントチャート



# リアルタイムに対応状況を把握

## 検索 & 詳細絞り込み

検索

全プロジェクト  すべての単語  タイトルのみ

チケット  文書

▼ チケット詳細検索

プロジェクト

- ソリューション部
  - 0.サポート業務引き継ぎ
  - 0.メンバー教育プロジェクト
  - 0.各種管理
    - サーバ作業申請システム
    - 資産管理
    - ウィルス対策ソフト管理
    - 稼働管理

[すべてにチェックをつける](#) | [すべてのチェックを外す](#)

ステータス 等しい ▼ 新規 ▼

トラッカー 等しい ▼ 資料作成 ▼

Account ID 等しい ▼

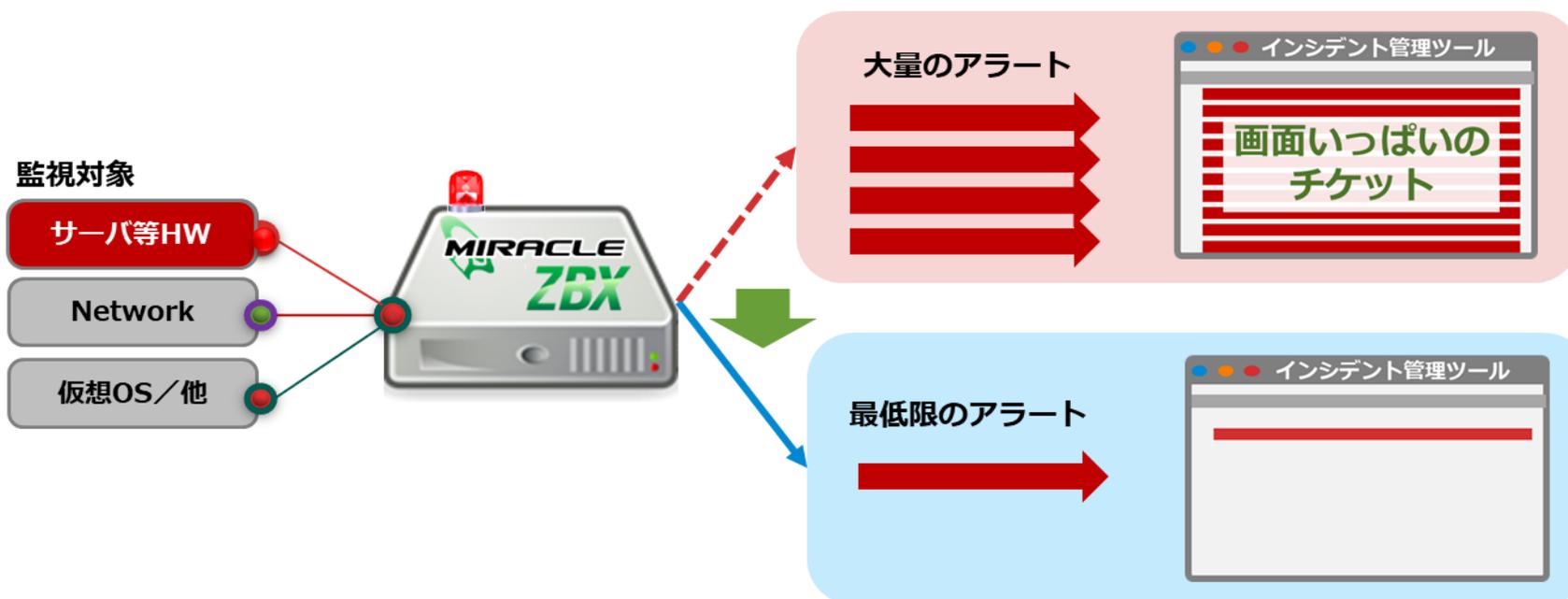


# 対応の履歴を共有することで 障害対応の復旧時間 (MTTR)を短縮

# インシデント管理をうまく回すには・・・

## うまく回すにあたっての理想

- ✓ インシデント登録は必要最低限にしたい。（それ以上はノイズにしかならない）
- ✓ インシデントは、人間によるアクションが必要なもののみに絞りたい。

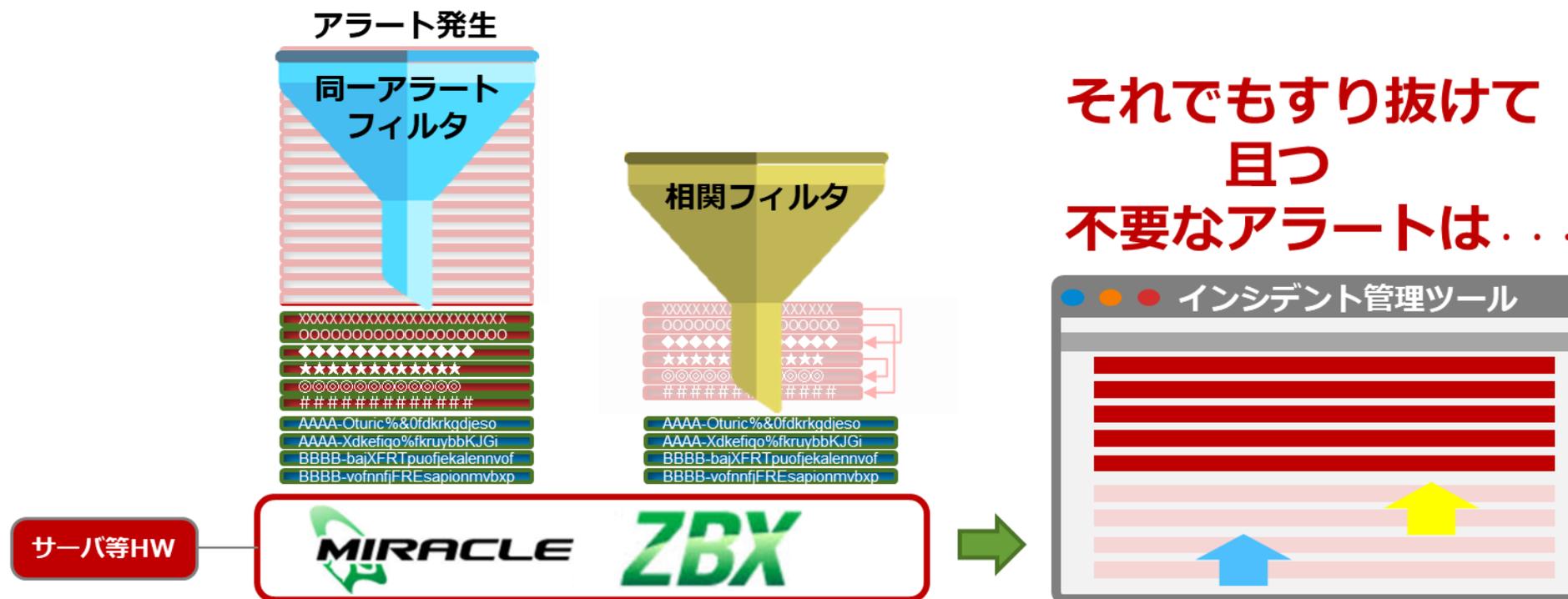


## 具体的には何をすべきなのか・・・・・・・・？

- インシデント管理に登録する前に“アラートの制御を！”

## Zabbix設定によるアラートの抑制

- ✓ 指定時間帯の同一アラートの抑制
- ✓ トリガーの「依存関係」設定による相関関係アラートの制御

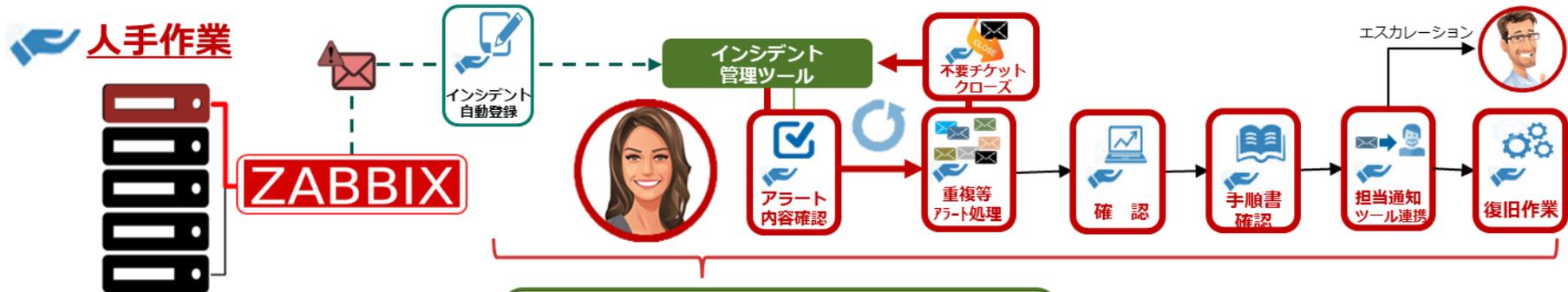


## SHERPA-IRの設定によるアラートの抑制

- ✓ アラート内容をルールベースで抑制
  - ✓ プラスアルファ (担当者判断、手順書URL、コマンド発行等)
- 【 都度重複旧延  
◆ 重複  
◆ 復線  
◆ 繰 】



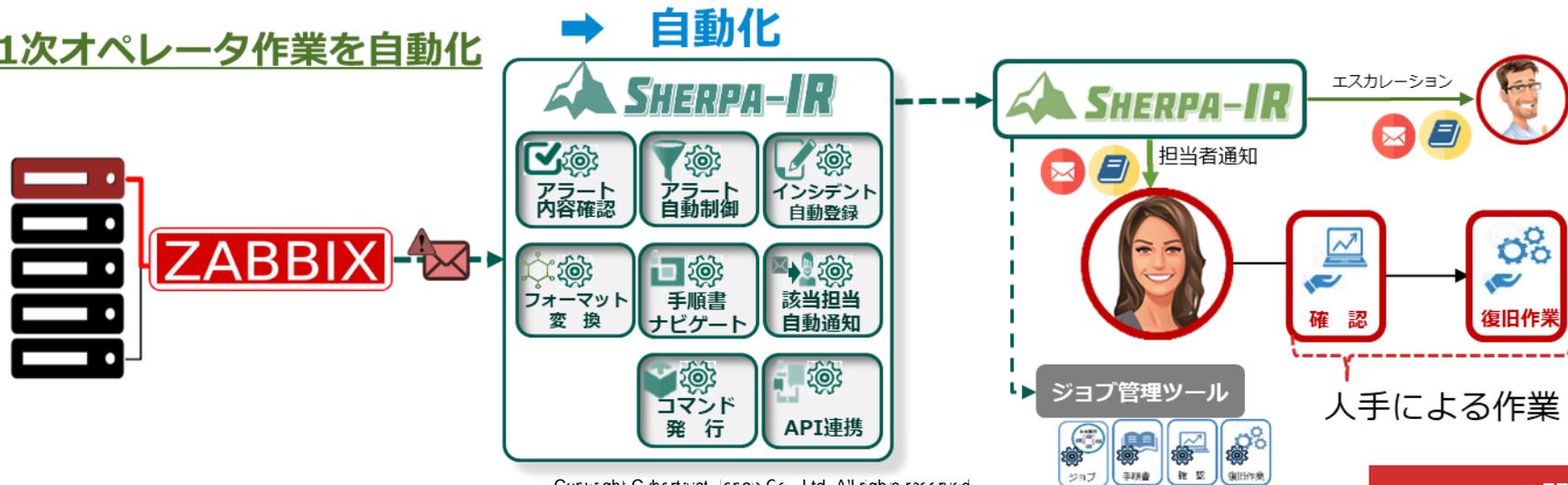
# SHERPA-IRは1次オペレータ作業の自動化を支援

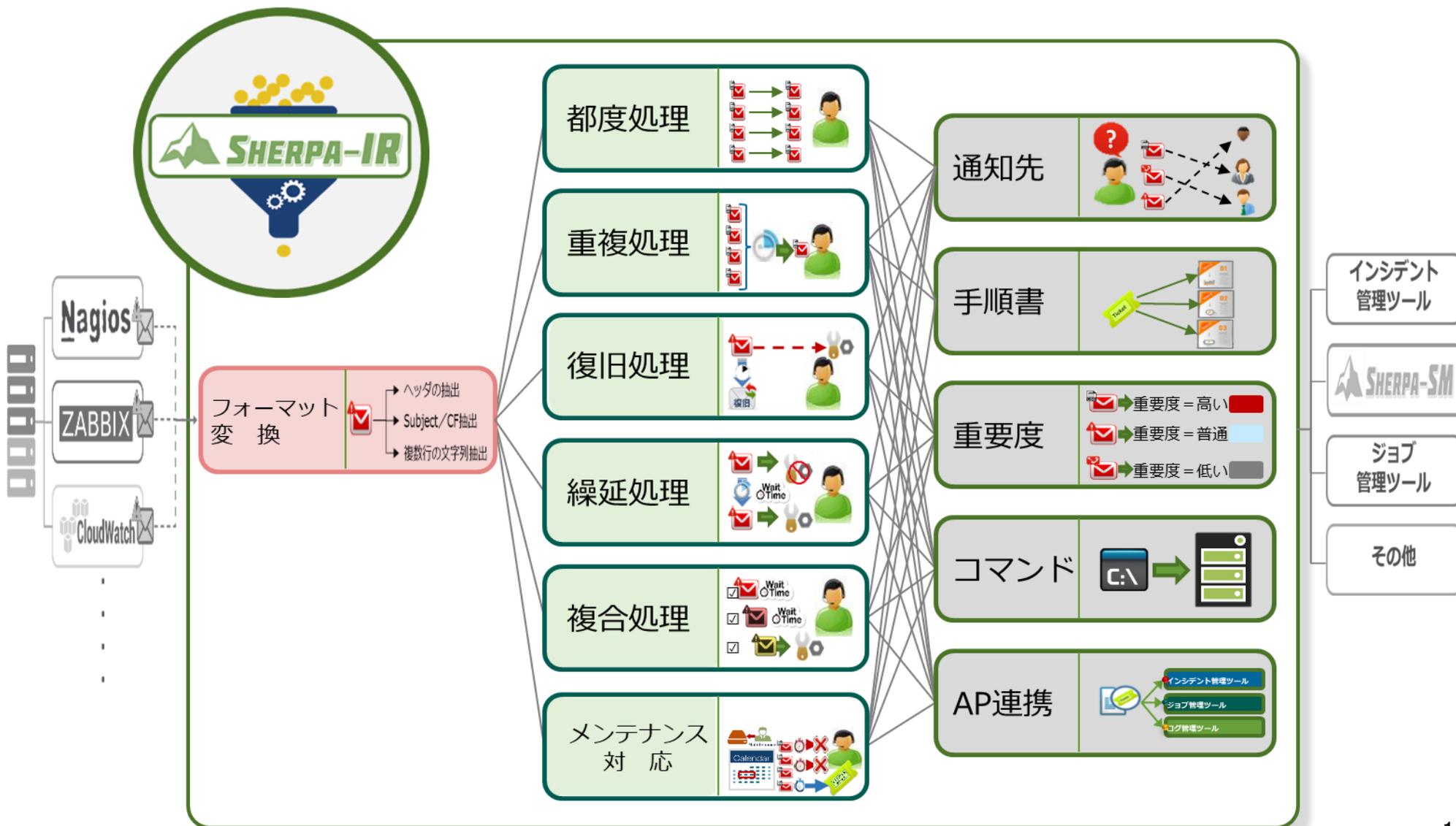


- 1次オペレーターの人手による作業**
- ✓ チケット内容確認と処理判断
  - ✓ 重複等不要チケットのクローズ処理
  - ✓ 該当手順書の検索
  - ✓ 後続ツールへの連携処理
  - ✓ 該当担当者への通知
  - ✓ 障害対応作業

- ➔ **障害対応の遅延**
- ➔ **ミスの発生**
- ➔ **サービス品質低下**

## 1次オペレータ作業を自動化





# SHERPA-IR 設定の流れ

## 抽出情報

概要 活動 チケット 抽出情報 更新情報 フィルタ情報

### 新しい抽出情報

抽出セット名 \*

説明

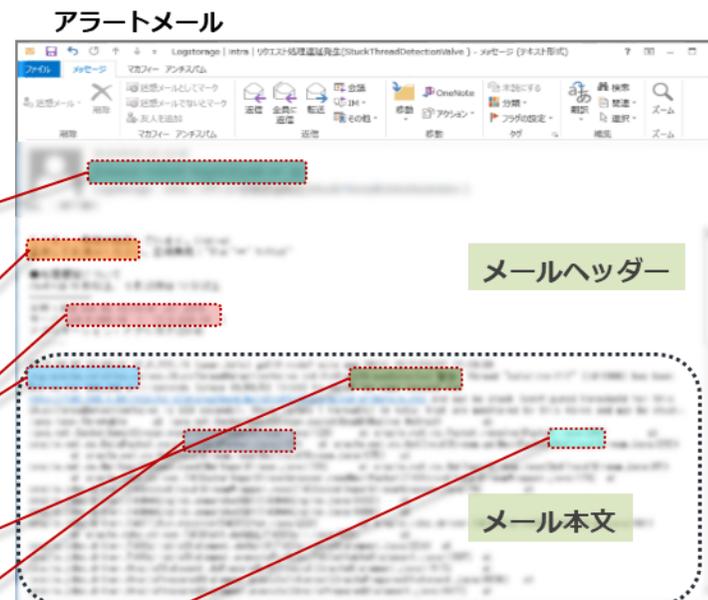
件名に対する正規表現 \*

大小無視

複数行可

#### カスタムフィールド

抽出CF名	抽出方法	正規表現	大小無視	複数行可
お客様名	標準	<input type="text" value="Alias: (.*)"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
ホスト	標準	<input type="text" value="Host: (.*)"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
トリガー名	標準	<input type="text" value="Notification Type: (.*)"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
通知区分	標準	<input type="text" value="State: (.*)"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
手順書URL	標準	<input type="text"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
障害判定	標準	<input type="text"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
対象Webサーバ	標準	<input type="text" value="Address: (.*)"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
エラー内容	標準	<input type="text" value="Service: (.*)"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
障害レベル	標準	<input type="text"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
対象URL	標準	<input type="text"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
復旧方法	標準	<input type="text"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
原因	標準	<input type="text"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>



フィールドに設定したい値を入力します。  
今回の例では、メールテンプレートの内容をフィールドに設定し、メールの件名からパターンを洗い出し、正規表現で設定。

※連携イメージ

## STEP1 どのようなアラートが来たら？

新しい更新情報

お客様名 \*

ホスト

トリガー名 \*

通知区分 \*

キーフィルタ名

有効

処理情報

処理時ステータス

処理時実行コマンド \*

手順書URL

非処理時ステータス

非処理時実行コマンド

処理条件

処理フィルタ名

処理タイプ  監視時間(分)

処理契機(発生回数)  対象チケット

イベントタイプ  追い越し

障害の更新情報

新しい抽出情報

抽出セット名 \*

説明

件名に対する正規表現 \*

大小無視

複数行可

カスタムフィールド

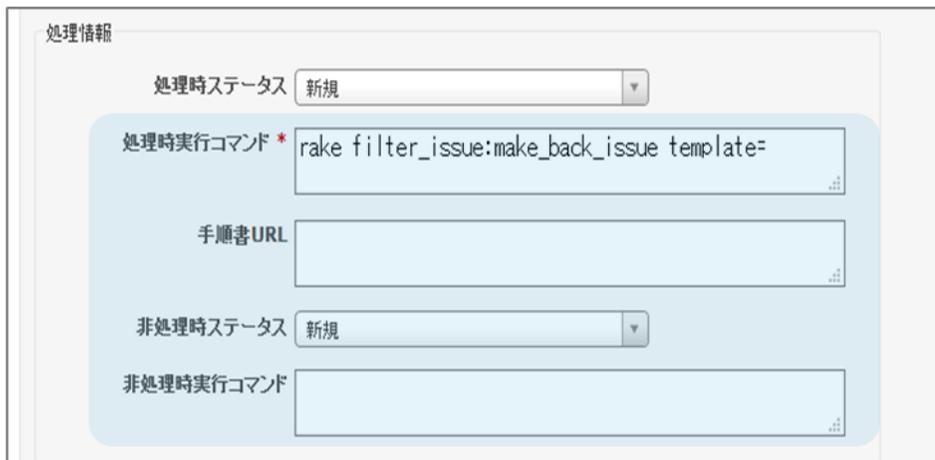
抽出CF名	抽出方法	正規表現	大小無視	複数行可
お客様名	標準	<input type="text" value="Address:(*)"/>	<input type="text" value=""/>	<input type="text" value=""/>
ホスト	標準	<input type="text" value="Host:(*)"/>	<input type="text" value=""/>	<input type="text" value=""/>
トリガー名	標準	<input type="text" value="Trigger:(*)"/>	<input type="text" value=""/>	<input type="text" value=""/>
通知区分	標準	<input type="text" value="Notification:(*)"/>	<input type="text" value=""/>	<input type="text" value=""/>
手順書URL	標準	<input type="text"/>	<input type="text" value=""/>	<input type="text" value=""/>
障害判定	標準	<input type="text"/>	<input type="text" value=""/>	<input type="text" value=""/>
対象Webサーバ	標準	<input type="text" value="Address:(*)"/>	<input type="text" value=""/>	<input type="text" value=""/>
エラー内容	標準	<input type="text" value="Service:(*)"/>	<input type="text" value=""/>	<input type="text" value=""/>
障害レベル	標準	<input type="text"/>	<input type="text" value=""/>	<input type="text" value=""/>
対象URL	標準	<input type="text"/>	<input type="text" value=""/>	<input type="text" value=""/>
復旧方法	標準	<input type="text"/>	<input type="text" value=""/>	<input type="text" value=""/>
原因	標準	<input type="text"/>	<input type="text" value=""/>	<input type="text" value=""/>

※連携イメージ

アラート内容を一意に判定する為に、事前に設定した“4つのキー項目” 文字列や\*等を設定します。

例) ここではプロジェクト名称、ホスト、トリガー名：プロBLEM・リカバーの通知区分“Ping監視、http監視”等を設定

## STEP 2 どのような処理をするか？



👉 処理したいコマンド登録（複数可）

👉 手順書URL情報を通知

👉 非処理時のコマンド登録（複数可）

**処理したい作業を記述します。**

**コマンド登録（複数可）や、利用する手順書のURL情報を登録します。**

（手順書はSHERPA-SMのWiki・文書にUPするとURLが表示され利用出来ます）

**また、日時を指定し通常の処理とは異なる処理（非処理）設定する場合に実行したいコマンド、手順書URLを設定することが出来ます。**

## STEP 3 フィルタリングをどうするか？

処理条件

処理フィルタ名	<input type="text"/>		
処理タイプ	都度	監視時間(分)	<input type="text"/>
処理契機(発生回数)	1	対象チケット	初回
イベントタイプ	障害	追い越し	NG
障害の更新情報	<input type="text"/>		

### 👉 処理フィルターを設定

- ◆ 都度：付加情報を付けて都度通知
- ◆ 重複：指定時間帯の同一アラート抑制
- ◆ 復旧：復旧報によるアラート抑制
- ◆ 繰延：期間繰延アラート抑制

### STEP 1 どのようなアラートが来たら？

アラートをキー情報で分別します。

※通知が重複しているかは見ていない



### STEP 3 フィルタリングをどうするか？

4つの処理タイプを使って重複等を制御します。





障害アラート



10:15

## 適応ルール“処 理”

【適応ルール】 時間帯 9:00 – 17:59  
担当 ジェニー  
手順書 障害手順書  
作業 コマンド入力

23:46

## 適応ルール“非処理”

【適応ルール】 時間帯 18:00 – 8:59  
【非処理】 担当 ジョン  
手順書 エスカレーション手順書  
作業 電話連絡

処理条件

処理フィルタ名

処理タイプ 復旧  監視時間(分) 20

処理契機(発生回数) 1  対象チケット 初回

イベントタイプ 障害  追い越し NG

障害の更新情報

非処理日時リスト

\* [0000-2359] ~ [0001-2400]

1	<input type="text"/>	18:00 ~ 08:59
2	毎日	<input type="text"/>
3	月曜~金曜	<input type="text"/>
4	土曜~日曜	<input type="text"/>
5	日曜日	<input type="text"/>
6	月曜日	<input type="text"/>
7	火曜日	<input type="text"/>
8	水曜日	<input type="text"/>
9	木曜日	<input type="text"/>
10	金曜日	<input type="text"/>
11	土曜日	<input type="text"/>
12	日曜日	<input type="text"/>

曜日や時間帯を考慮し、  
通常の処理をしない設定

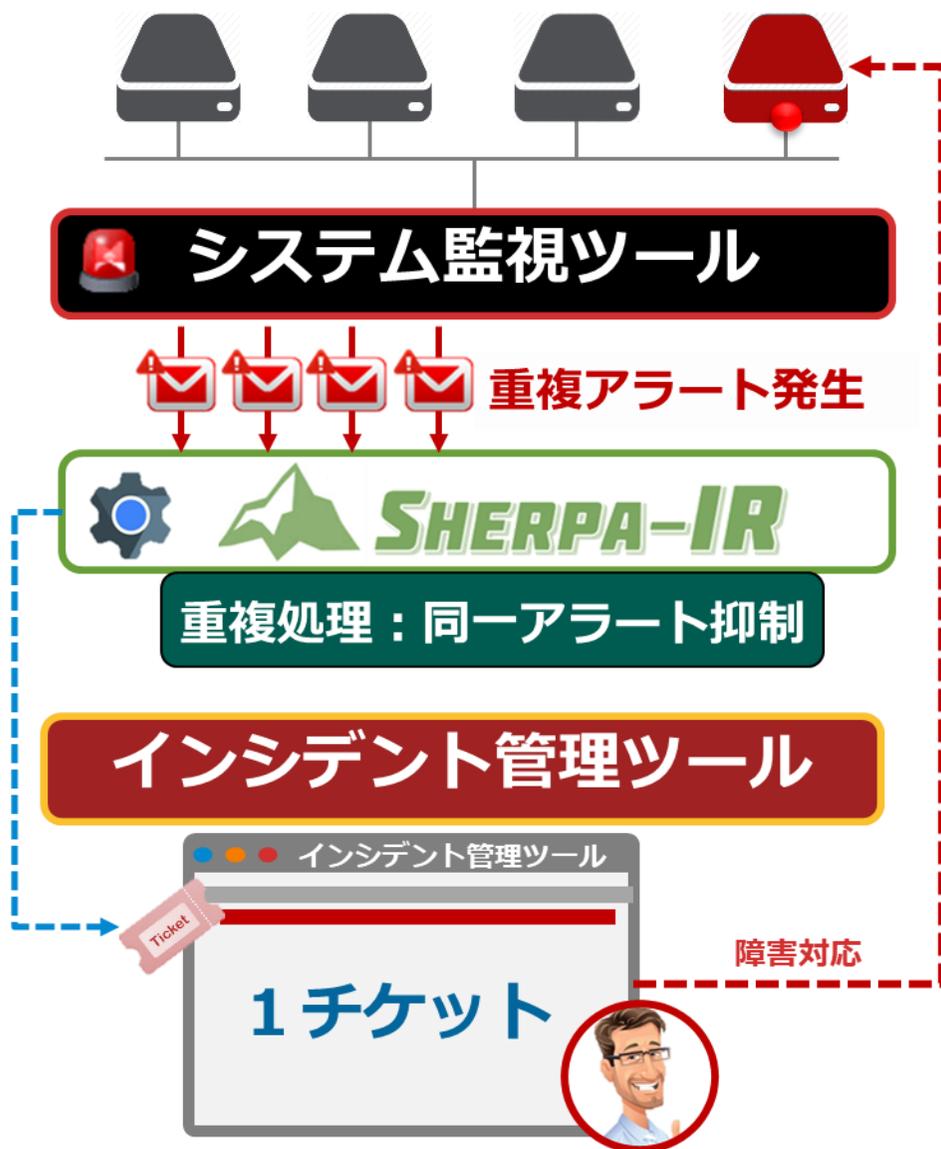


## 都度処理

重要な障害で都度、担当者や手順書情報を付加して通知したい場合利用します。

## 都度処理対応のメリット

1. 重要アラートを都度通知
2. 担当者通知で対応漏れの削減
3. 手順書リンクで初動時間の短縮
4. ミスの削減



## 重複処理

同一のアラートが“指定した時間帯”に“指定回数”通知された場合に、インシデント登録を行います。

## 重複処理対応メリット

1. アラート内容確認から解放
2. 重要アラート見落とし削減
3. ミスの軽減
4. サービスレベルの均一化



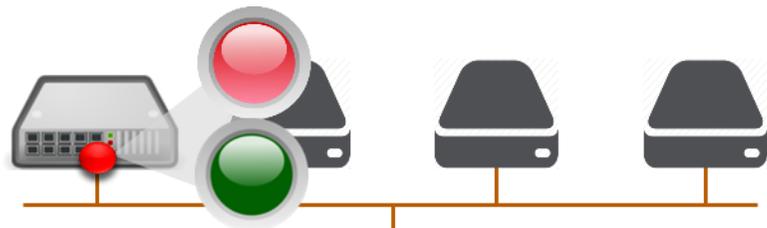
## 繰延処理

既に障害対応作業に取掛っていても、障害復旧していなければ、設定時間をすぎると新たにインシデントが作成されてしまいます。

繰延処理は、指定した時間内に同一のアラートが通知された場合、指定時間のタイマーをクリア（繰延）し、制御を継続することができます。

## 繰延処理対応メリット

1. 作業時間を気にすることなく、障害対応に専念できる



## システム監視ツール



リンクダウン通知  
リンクアップ通知



修復処理ルール適応

Timer



## インシデント管理ツール



## 復旧処理

復旧処理は、対象機器からの“障害報”と“復旧報”を考慮する処理タイプです。LinkDown/LinkUP等ネットワーク機器で、“障害報”が通知された場合、一定時間“対”となる“復旧報”を静観する場合があります。復旧処理では“障害報”が来ても直ぐにチケット作成指示を出さず、一定時間“復旧報”を持ち、通知されれば障害報を無視し、通知が無ければチケット作成指示を実施します。

## 繰延処理対応メリット

1. “対”となるアラート待ちからの解放
2. 不要チケットの消込作業削減



## システム監視ツール



メンテナンス時間帯の為  
アラートを見逃す



メンテ対応

12月24日  
00:00~08:59

## インシデント管理ツール



## メンテナンス時処理

メンテナンス時のアラート制御は、“指定機器”及び“指定時間帯”を非処理機能を利用して行います。指定時間帯のメンテナンス機器からのアラートは無視されます。メンテナンス時間帯でも、指定されていない機器からのアラートは、通常の制御として処理されます。

## メンテナンス時処理メリット

1. 大量不要アラートからの解放
2. 不要チケットの消込削減

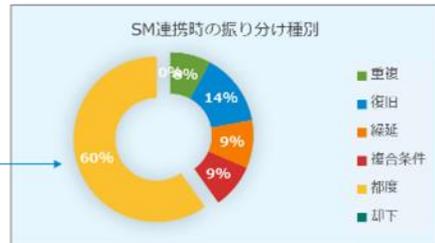
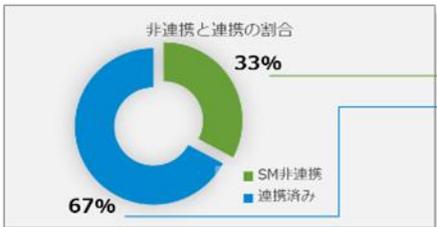
# SHERPA-IR機能：レポート



## 単月の登録内容



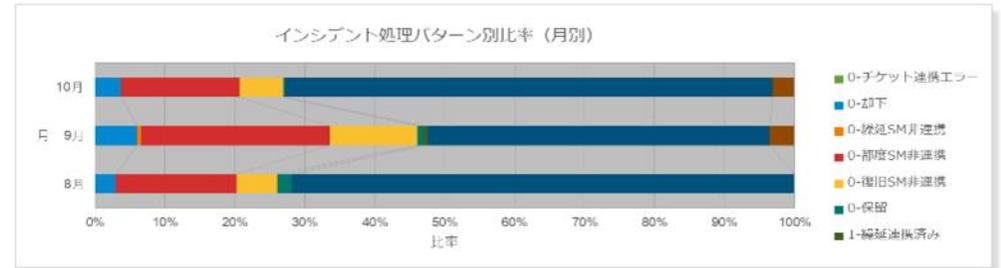
## 単月のIR処理数とSMへの連携数 (下のグラフは3割ほどIRで処理)



※ 処理タイプ「都度」が良く使われている。  
 ※ 連携すべきインシデント化の精査が必要。

## 3か月のIRで利用している処理タイプ別割合

サマリ	列ラベル								
行ラベル	0-チケット連携エラー	0-却下	0-繰延SM非連携	0-都度SM非連携	0-復旧SM非連携	0-保留	1-繰延連携済み	1-都度	
8月		7		41	14	5		17	30
9月		62	6	277	130	9	6	503	1000
10月	1	58		268	98	4		1100	49
総計	1	127	6	586	242	18	6	1774	86
									2846



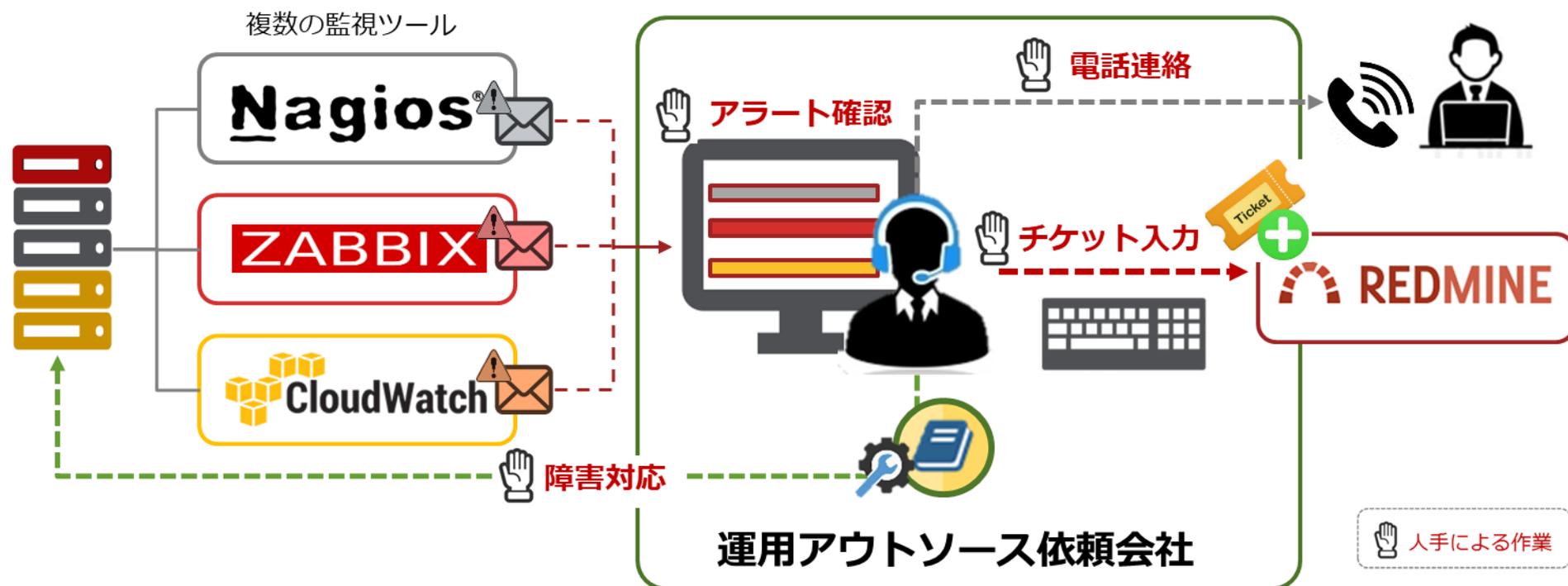
## SHERPA-IR Reporterを利用して



処理タイプと処理件数等情報から運用改善の仮説検証へ

# ユーザ事例：概要

複数の監視ツールが導入され、異なるアラートフォーマットの為、インシデント管理ツールに自動起票が出来ず、オペレータがアラート内容を確認しチケット起票を行った上で、手順書に従い障害対応を実施していた。



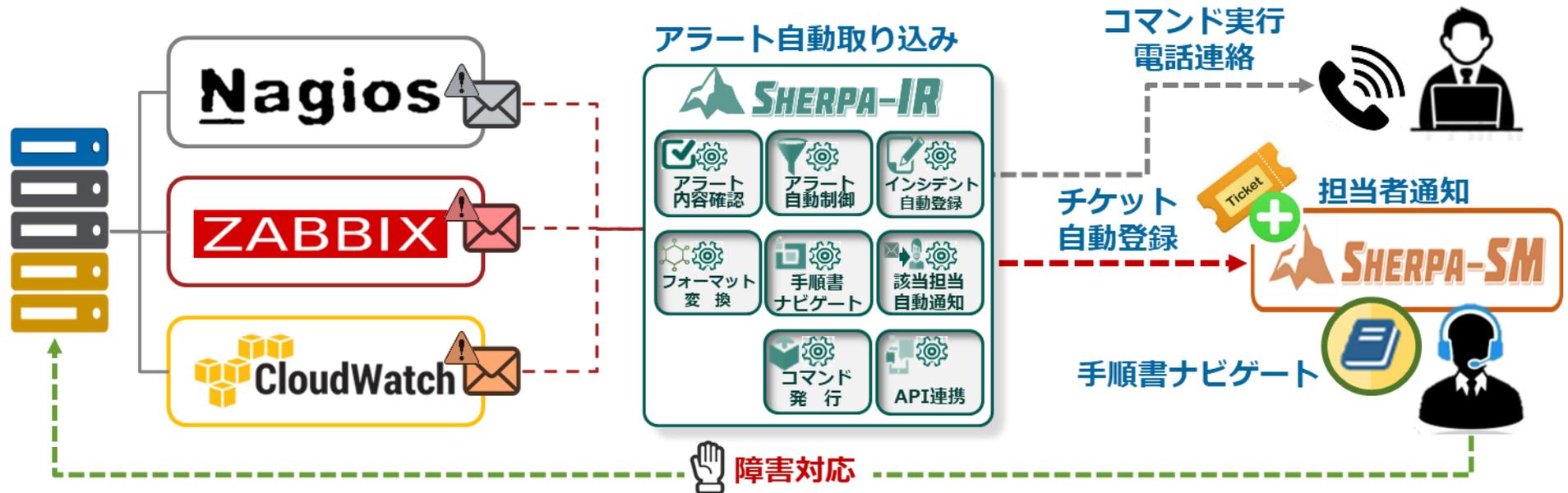
自動化を含めた運用改善を始めたきっかけは・・・

## 運用アウトソース先よりサービス費用アップの要求



# ユーザ事例：新作業の流れ

人手による処理（アラート内容の判別、手順書内に記載の文字列抽出起票、/実施コマンド確認）を、SHERPA-IRにルール設定し自動実行することで、漏れのない迅速な障害対応を実現。



複数ツールからのアラート取込み

手順書確認負荷の軽減

対応速度の向上

運用ミスの低減

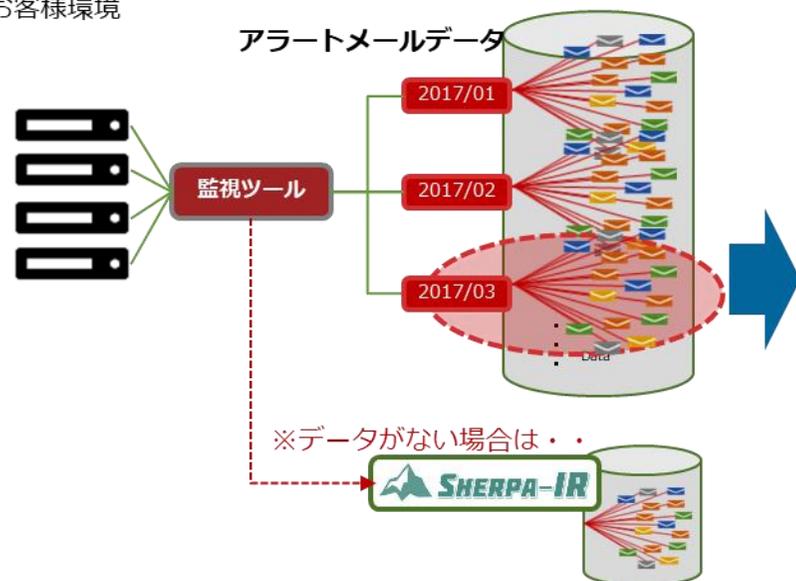
APIを使用した運用の自動化

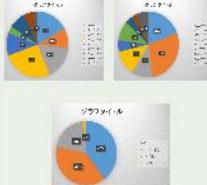
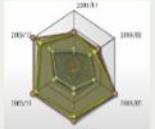
レポート作成時間の軽減

# ユーザ事例：SHERPA-IR導入アセスメント

実際のお客様のアラートメールデータを分析して、SHERPA-IR導入の効果がありそうかの予測

お客様環境



区分	サンプリング	分類	ルールパターン	削減効率																
作業	一定期間メールインシデントをもらう又は、自社SHERPA-IRに飛ばしてインシデントを蓄積	障害報および復旧報が発生するインシデントを特定	定期的に発生しているインシデントを見つけ、重複等ルールが使えるか予測	IR導入による削減予想レポート また、障害対応の自動化による削減予測																
アウトプット		インシデント分類 <table border="1"> <thead> <tr> <th>分類</th> <th>件数</th> <th>割合</th> <th>件数</th> </tr> </thead> <tbody> <tr> <td>障害報</td> <td>10</td> <td>10%</td> <td>200</td> </tr> <tr> <td>復旧報</td> <td>10</td> <td>10%</td> <td>200</td> </tr> <tr> <td>定期メンテナンス</td> <td>10</td> <td>10%</td> <td>200</td> </tr> </tbody> </table>	分類	件数	割合	件数	障害報	10	10%	200	復旧報	10	10%	200	定期メンテナンス	10	10%	200	IRルール適応分類 	導入効果予測 
分類	件数	割合	件数																	
障害報	10	10%	200																	
復旧報	10	10%	200																	
定期メンテナンス	10	10%	200																	



## 利用目的

1. 導入効果予測の可視化
2. 社内稟議 説明資料

SHERPA-IR導入アセスメントレポート（有料）



# ユーザ事例：要求事項の確認と処理タイプへのリンク

お客様の作業条件を確認し、SHERPA-IRのルールに落とし込んでいきます。

- 重複 -> “tx”を含むホスト名の場合、重複制御
- 都度 -> “app”を含むホスト名の場合、都度電話連絡
- 復旧 -> “spice”を含むホスト名のHTTPで20分以内にリカバリを検知した場合、電話通知不要

## 重複

```
件名：** PROBLEM Service Alert: tx-ap01/
Application Log - API is CRITICAL **
***** Nagios *****
Notification Type: PROBLEM
Host: tx-ap01
Alias: ap01.tx.aws.inb-xxxxx .jp
Address: 10.16.20.10
Service: Application Log - API
State: CRITICAL
Date/Time: Mon Nov 21 15:23:01 JST 2016
```

## 都度

```
件名：** PROBLEM Service Alert: cmf-app01/Application
Log is CRITICAL **
Host: cmf-app01
Alias: app01.cmf.aws.inb-xxxxx .jp
Address: 10.16.46.10
Service: Application Log
State: CRITICAL
Date/Time: Sat Jul 22 11:57:44 JST 2017
Additional Info:
CRITICAL - (2 errors) - 22-Jul-2017 11:55:39.807
SEVERE [ajp-nio-8009-exec-1353]
org.apache.catalina.core.StandardHostValve.custom
Exception Processing
ErrorPage[exceptionType=java.lang.Exception,
location=/WEB-INF/views/common/error/unhandledSystemError.ht
ml] ...
```

## 復旧

```
件名：** PROBLEM Service Alert: [xxxxx ]spice-api/HTTP is
CRITICAL **
***** Nagios *****
Notification Type: PROBLEM
Host: [xxxxx ]spice-api
Alias: [xxxxx ]spice-api
Address: spice-native-api.xxxxx .jp
Service: HTTP
State: CRITICAL
Date/Time: Sat Jul 22 12:08:23 JST 2017
↑↓ 相対するメール
件名：** RECOVERY Service Alert: [XXXXX ]spice-api/HTTP is OK **
***** Nagios *****
Notification Type: RECOVERY
Host: [xxxxx ]spice-api
Alias: [xxxxx ]spice-api
Address: spice-native-api.xxxxx .jp
Service: HTTP
State: OK
Date/Time: Sat Jul 22 12:13:13 JST 2017
```

# ユーザ事例：IRルールへの落とし込み

どのようなアラートが来たらどの処理（フィルタ含む）をするかを整理し、SHERPA-IRのルールに設定していきます。

システム監視項目一覧

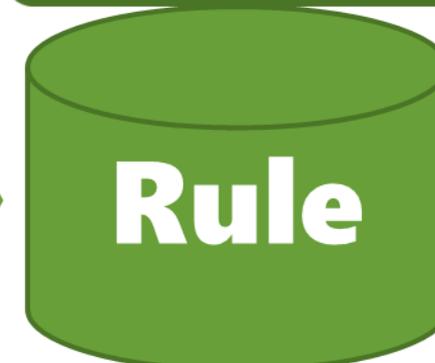


アラートメールサンプル ①、②、③、④

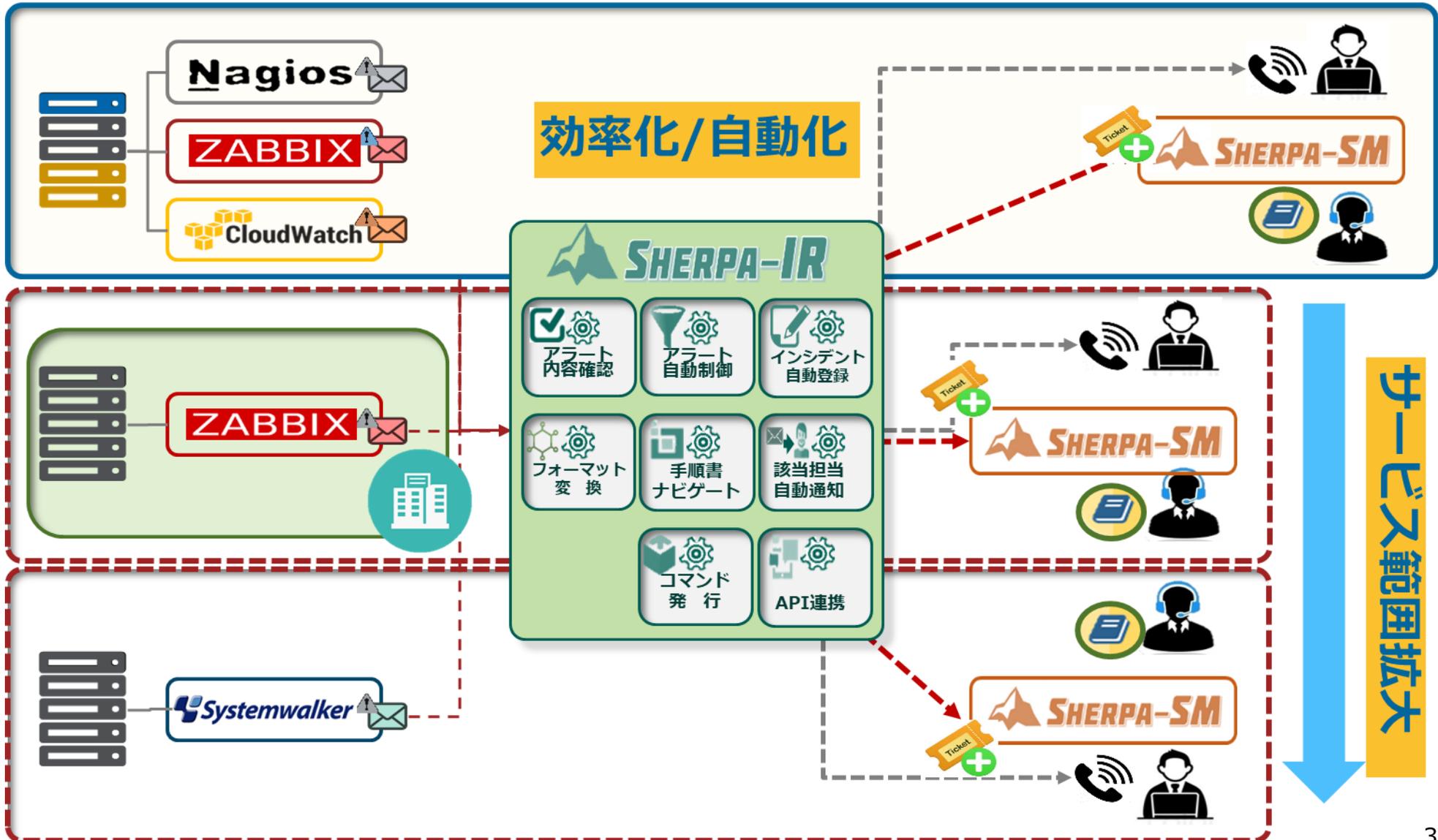


フィルタールール

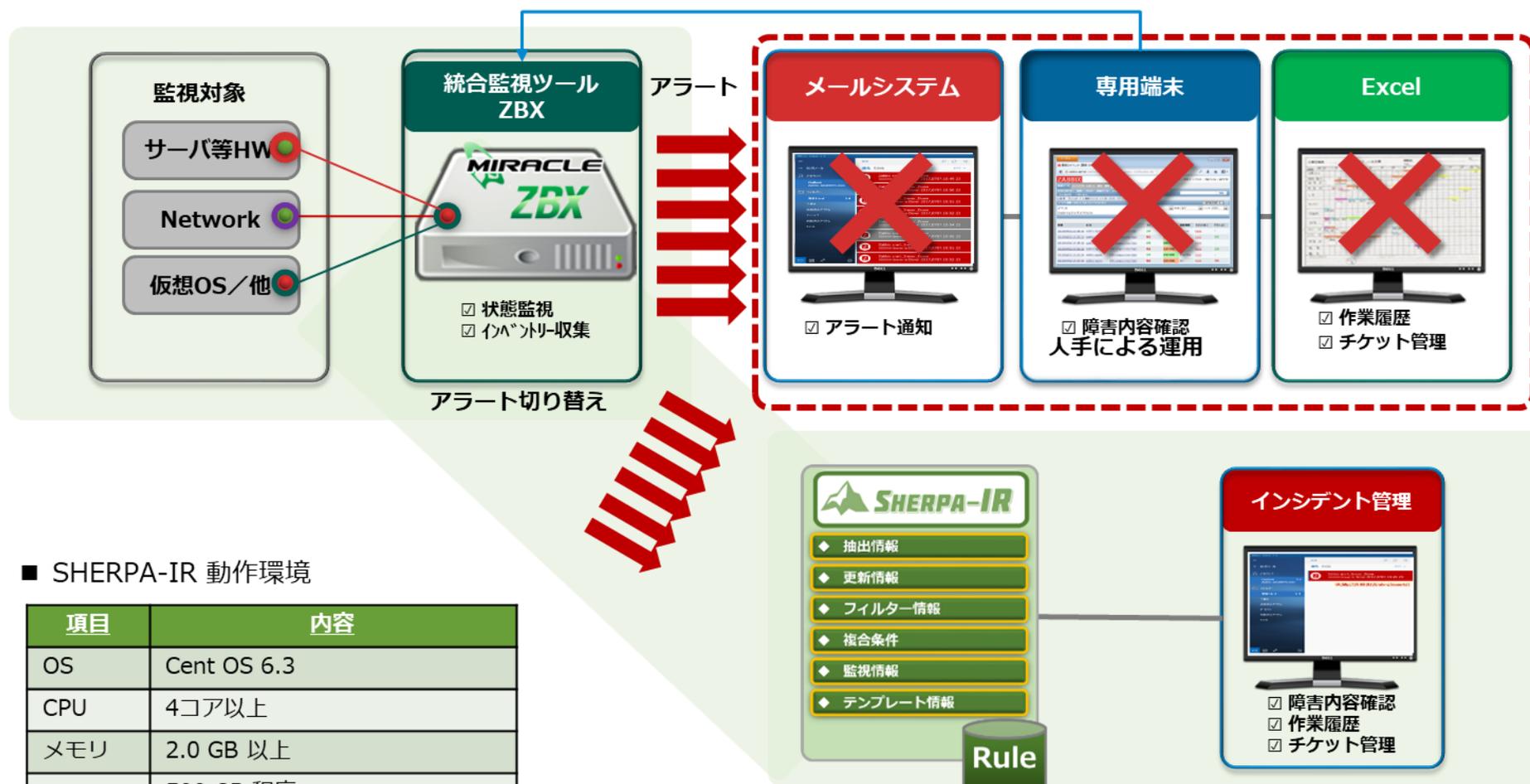
ホスト名	サービス名	監視内容	アラート対応	アラート発生時の対応	サンプル種別	処理タイプ
cmf-app01	Application Log	アプリケーションログ	アプリ	チケット作成	サンプル① チケット作成のみ	都度
tx-dbr01	MySQL	MySQLサービス	インフラ	チケット作成	サンプル② 要電話通知	都度 (電話通知)
man-mx02	System Load	ロードアベレージ	インフラ	チケット作成 & AWシステム担当へ電話エスカレーション	サンプル③ 要電話通知(重複)	重複
spice-dbm01	Disk Usage	ディスク空き容量	インフラ	チケット作成 & AWシステム担当へ電話エスカレーション	サンプル④ チケット作成のみ(復旧)	復旧
tx-ap	Application Log - API	アプリケーションログ	アプリ	チケット作成 更に30分で3回以上発生した場合のみアプリ担当へ電話連絡		重複
spice-api	HTTP	HTTPサービス	インフラ	お客様にも電話連絡する。 20分でリカバリを検知した場合システム担当者を含め電話連絡不要		復旧
spice-push01	Memory Usage	使用可能メモリ容量	インフラ	チケット作成		重複+復旧



# ユーザ事例：効率化/自動化の実現



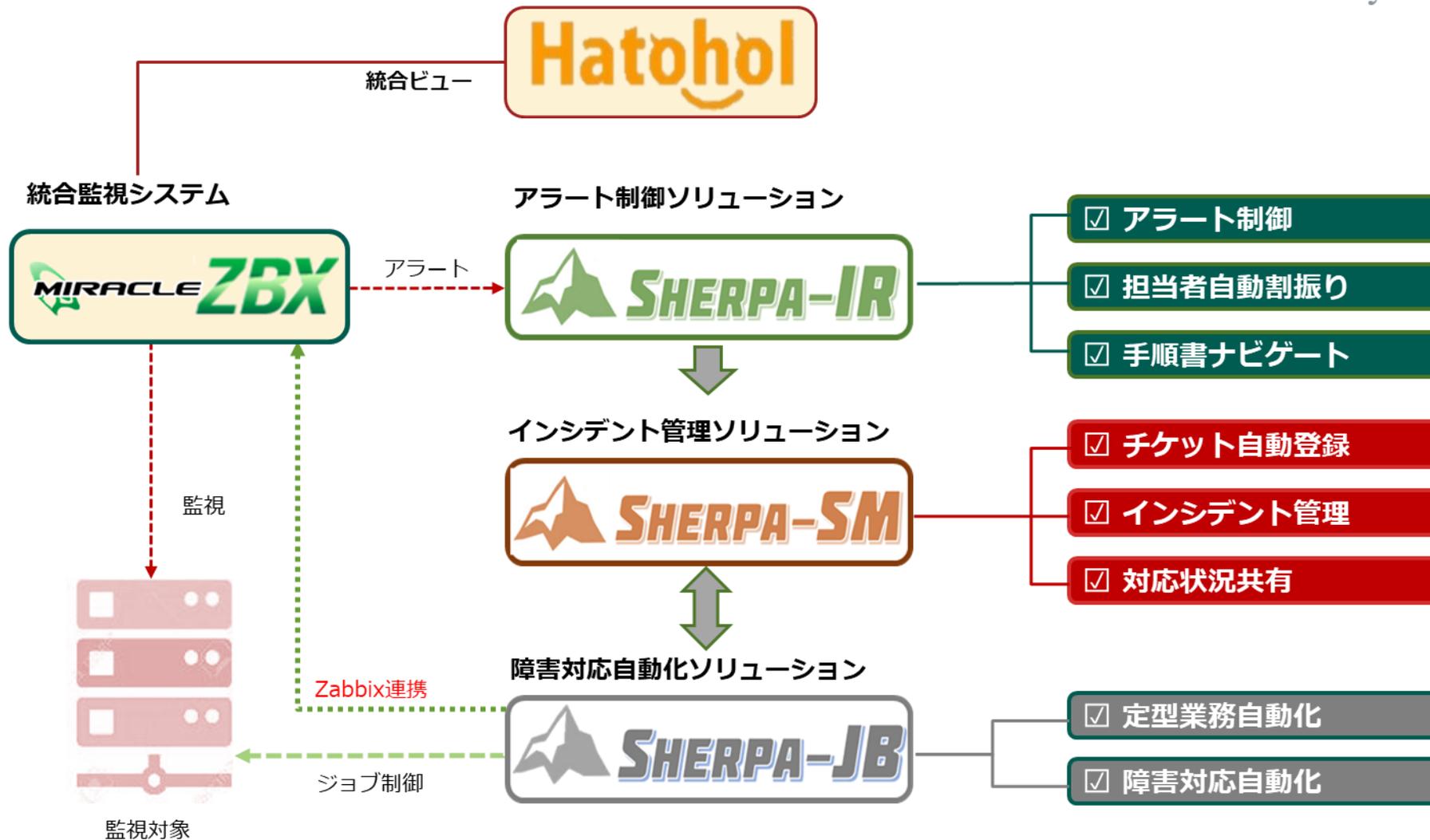
導入は大きなシステムの変更は必要ありません。



## ■ SHERPA-IR 動作環境

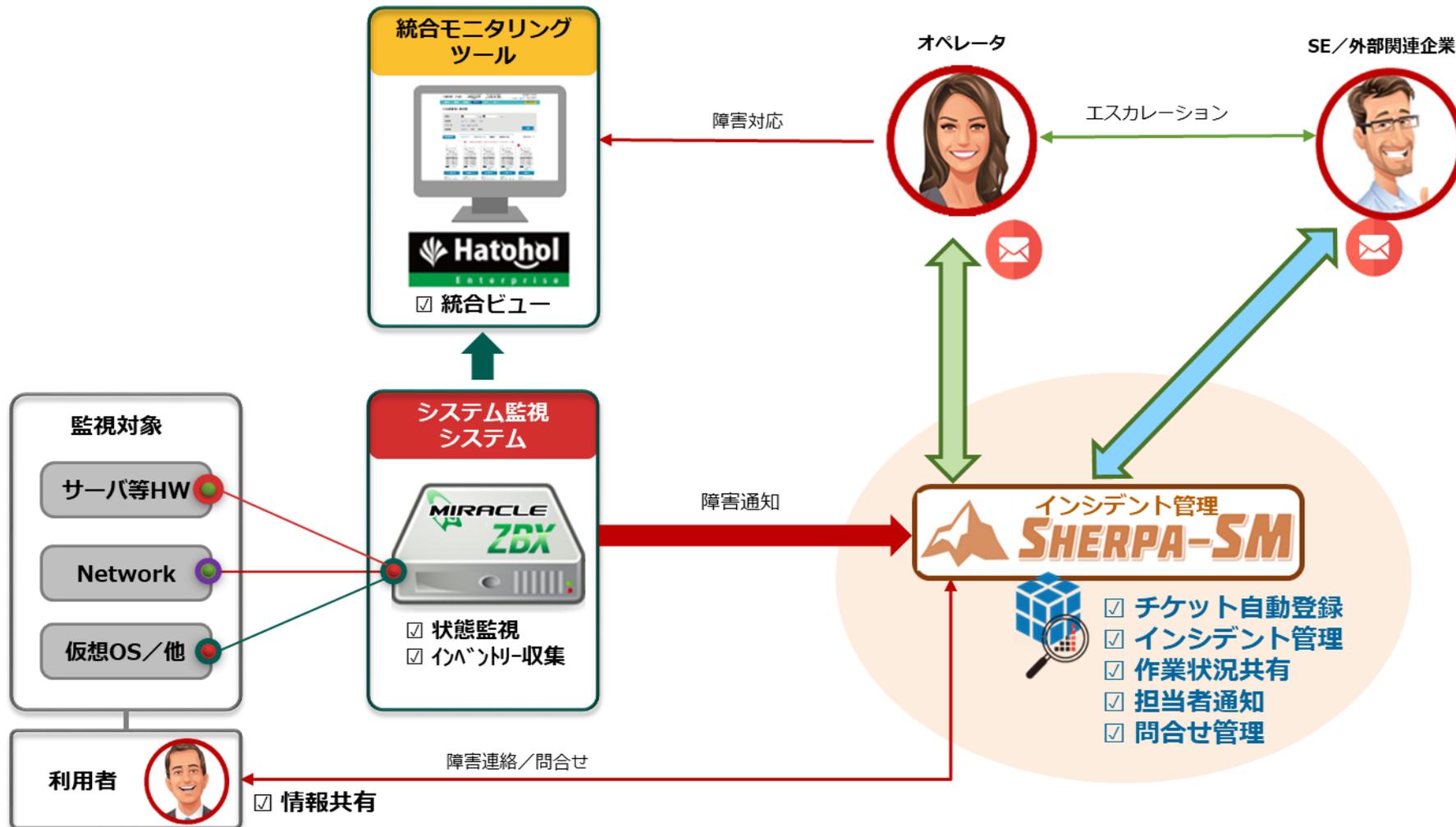
項目	内容
OS	Cent OS 6.3
CPU	4コア以上
メモリ	2.0 GB 以上
DISK	500 GB 程度 (添付容量等により異なります。)

# ご提供ソリューションMAP



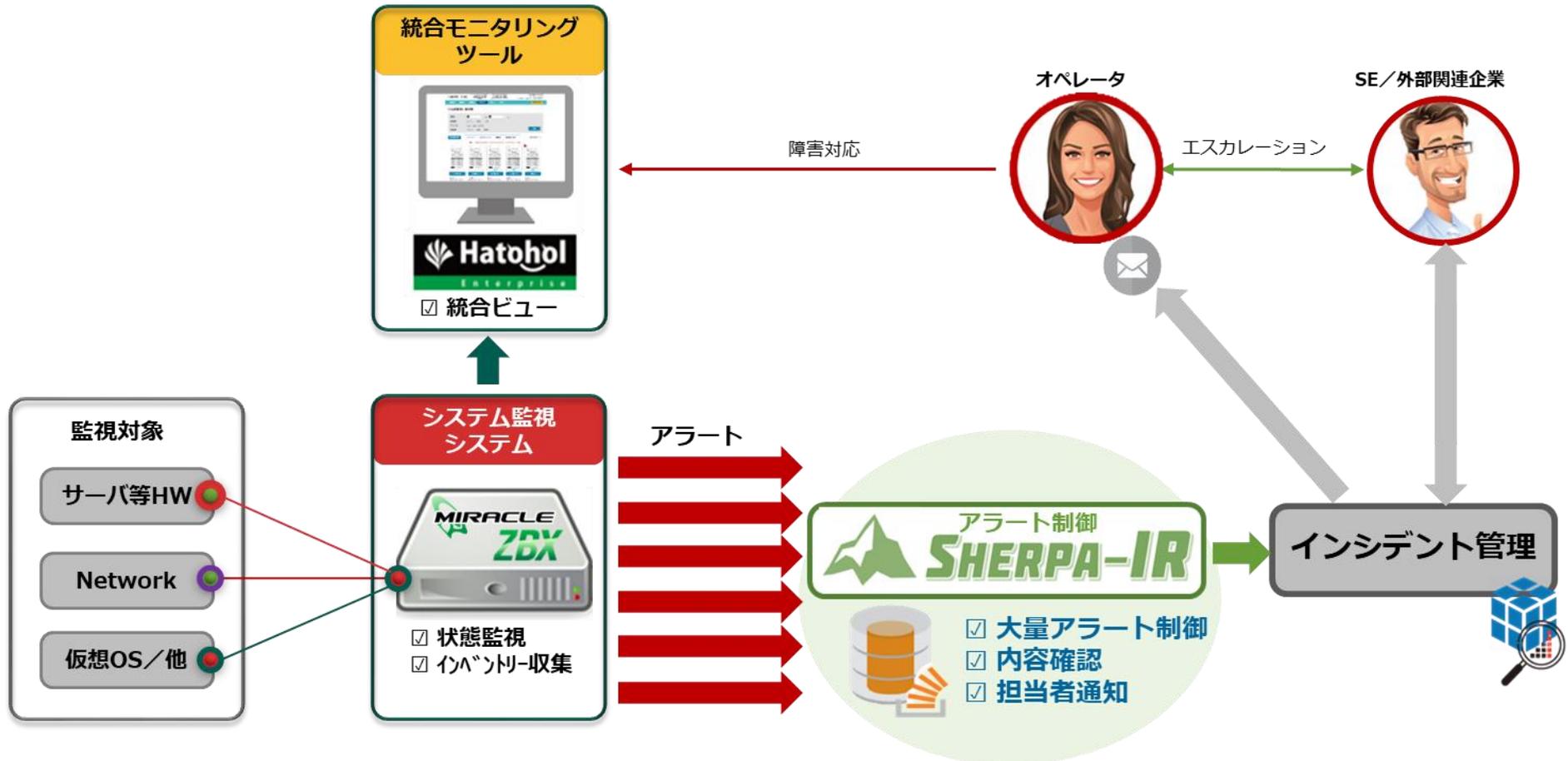
## 出来るところから運用業務の自動化を始めましょう！

## ✓ インシデント管理でお困りの方



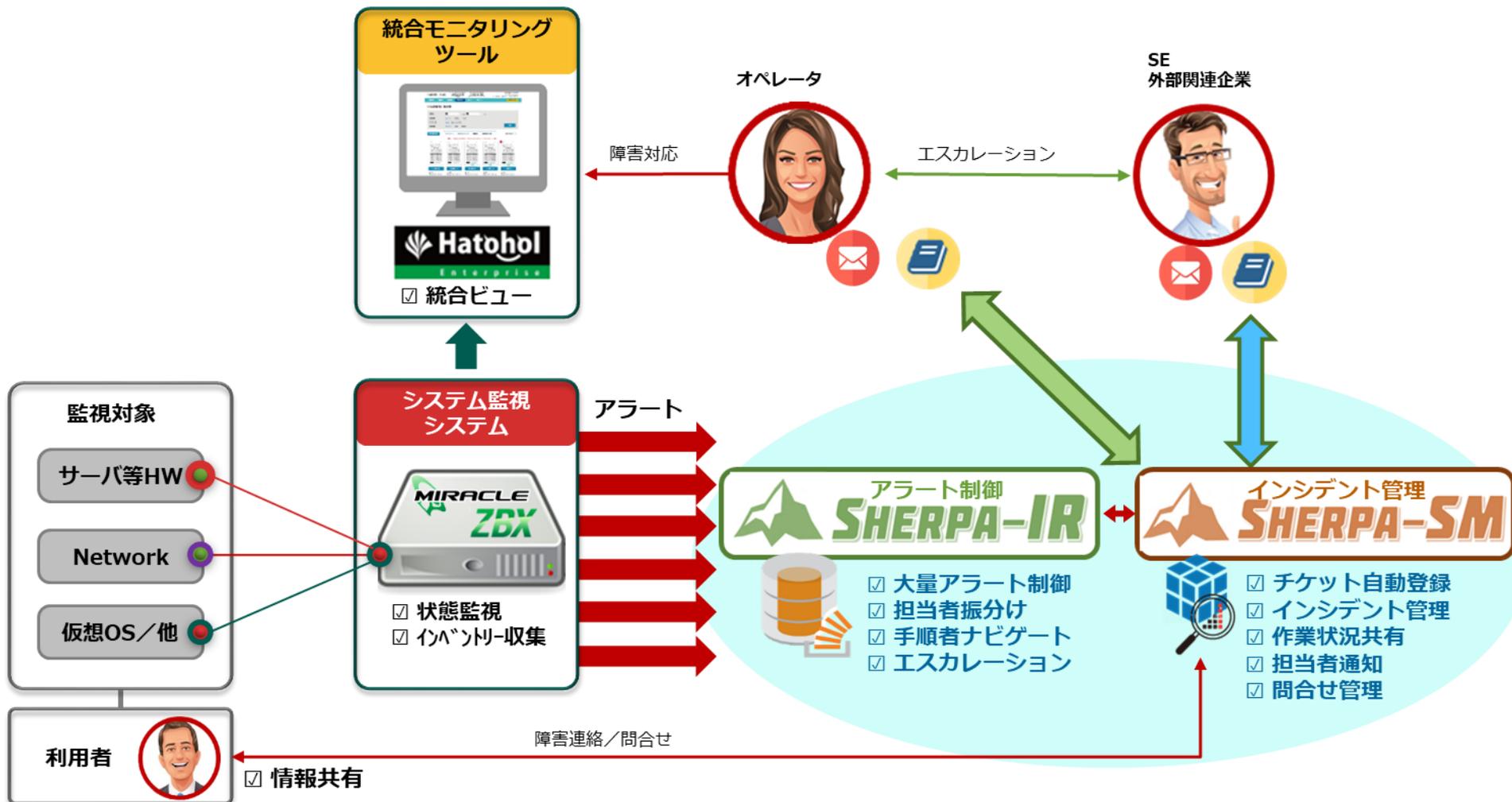
障害対応状況把握・共有で運用品質の向上のご提案をします。

## ✓ 監視ツールからのアラート制御でお困りの方



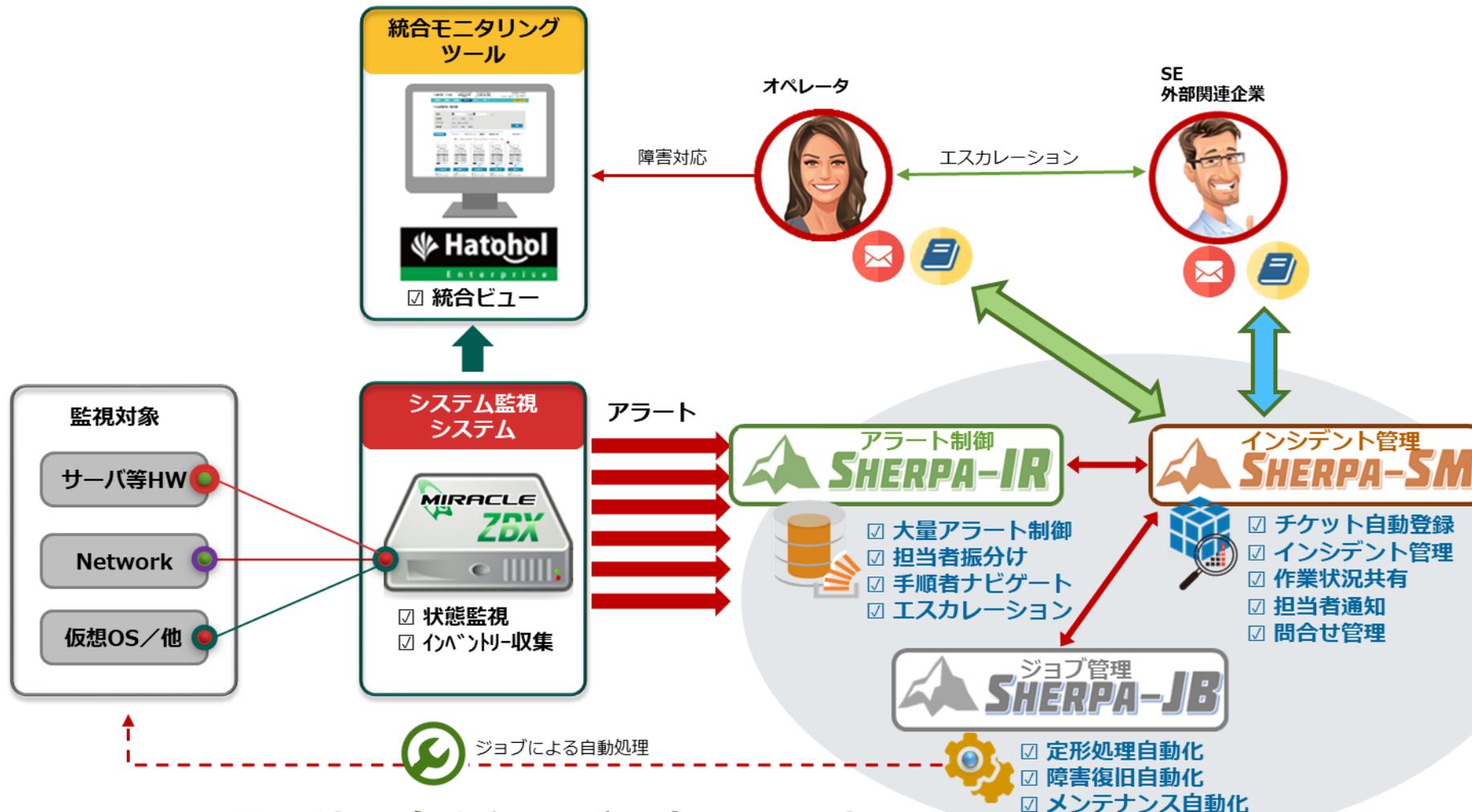
大量アラート制御で、1次オペレータ作業削減のご提案をします。

## 1 次オペレータ作業の効率化でお困りの方



運用品質の向上と運用リソースの最適化のご提案をします。

## ✓ システム運用の効率化でお困りの方



運用全体の自動化のご提案をします。

# END

# ご清聴ありがとうございました