# Zabbixアラートのインシデント管理ツール連携と、RPAによる運用の自動化・効率化

cybertrust

# 運用管理をとりまく環境

cybertrust


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**





公開

# 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。**

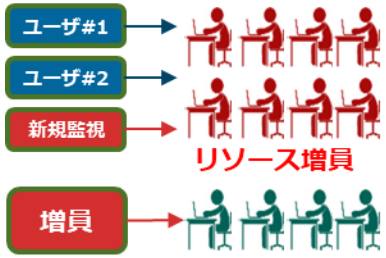
断念してしまう理由
“ITIL” 呪文に掛かって運用改善に取り生み出すと・・・・
1．全てを一度に取り組まないと成果が出ないのではないか？
2．“ITIL”で書かれていることを、具体的に何をすればよいのか?

# 運用業務の自動化（RPA）で期待できる効果


## 1.業務の効率化


## 2.人的ミス削減

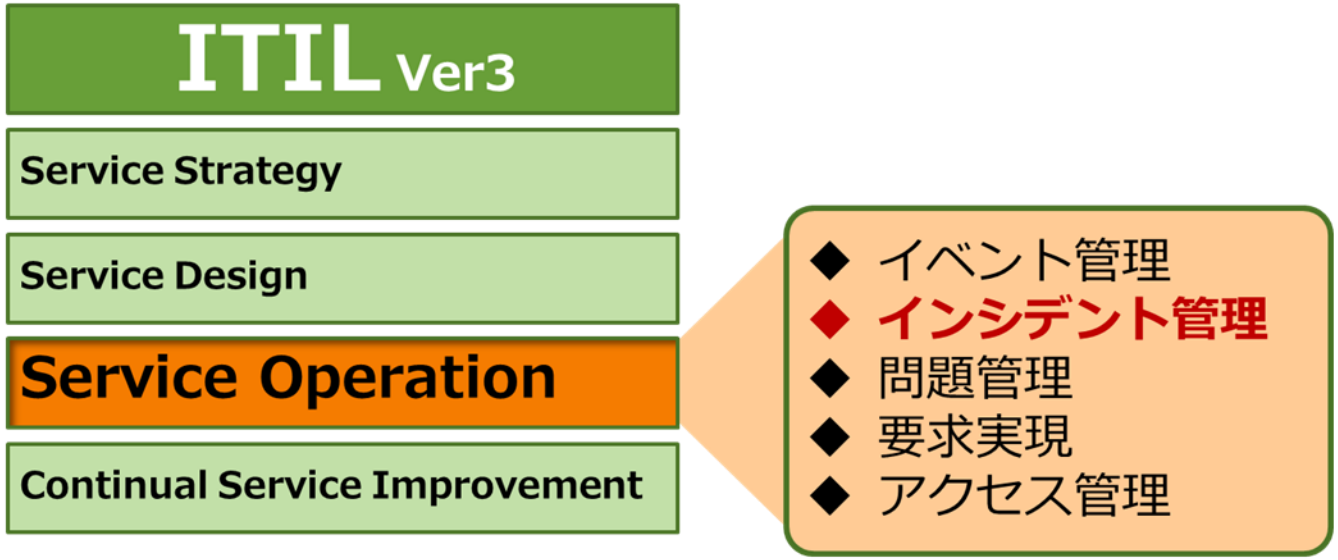

## 3.運用プロセスの定着




## 4.サービス拡大に対応

# インシデント管理

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“




**Zabbixから発生するアラートを、如何にスムーズに**
**インシデント管理と自動連携して行くのが良いのか？**



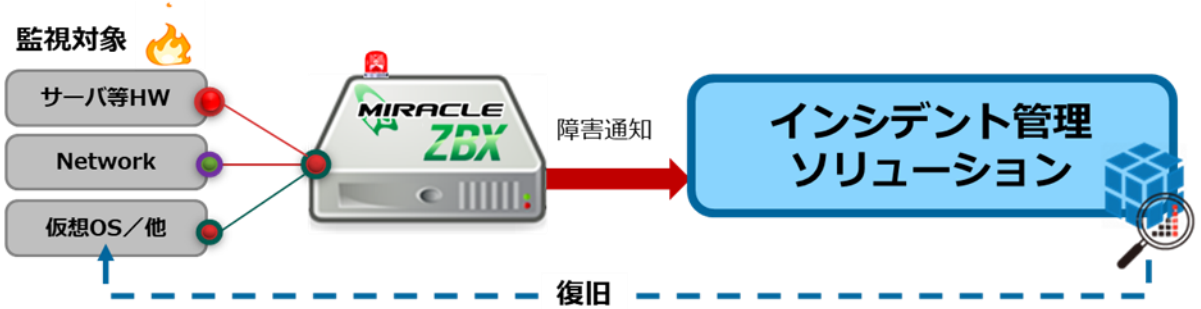


公開

# インシデント管理とは

## 目　的

**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

**シンプルではありますが、確実に行うことは大変です。**

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**

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**

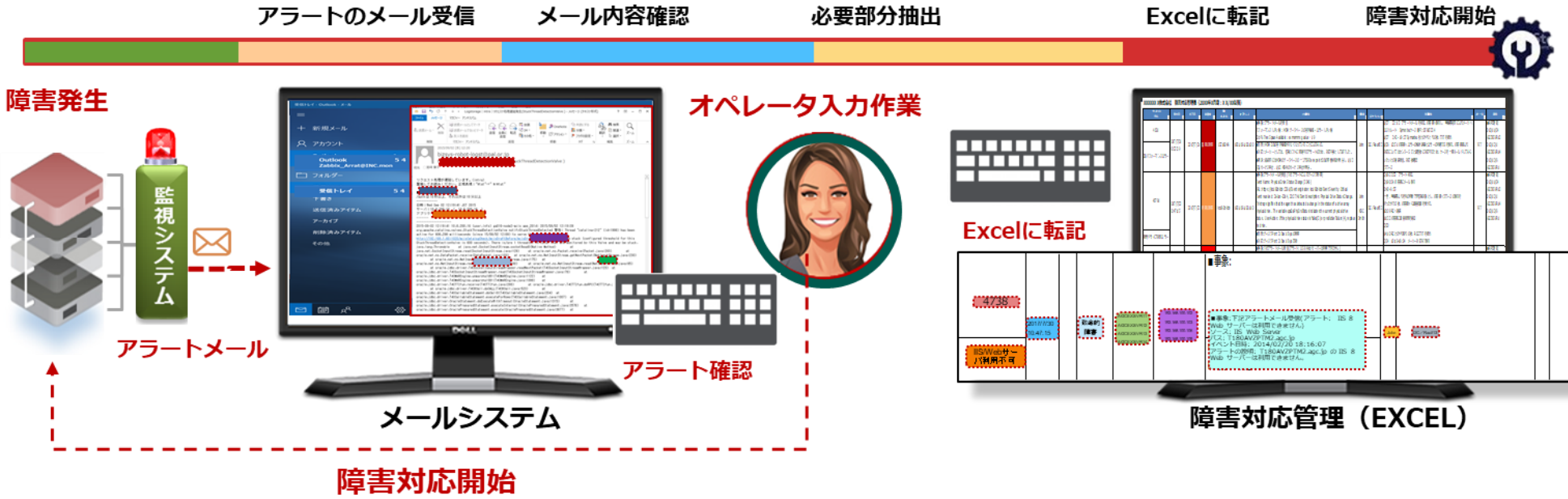




公開

# 実際に現場は・・・・

## 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

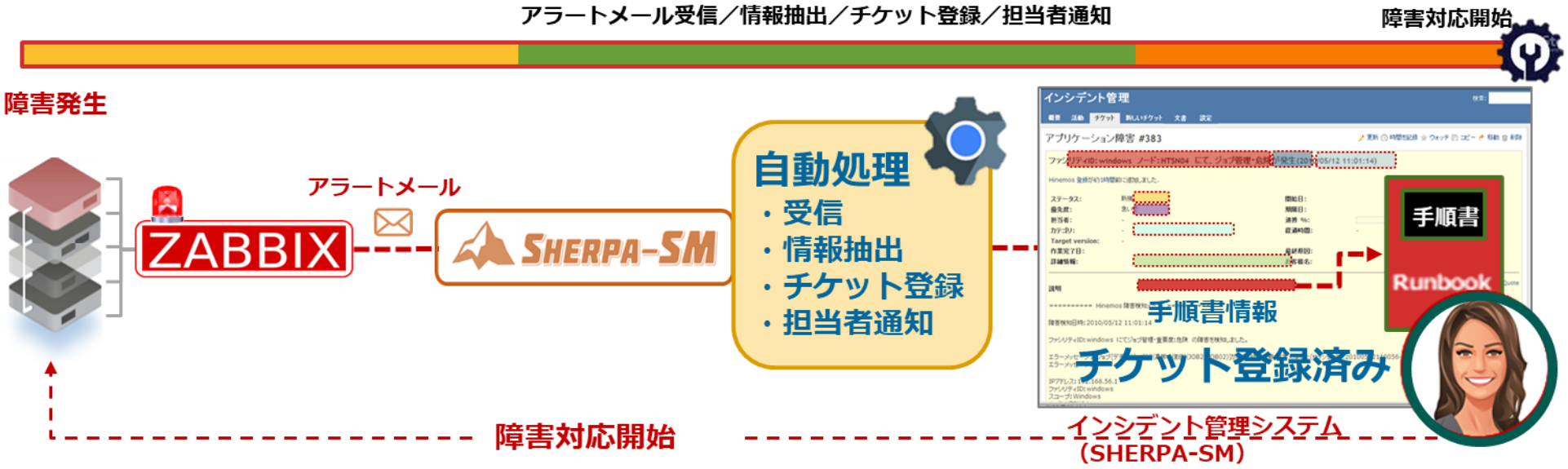



## 【課 題】

- ✓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

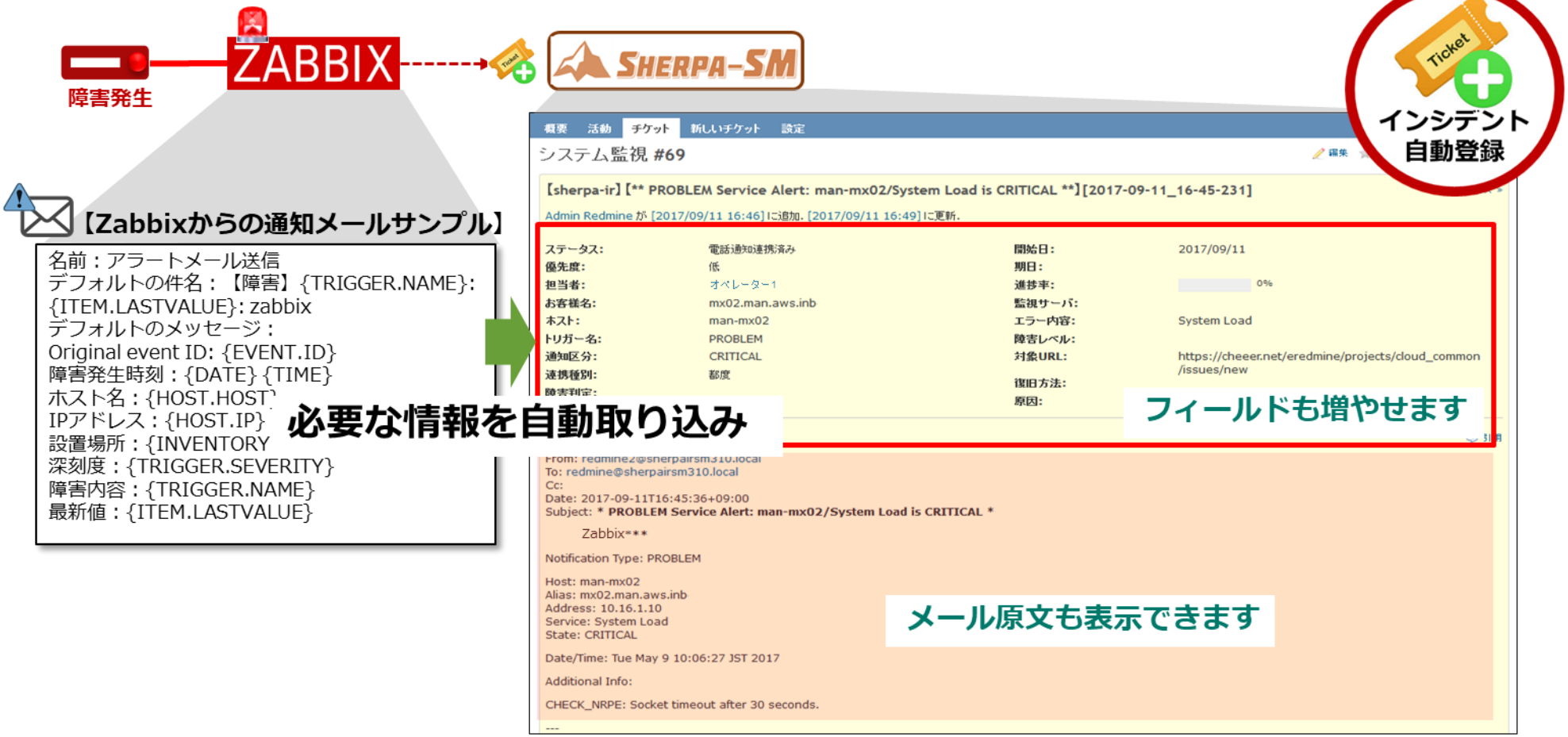



【効果】

- ✓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（譲り合っての対応遅延防止）
- ✓全ての対応状況把握と進捗状況の把握

# Zabbixからインシデント自動登録

## 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




# 記入漏れや情報不足などのミスを防止

cybertrust

# マイページ


**自分の担当案件がひと目で把握でき**
**対応漏れも防止**

cybertrust

# 優先度表示

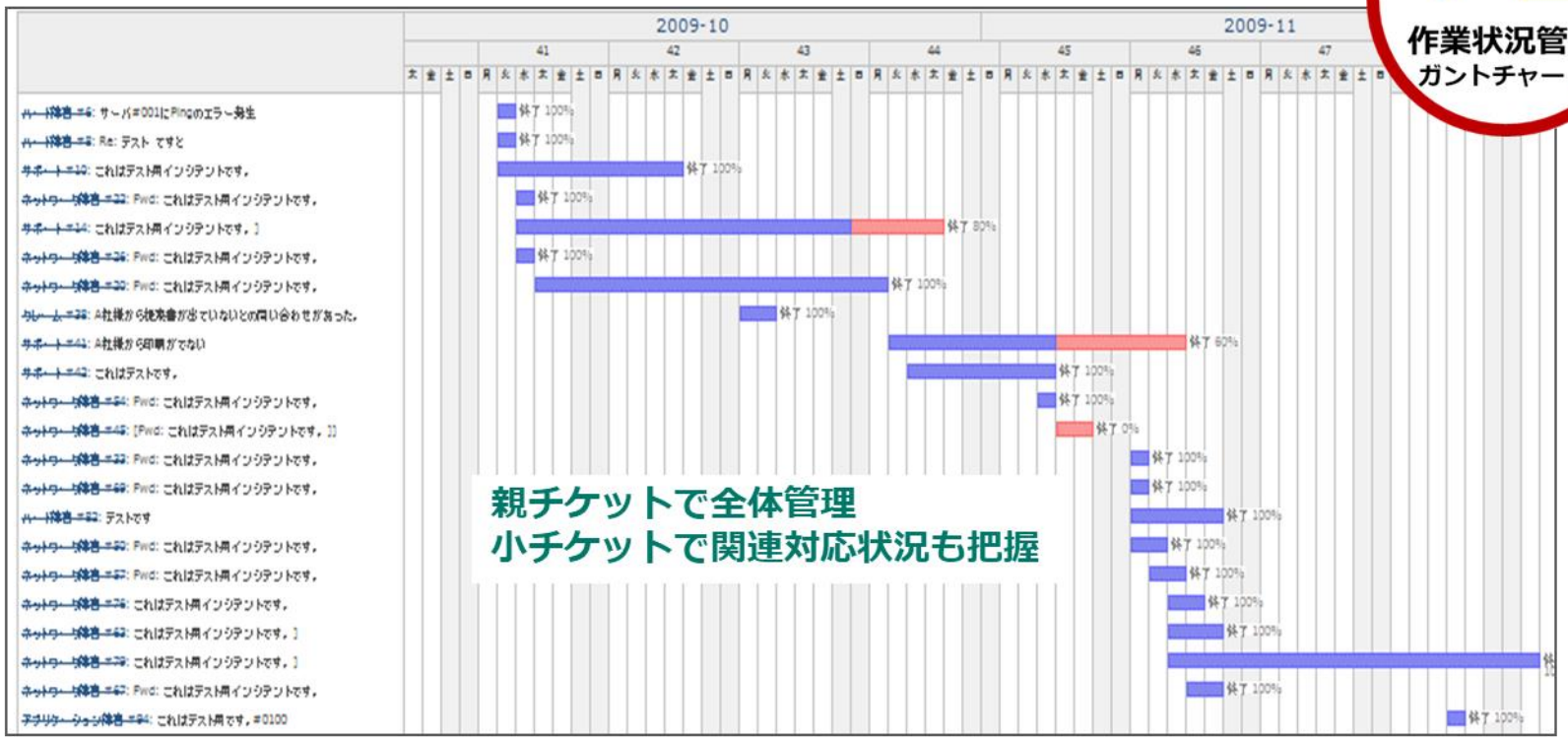

| ✔ | # | トラッカー | ステータス | 優先度 | 題名 | 起票者 | 担当者 | 更新日 |
|---|---|---|---|---|---|---|---|---|
| ☐ | 383 | アプリケーション障害 | 新規 | 急いで | ファシリティID: windows ノード: HTSN04 にて、ジョブ管理・危険が発生(2010/05/12 11:01:14) | Hinemos 登録 | | 2010年 月12日 11:01 A |
| ☐ | 382 | アプリケーション障害 | 新規 | 急いで | ファシリティID: windows ノード: HTSN04 にて、ジョブ管理・危険が発生(2010/05/12 10:49:45) | Hinemos 登録 | | 2010年05月12日 10:49 AM |
| ☐ | 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05月12日 10:38 AM |
| ☐ | 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05月11日 18:19 PM |
| ☐ | 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05月11日 18:19 PM |
| ☐ | | | | | | | 管理 運用 | 2010年05月11日 18:19 PM |
| ☐ | 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05月11日 18:19 PM |
| ☐ | 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05月11日 18:19 PM |

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

**各障害対応が、どの程度急を要するものの見極めも必要**

cybertrust

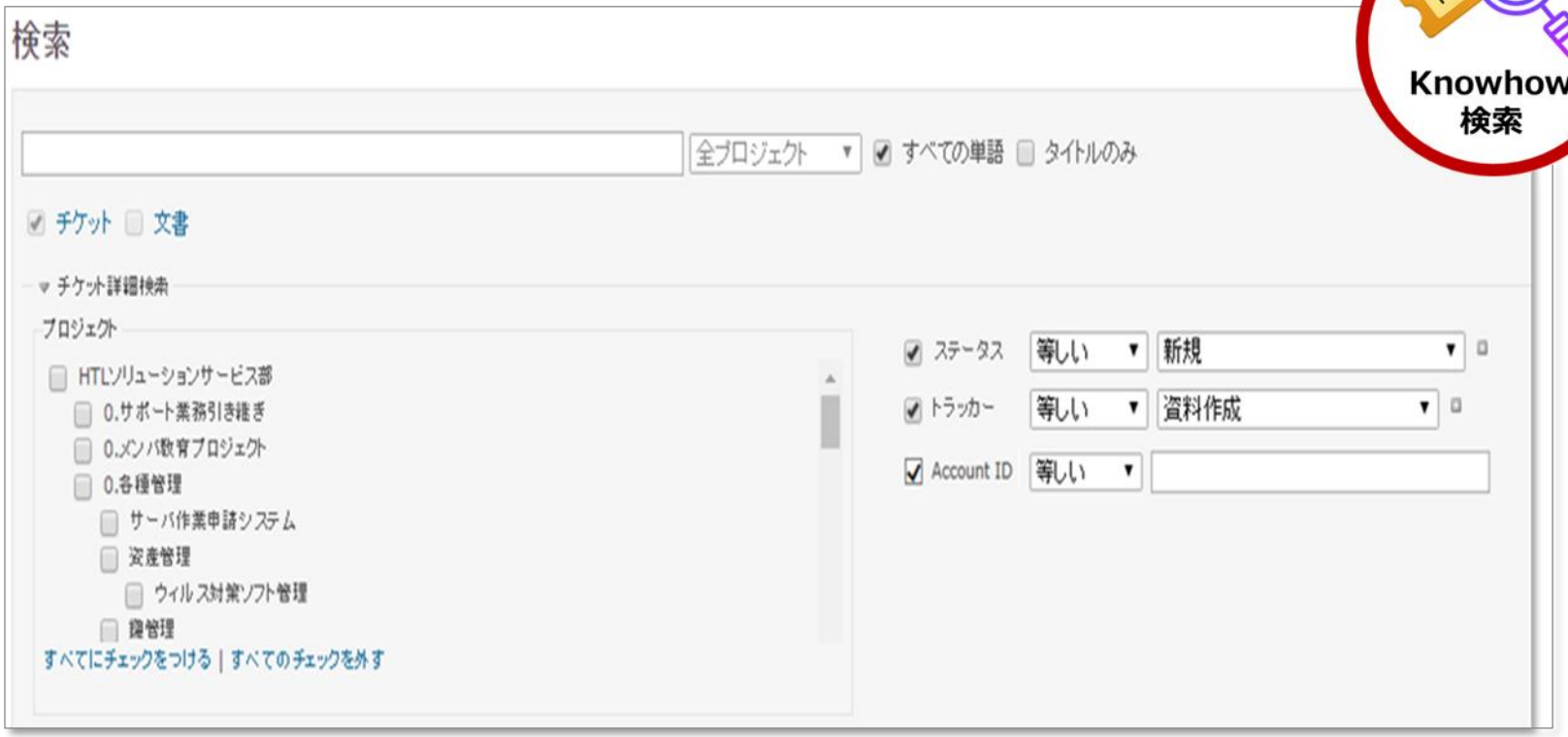
# ガントチャート




# リアルタイムに対応状況を把握

cybertrust

# 検索＆詳細絞り込み



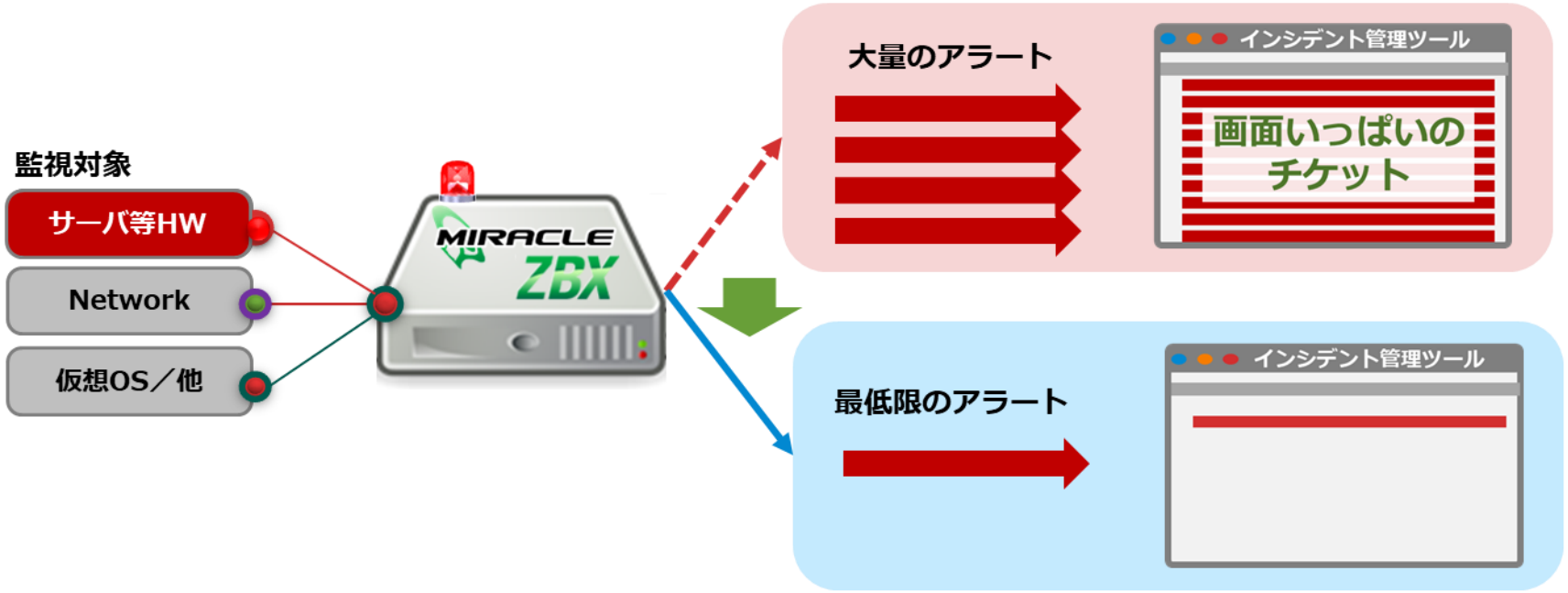


# 対応の履歴を共有することで
# 障害対の復旧時間（MTTR)を短縮

# 課題を解決するには・・・・

## インシデント管理をうまく回すにあたっての重要なポイント

- ✓ 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。



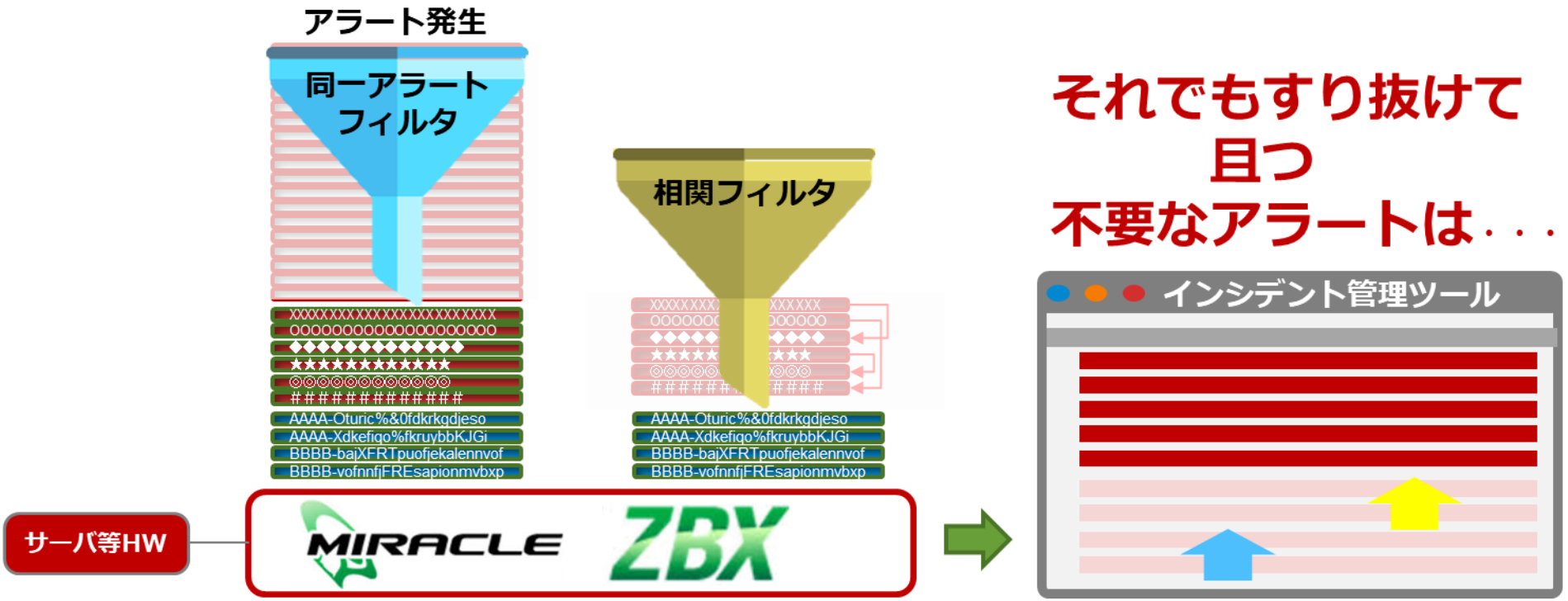

## 具体的には何をべきなのか・・・・・・？

- ➢ インシデント管理に登録する前に“アラートのフィルターを！

# アラートフィルター：Zabbix

## 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



# アラートフィルター：SHERPA-IR（アラート制御ツール）

## 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

- ◆都　度
- ◆重　複
- ◆復　旧
- ◆繰　延

✓プラスアルファ（担当者判断、手順書URL、コマンド発行等）

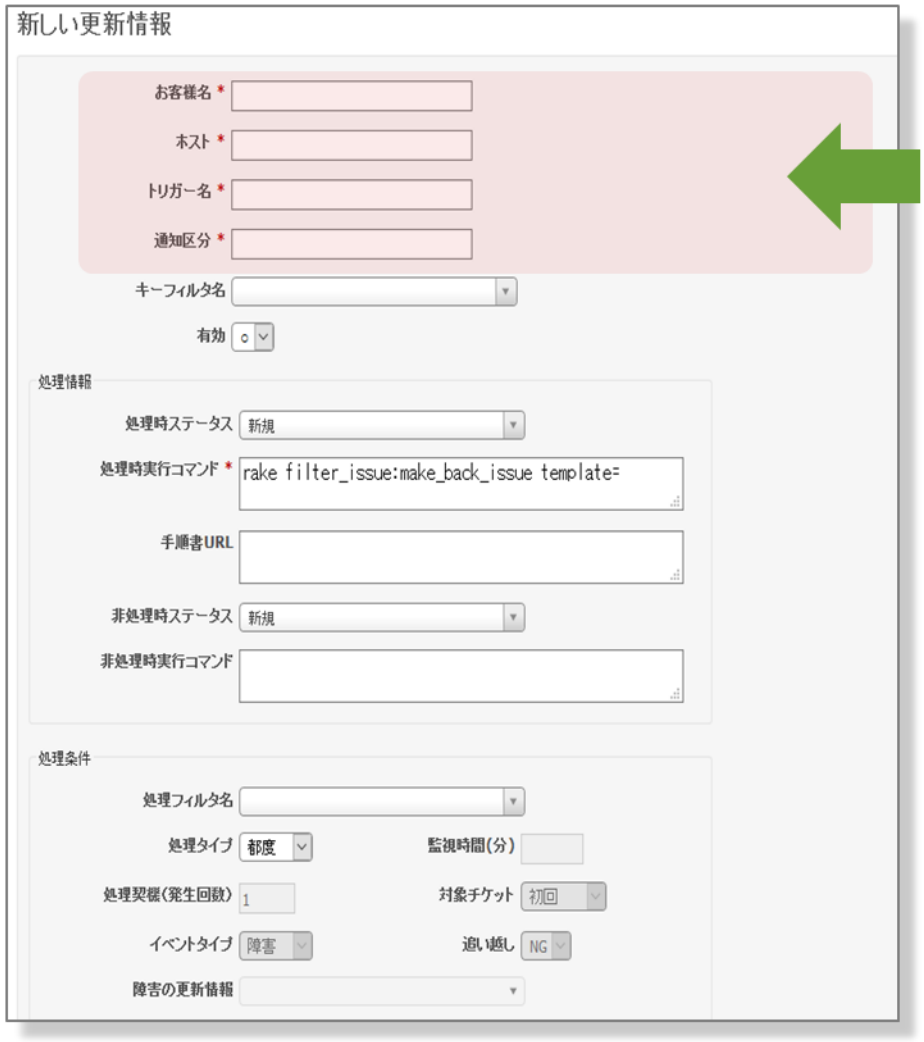





公開

# SHERPA-IR　設定の流れ

## STEP1　どのようなアラートが来たら？




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.LOCATION}
深刻度：{TRIGGER.SEVERITY}
障害内容：{TRIGGER.NAME}
最新値：{ITEM.LASTVALUE}

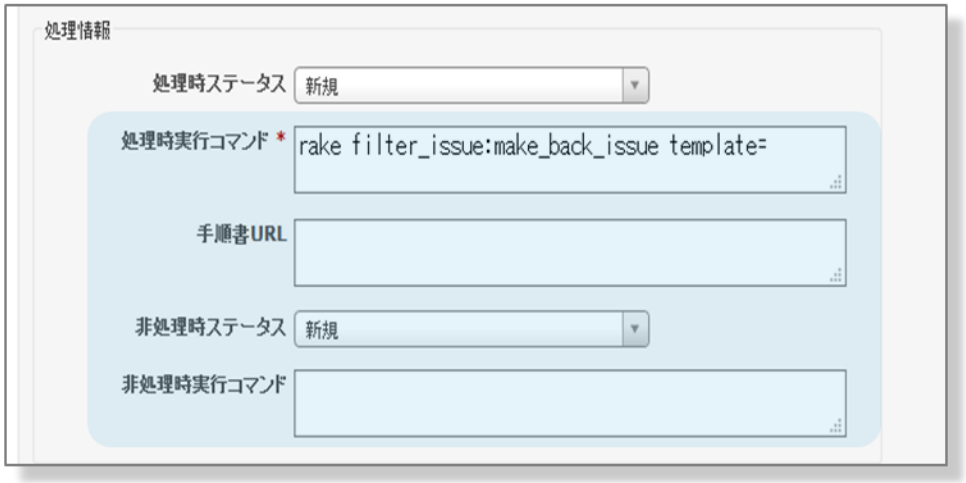
**アラート電文（フォーマット）**

**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＊等を使った正規表現にて設定します。**

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

# SHERPA-IR　設定の流れ

## STEP２ 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UR情報を登録します。**
（手順書はSHERPA-SMのWiki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。**

# SHERPA-IR　設定の流れ

## STEP３フィルタリングをどうするか？


☜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

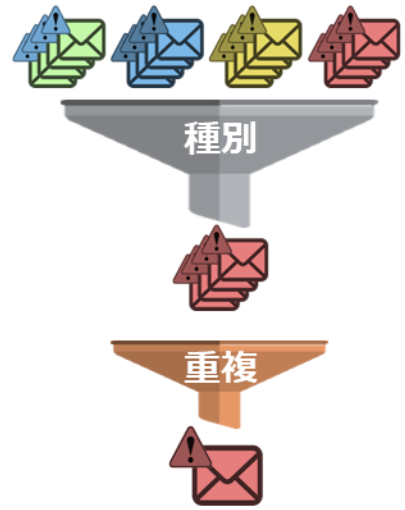
**STEP1　どのようなアラートが来たら？**

アラートを種別を判別します。

※通知が重複しているかは見ていない

**STEP３　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を制御します。

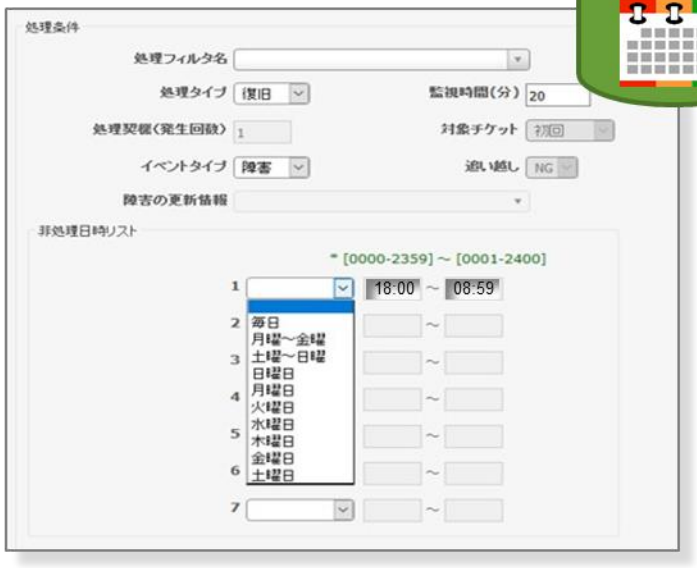


# 非処理の設定

障害アラート

SHERPA-IR

Rule




**曜日や時間帯を考慮し、**
**通常の処理をしない設定**

10:15

23:46

適応ルール“処　理”


【適応ルール】

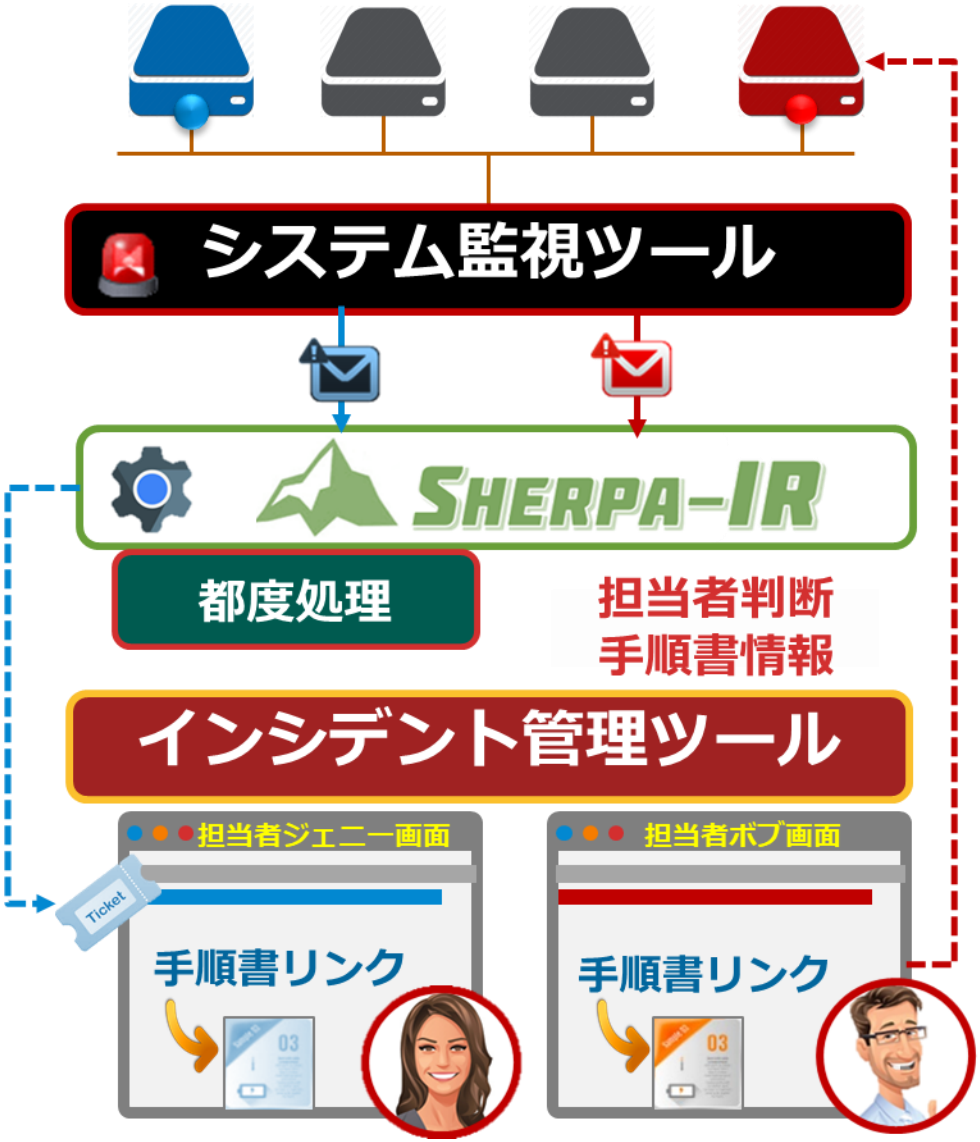
| | |
|---|---|
| 時間帯 | 9:00－17:59 |
| 担当 | ジェニー |
| 手順書 | 障害手順書 |
| 作業 | コマンド入力 |

適応ルール“非処理”


【適応ルール】
【非処理】

| | |
|---|---|
| 時間帯 | 18:00－8:59 |
| 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 |

# 都度処理



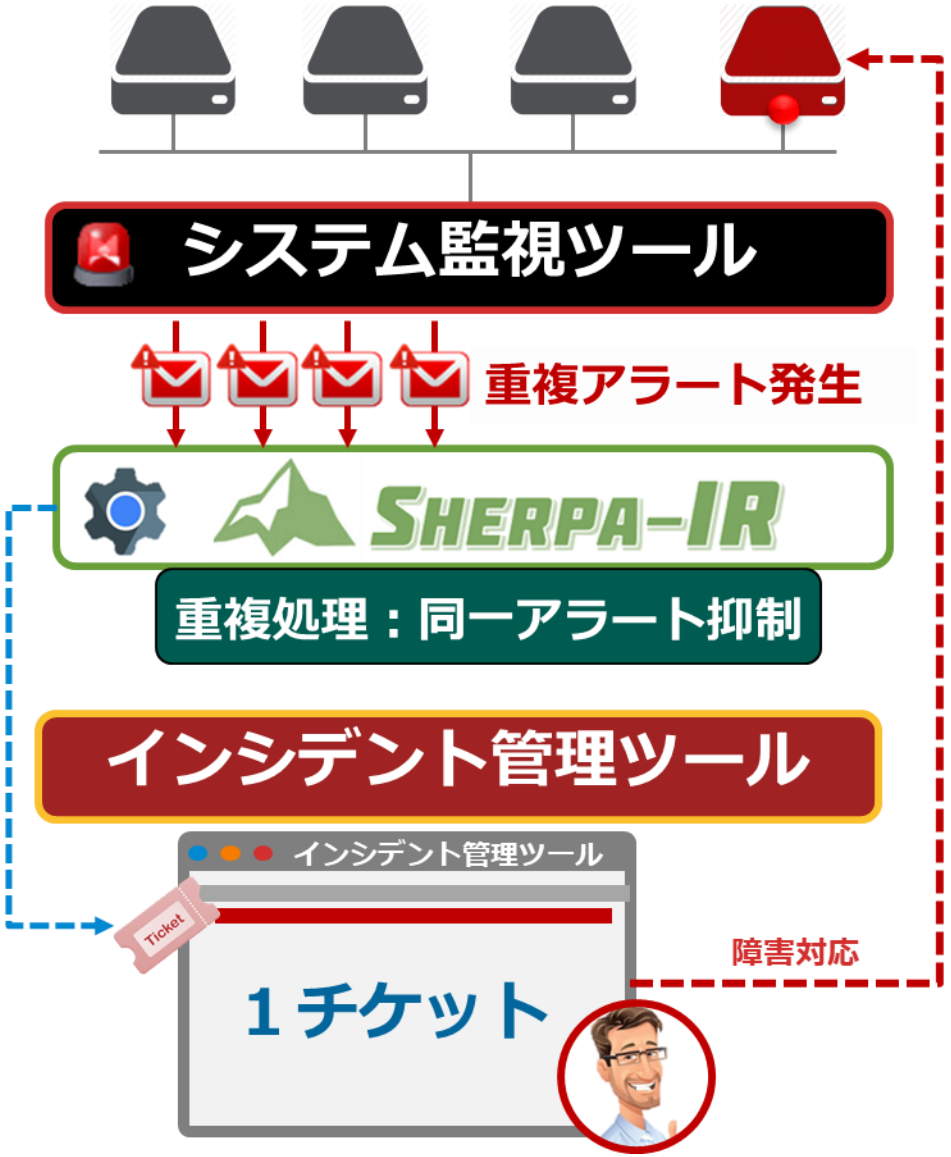

## 都度処理

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




## 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　（作業対応中の同一アラート制御）


## 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、新たにインシデントが作成されてしまいす。
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ー（繰延）し、制御を継続することが出来ます。

## 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

# 復旧処理


## 復旧処理

復旧処理は、対象機器からの“障害報”と“復旧報”を考慮するフィルターです。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

# メンテナンス時間帯の制御　　非処理


## メンテンンス時処理

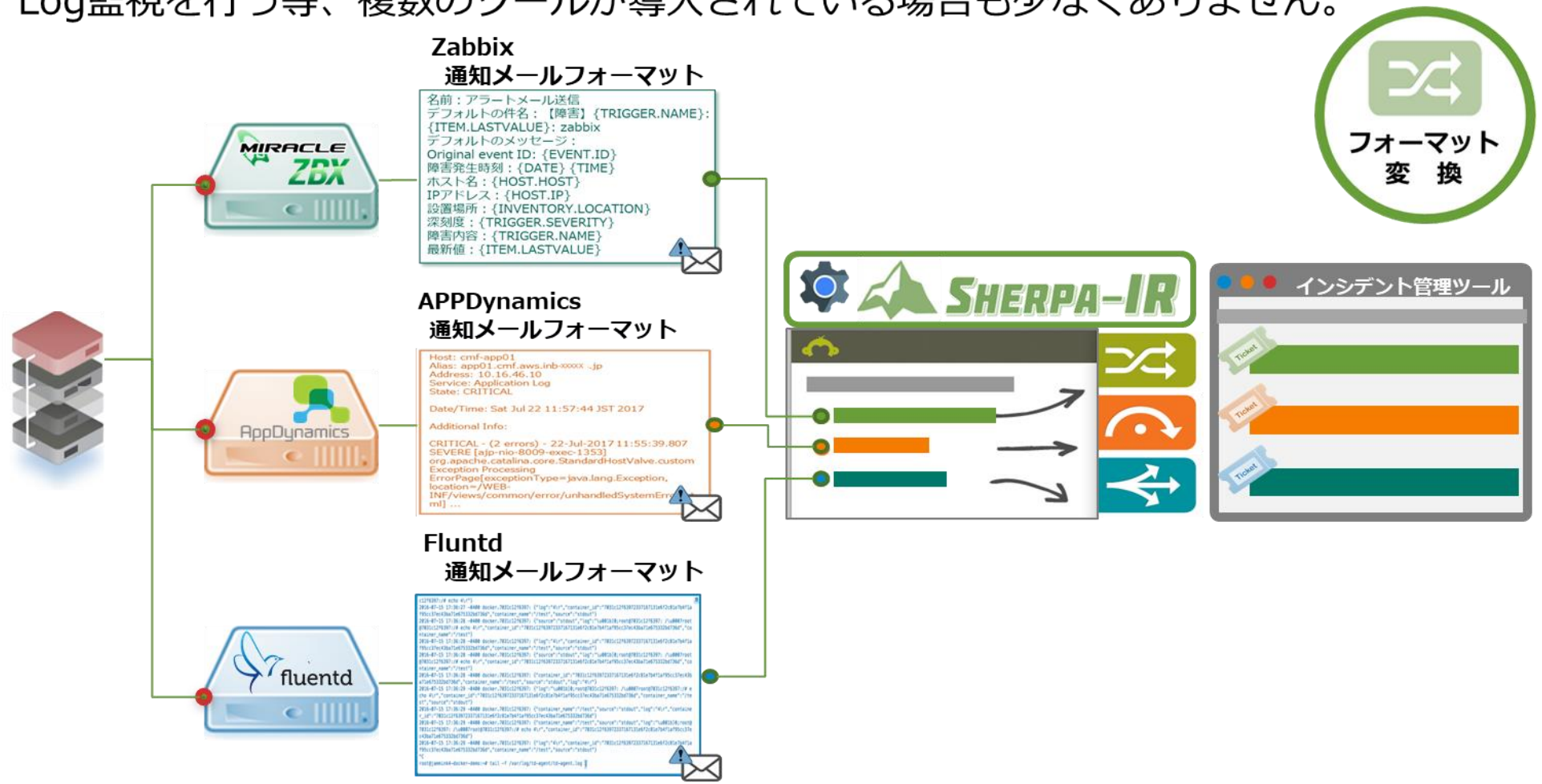
メンテナンス時のアラート制御は、“指定機器”及び“指定時間帯”を非処理機能を利用して行います。指定時間帯のメンテナンス機器からのアラートは無視されます。　メンテナンス時間帯でも、指定されていない機器からのアラートは、通常の制御として処理されます。

## メンテン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：フォーマット変換

実際の運用現場では、統合監視ツール以外に、アプリケーションパフォーマンス監視やLog監視を行う等、複数のツールが導入されている場合も少なくありません。

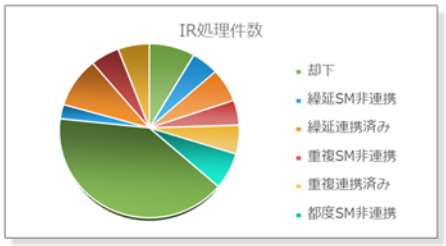


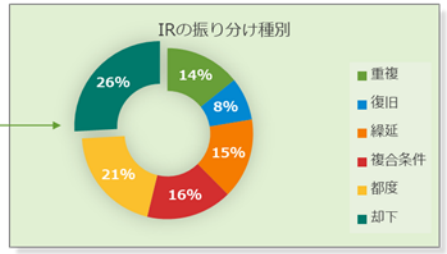

異なる監視ツールのアラートを自動取込み

# SHERPA-IR機能：レポート

cybertrust

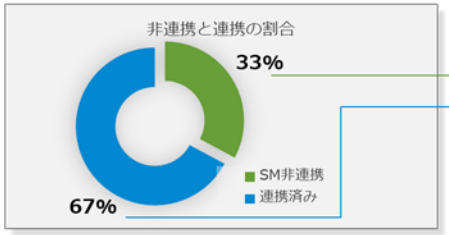
**単月の登録内容**

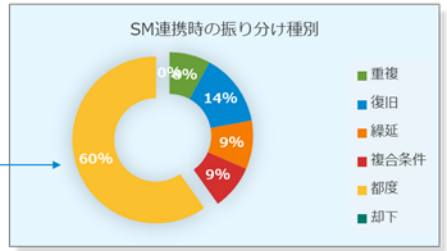


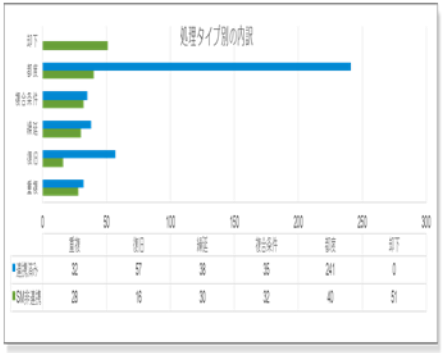

**単月のIR処理数とSMへの連携数**

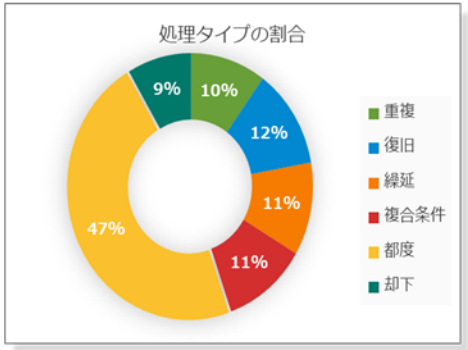
(下のグラフは3割ほどIRで処理)



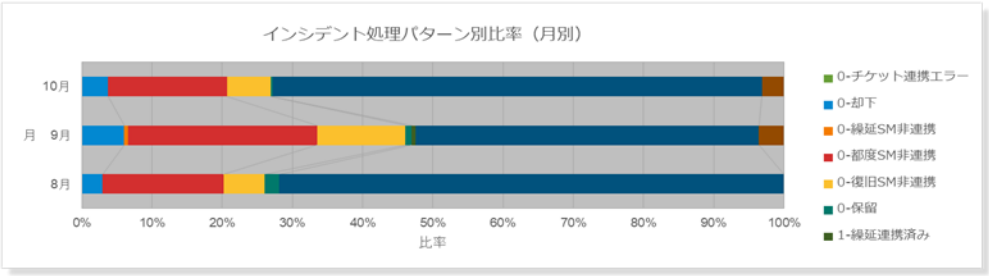

※ 処理タイプ「都度」が良く使われている。
※ 連携すべきインシデント化の精査が必要。




**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 | | |
| 8月 | | 7 | | 41 | 14 | 5 | | 17 | | |
| 9月 | | 62 | 6 | 277 | 130 | 9 | 6 | 503 | | 50 |
| 10月 | 1 | 58 | | 268 | 98 | 4 | | 1100 | 49 | 1578 |
| 総計 | 1 | 127 | 6 | 586 | 242 | 18 | 6 | 1774 | 86 | 2846 |




## SHERPA-IR Reporter

**IR-SMの比較レポート**
IRとSMのチケット番号をキーにチケット単位でIRからSMへ連携状況を出力

**IR集計結果レポート**
IRのデータを処理タイプ別に集計

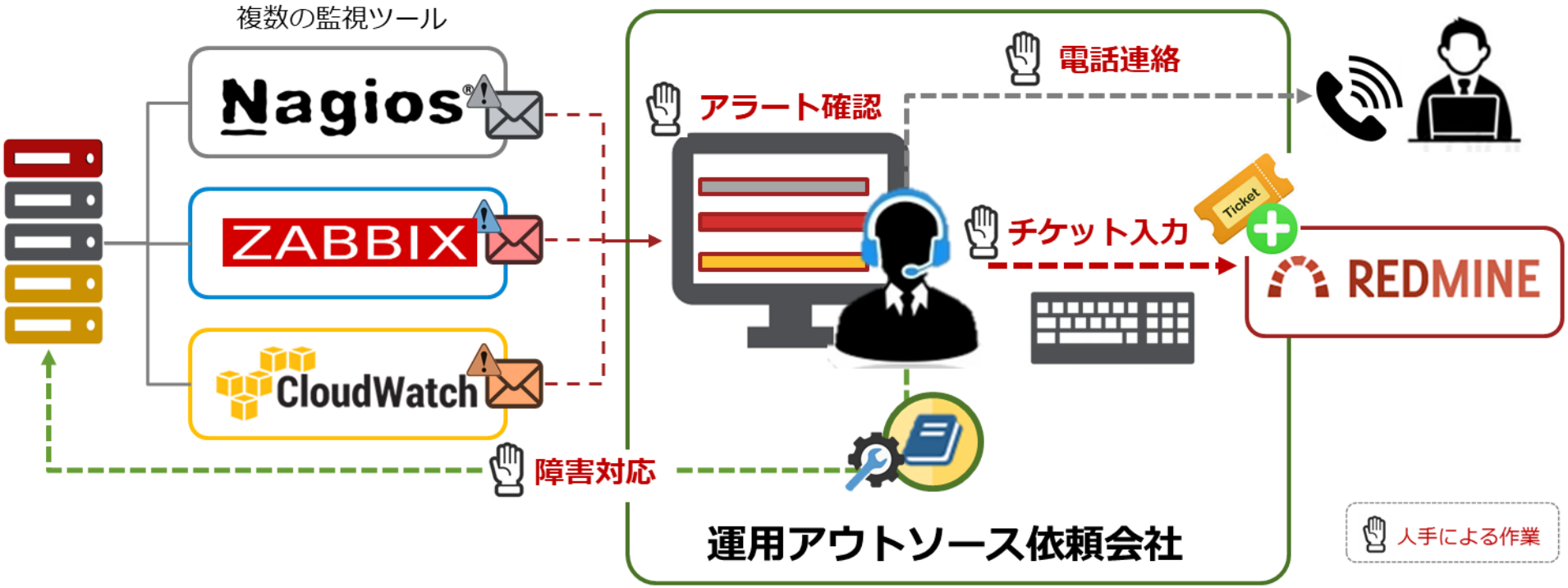
**IR全体レポート**
IR集計結果データを元に、IRからSMへデータ連携の抑制状況をグラフ化

**IR月別推移レポート**
IR集計結果データを元に、IRからSMへデータ連携の抑制状況を月別にグラフ化

処理タイプと処理件数等から運用改善の仮説検証へ展開

# ユーザ事例：概要

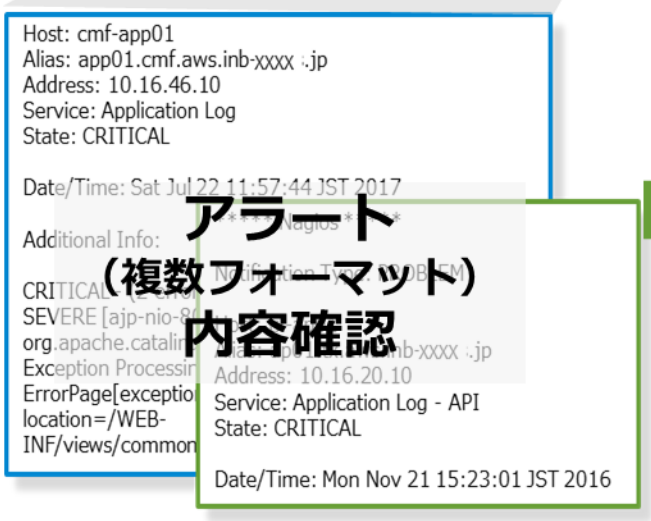
複数の監視ツールが導入され、異なるアラートフォーマットの為、インシデント管理ツールに自動起票が出来きず、オペレータがアラート内容を確認しチケット起票を行った上で、手順書に従い障害対応を実施していた。



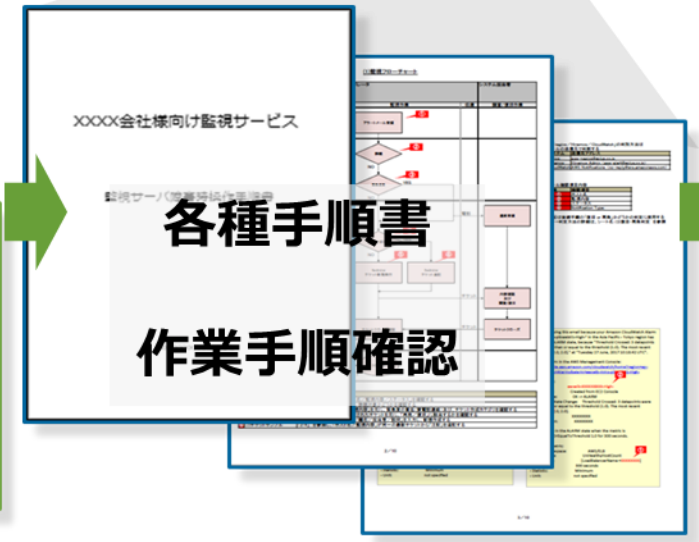

自動化を含めた運用改善を始めたきっかけは・・・

**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：旧作業の流れ


アラート
（複数フォーマット）
内容確認




各種手順書

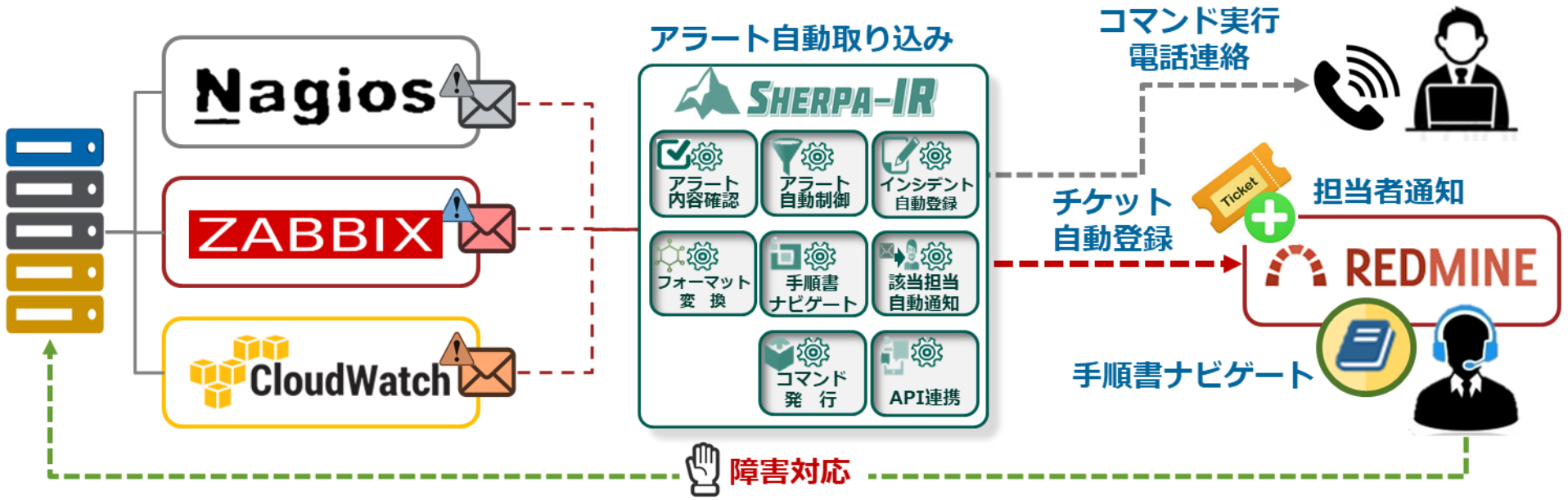
作業手順確認

条件確認表

条件による
処理確認

# 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：導入までの流れ

## 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.html] ...

**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

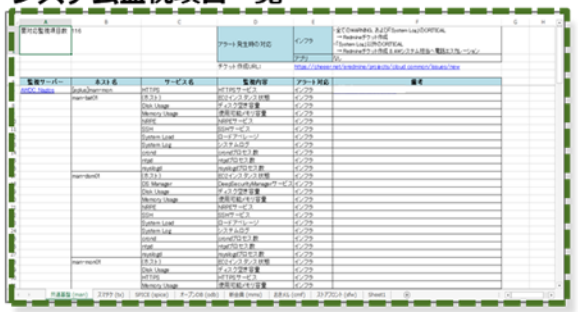
Date/Time: Sat Jul 22 12:13:13 JST 2017

# ユーザ事例：導入までの流れ

cybertrust

## 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

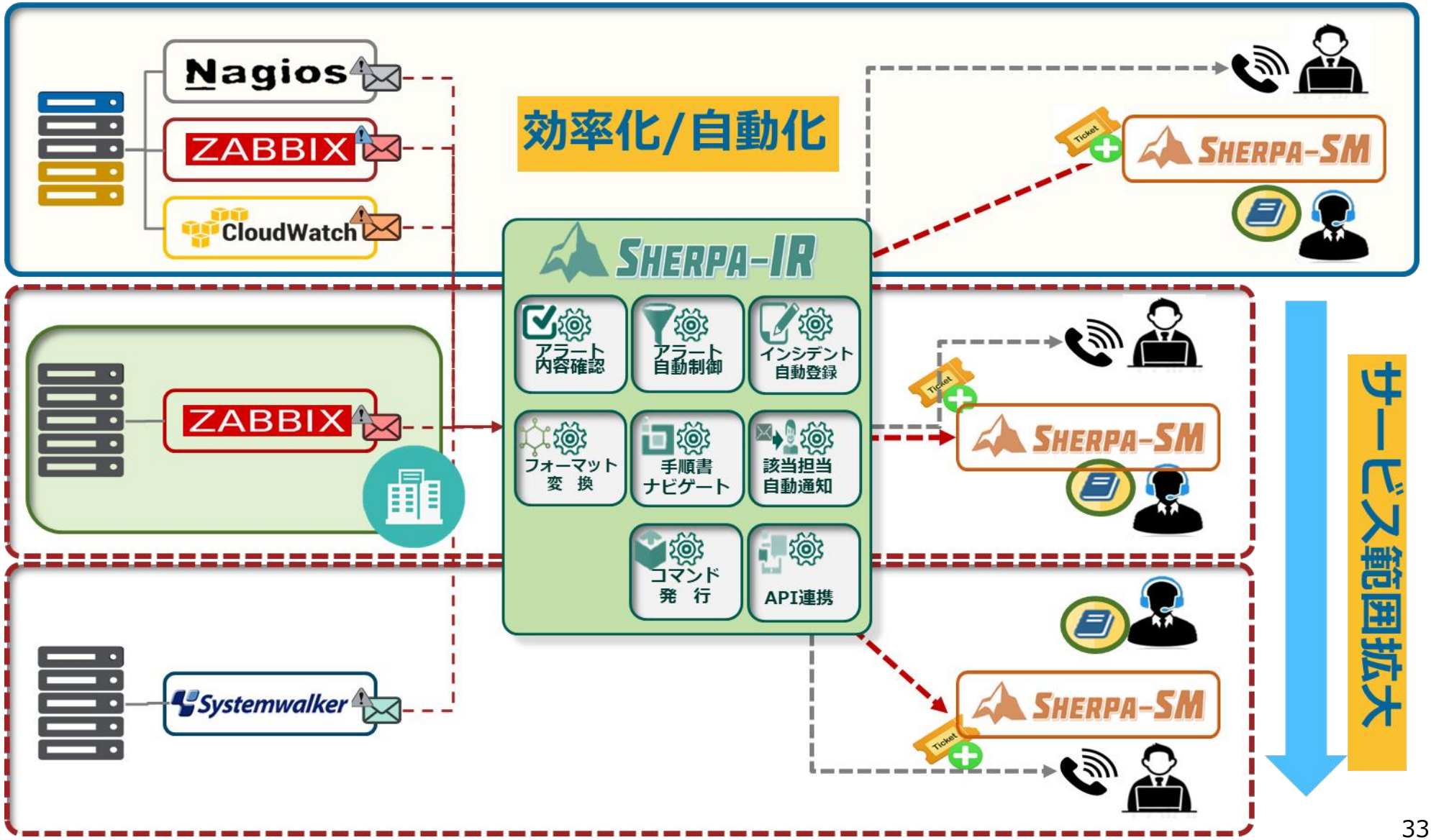
フィルタールール

| 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<br>チケット作成のみ | 都度 |
| tx-dbr01 | MySQL | MySQLサービス | インフラ | | | |
| 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②<br>要電話通知 | 都度<br>(電話通知) |
| 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 | | |
| tx-ap | Application Log - API | アプリケーションログ | アプリ | チケット作成<br>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 | サンプル③<br>要電話通知(重複) | 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。<br>20分でリカバリを検知した場合<br>システム担当者含め電話連絡不要 | | 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 | サンプル④<br>チケット作成のみ(復旧) | 重複+復旧 |

SHERPA-IR

Rule

# ユーザ事例：効率化/自動化の実現



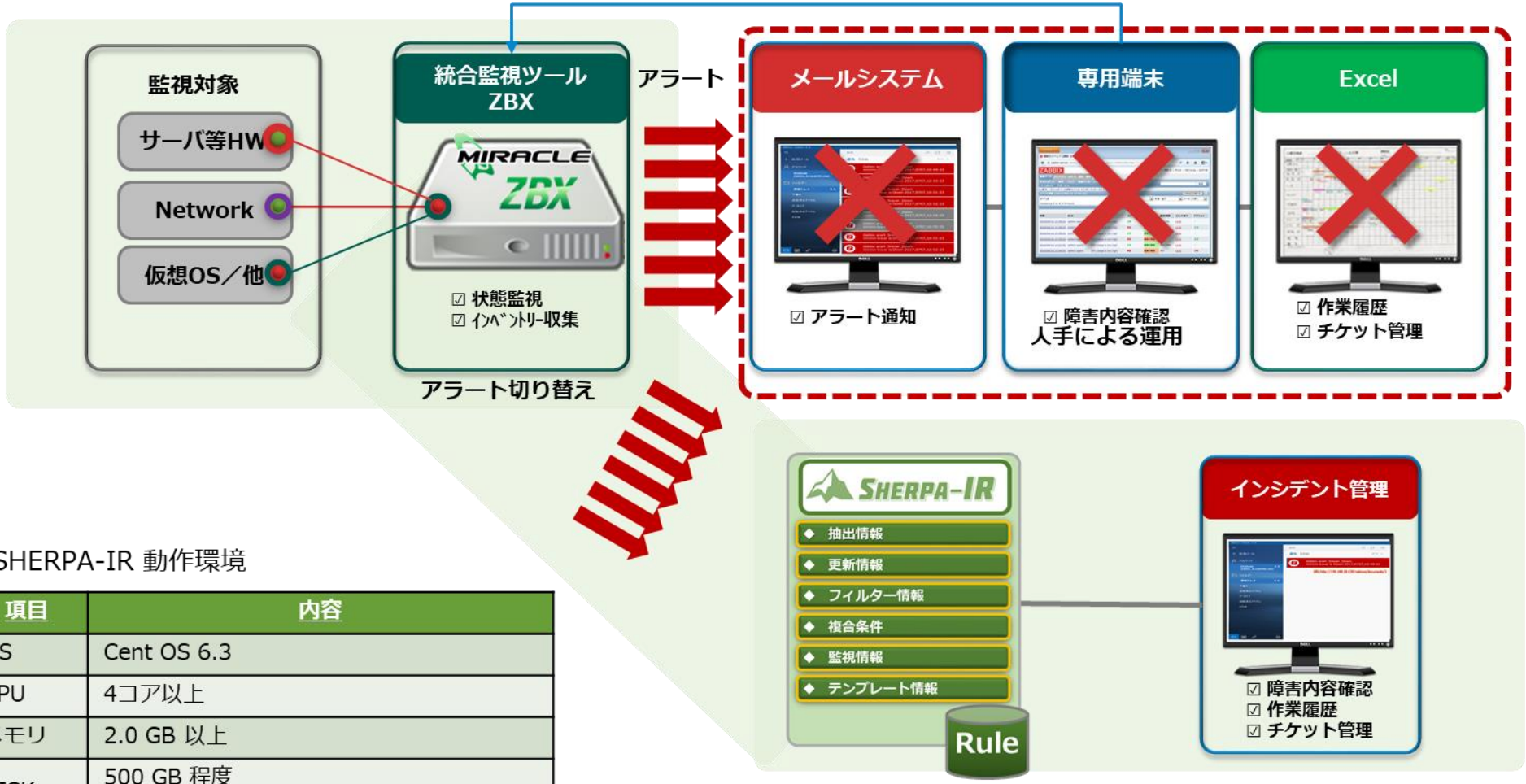





公開

# SHERPA-IRの配置

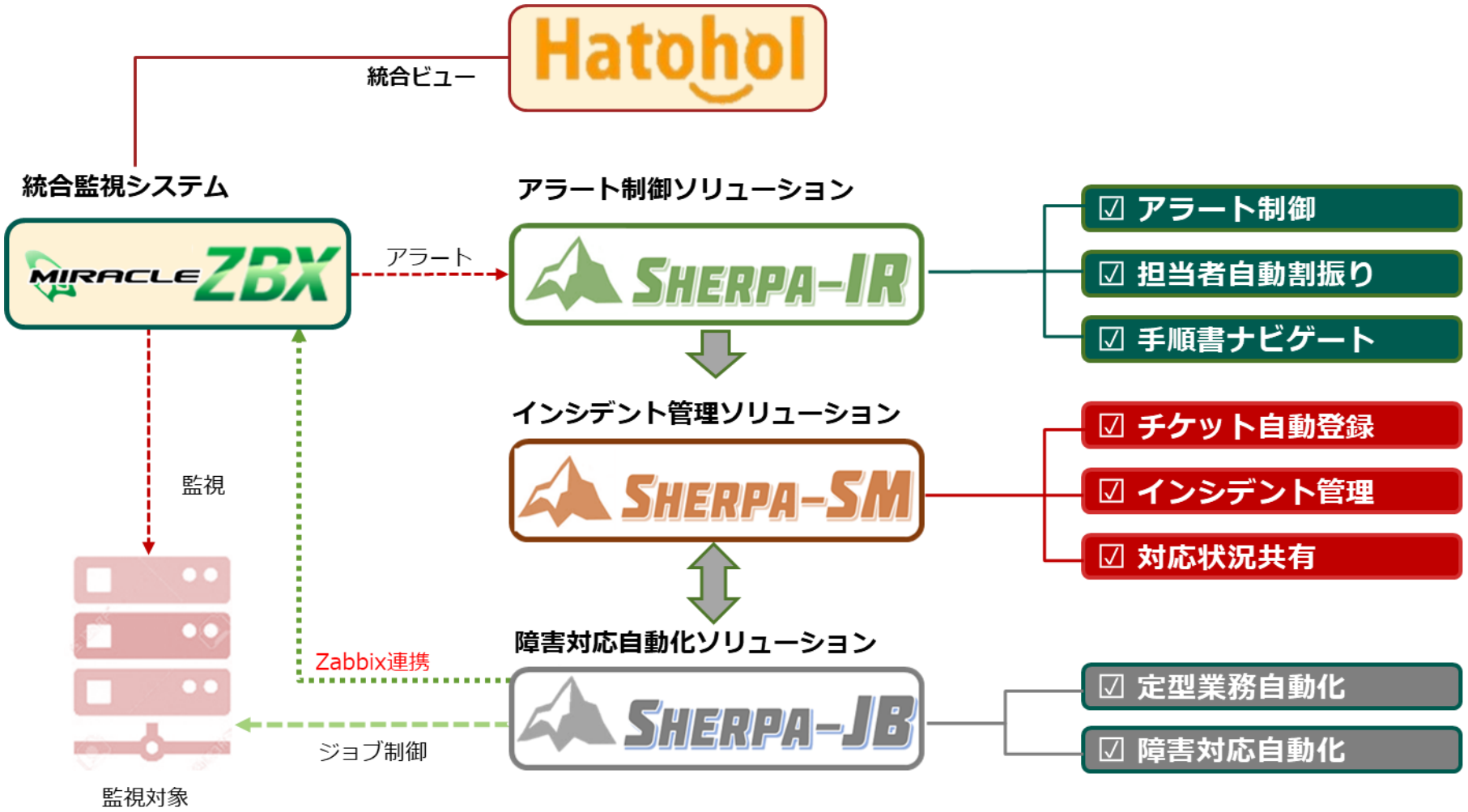
導入は大きなシステムの変更は必要ありません。




■ 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<br>(添付容量等により異なります。) |

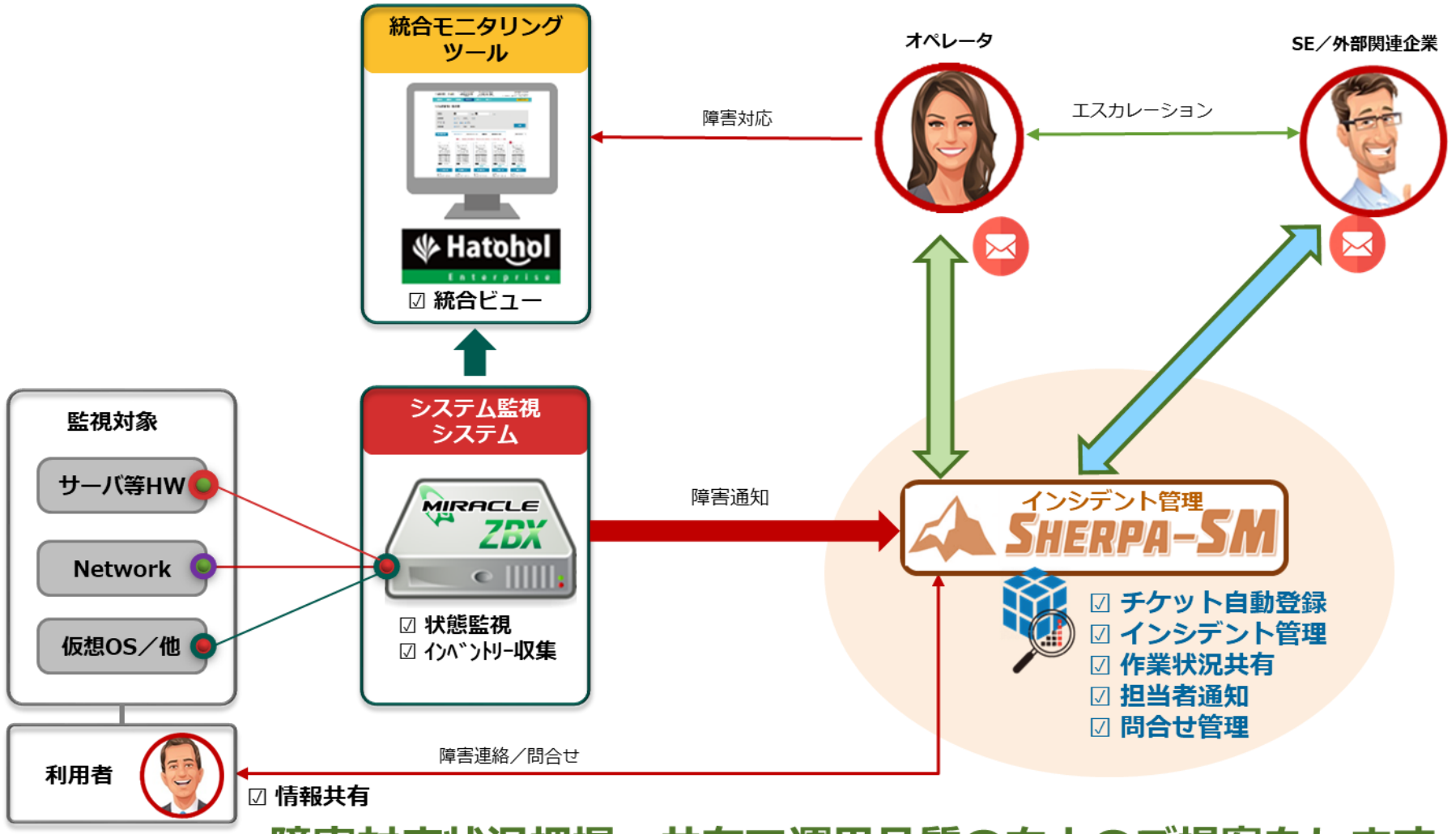
# ご提供ソリューションMAP




**出来るところから運用業務の自動化を始めましょう！**

# 目的別ソリューション

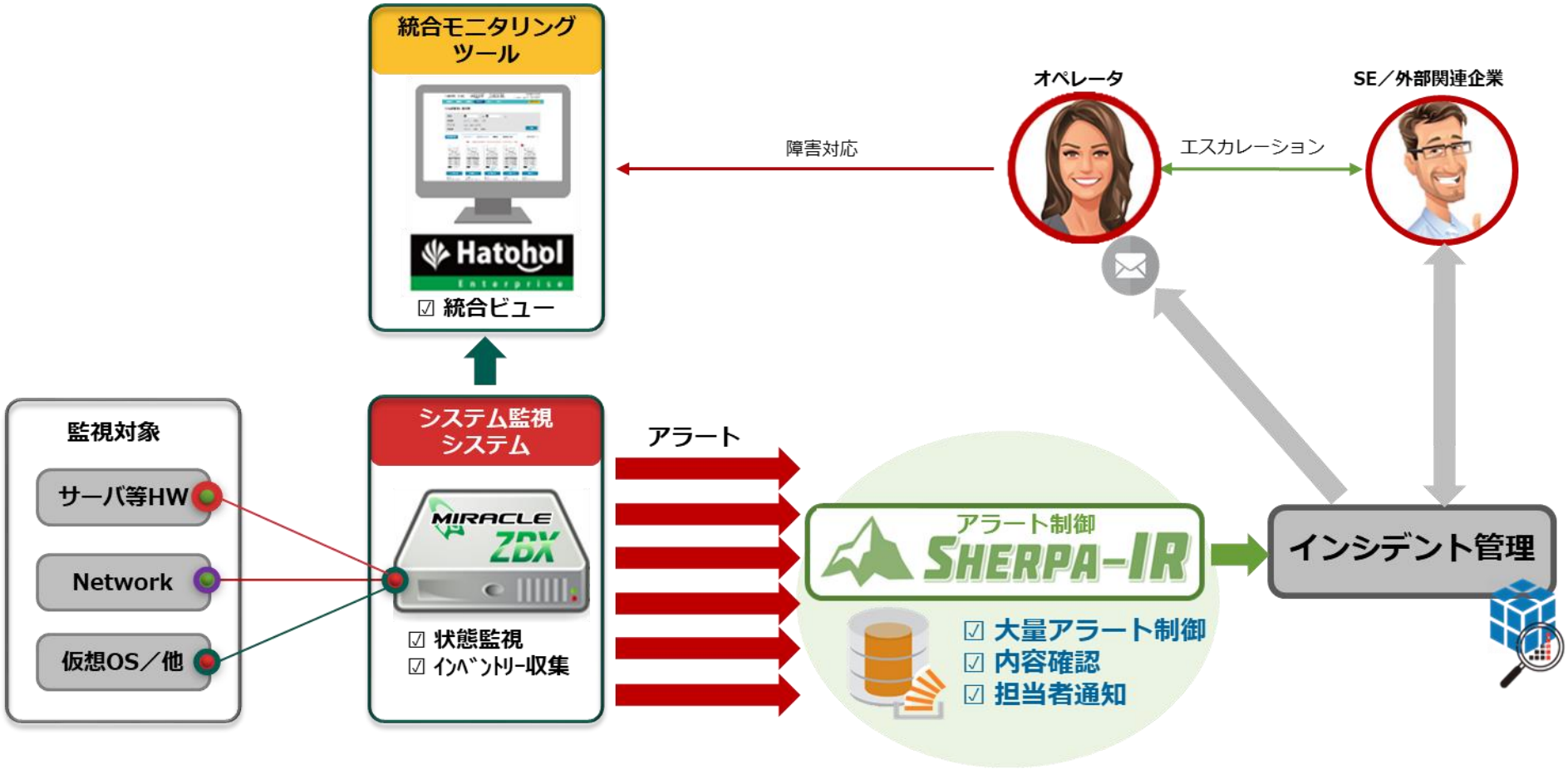
## インシデント管理でお困りのお客様




**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。**

# 目的別ソリューション

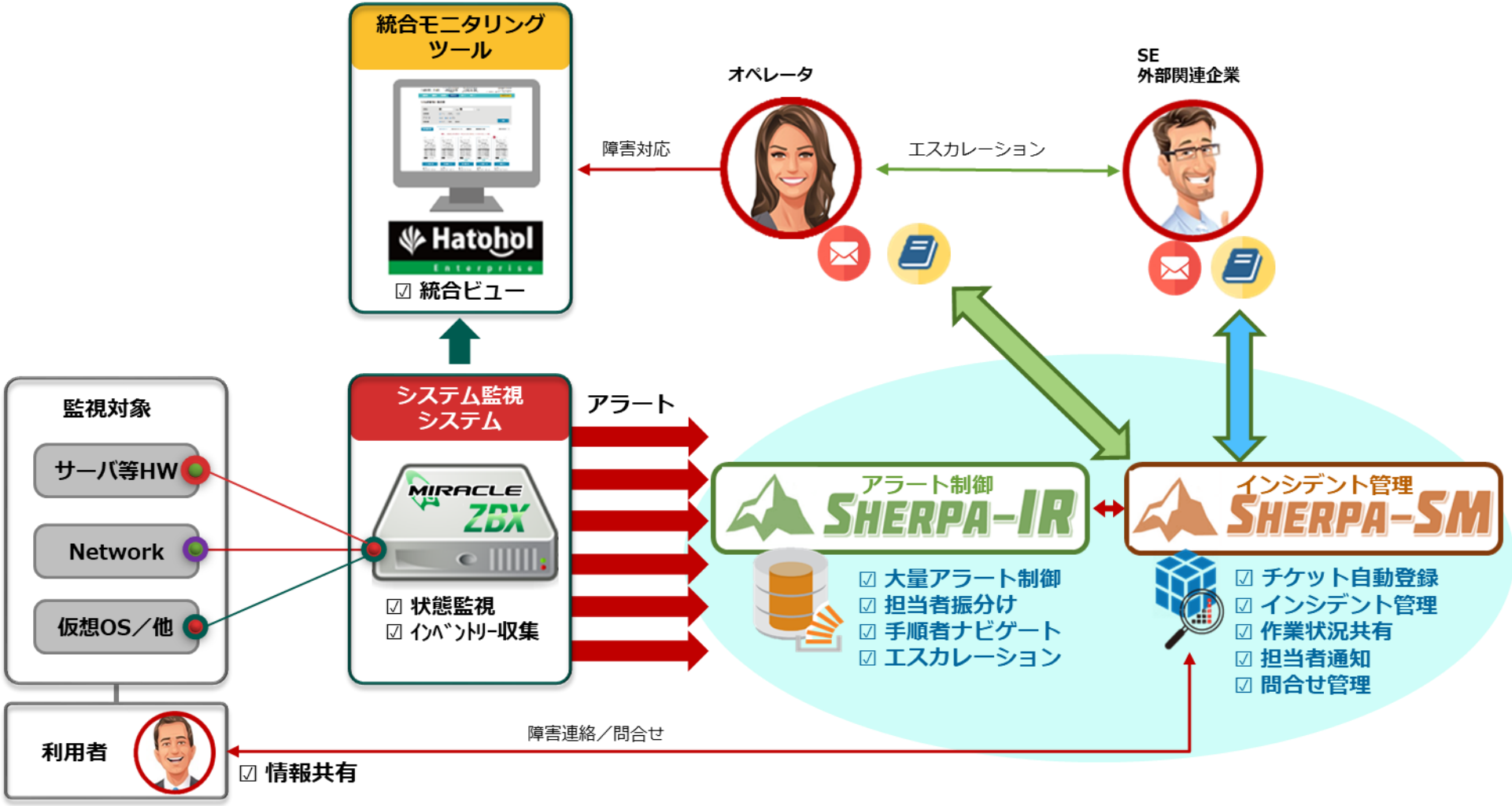
## システム監視システムからの大量アラートでお困りのお客様




**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。**

# 目的別ソリューション

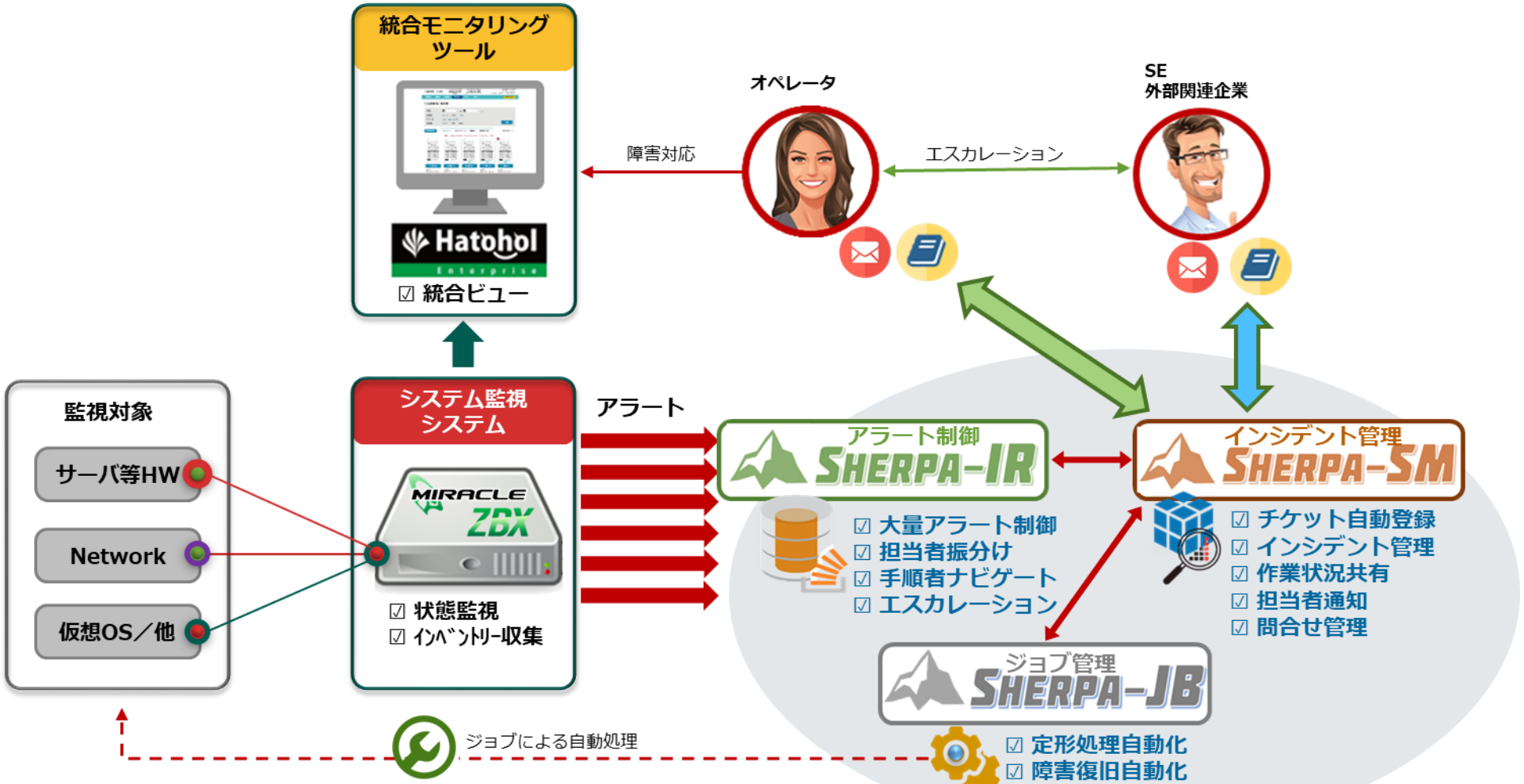
## １次オペレータ作業の効率化でお困りのお客様




**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。**

# 目的別ソリューション

## システム運用の効率化でお困りのお客様




**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