

～商用OSSをどう使い分ける？どう連携する？～

# 商用監視ソリューションとMIRACLE ZBXの 併存および移行によるコストダウンの手法

2016/10/14

ミラクル・リナックス株式会社  
マーケティング本部 青山 雄一



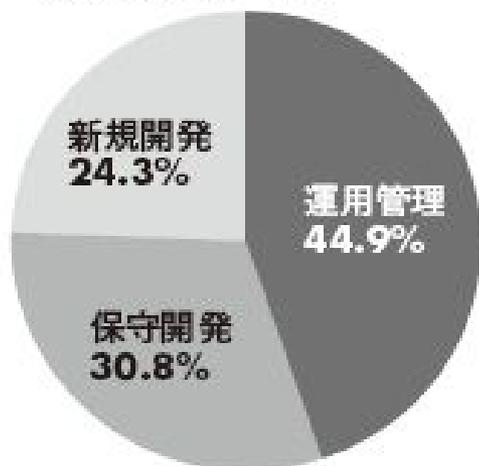
- ・ 商用監視ツールとの併用方法
- ・ 商用監視ツールとの併用検証結果
- ・ その他のツールとの連携
  - ジョブコントローラー
  - 監視コンソール

# 運用管理の課題

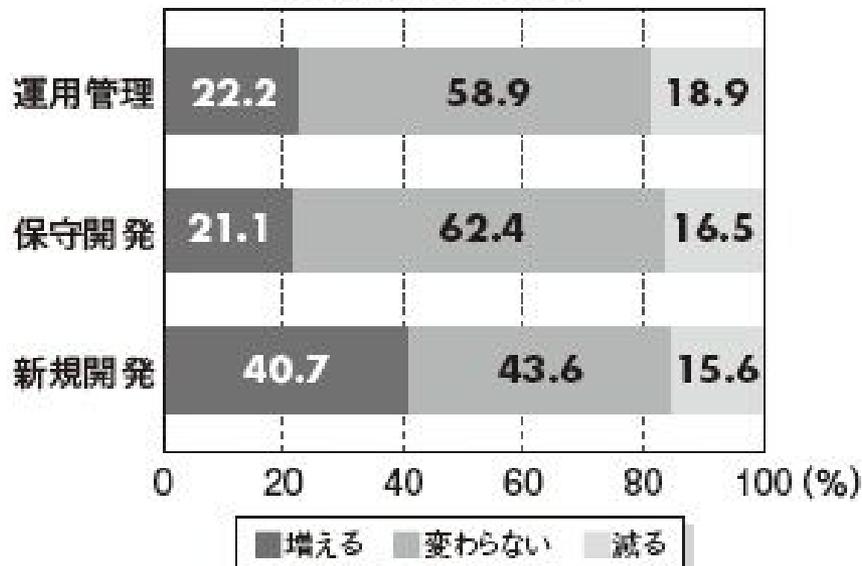


- 監視対象が増えてコストが増えた
- 仮想化やクラウドの利用が増えてマシンの増減が激しい
- ライセンス体系が複雑で理解できない

◆2012年度のIT関連コスト内訳  
(有効回答数=491)



◆今後の見通し  
(有効回答数=582)



日経BPシステム運用ナレッジ  
「企業情報システムの運用管理に関する実態調査 2013」  
<http://corporate.nikkeibp.co.jp/information/newsrelease/newsrelease20130610.shtml>

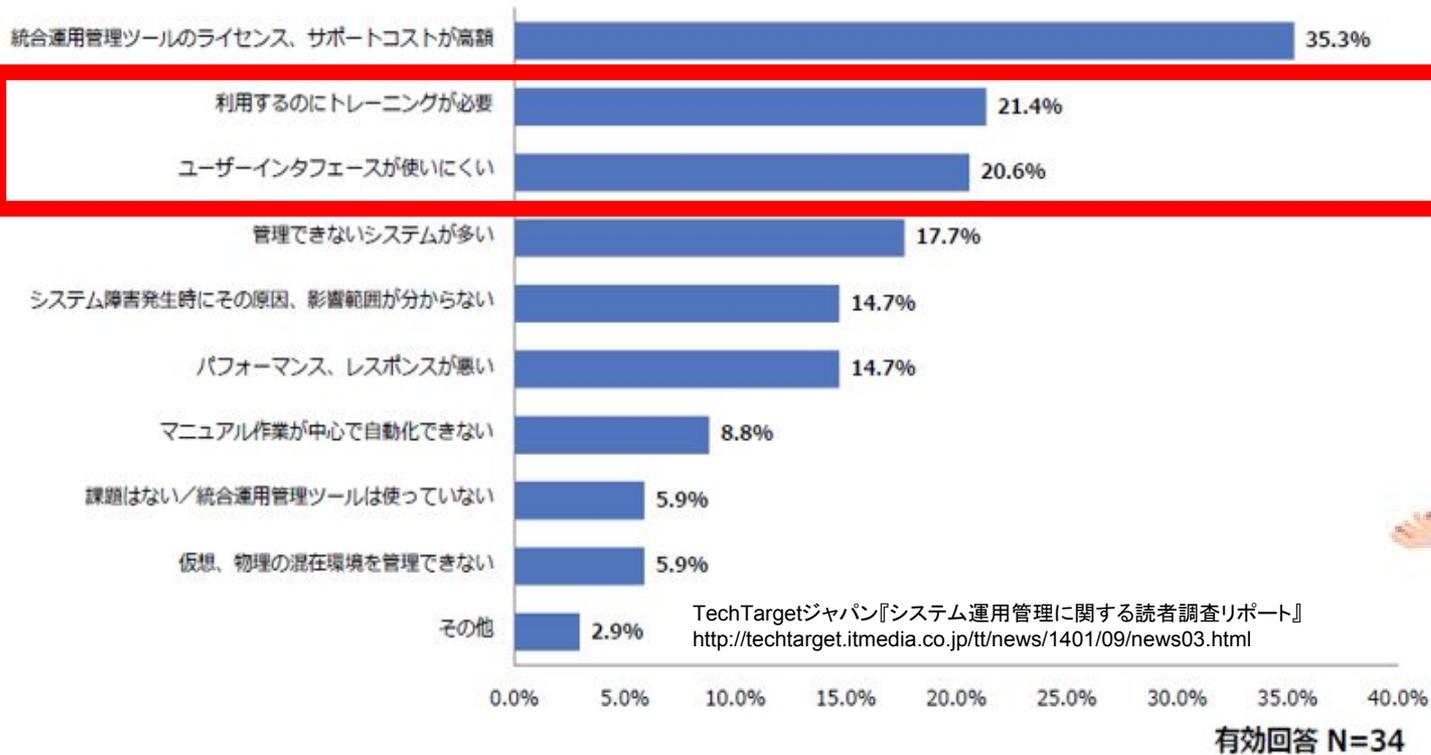
Do the Next, Open your Window

MIRACLE

# OSS導入時の不安



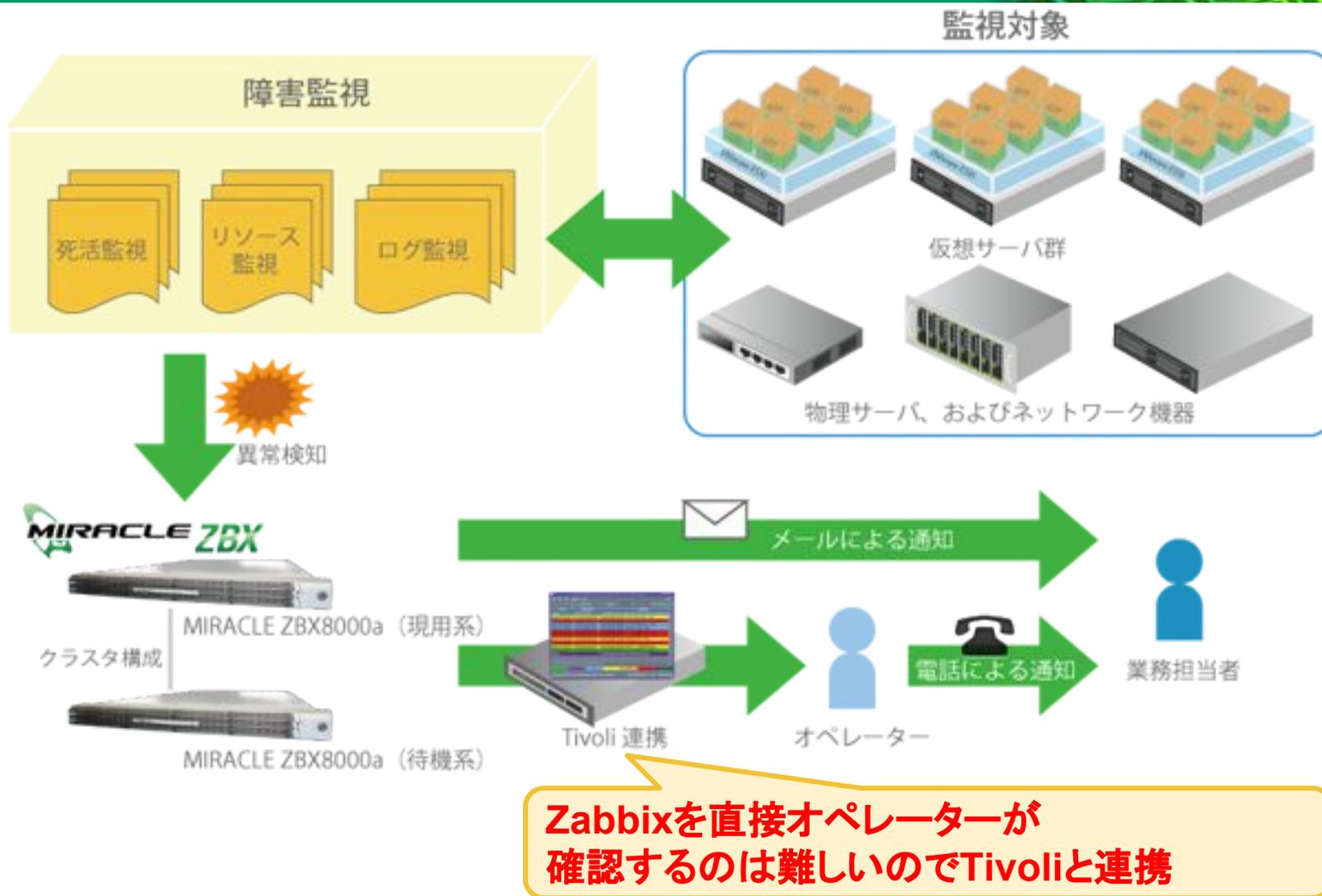
- 運用はあまり変えたくないんだけど・・・
- いきなり全部入れ替えるのは不安
- OSSを使って何か問題が起きたらどうすれば良い？



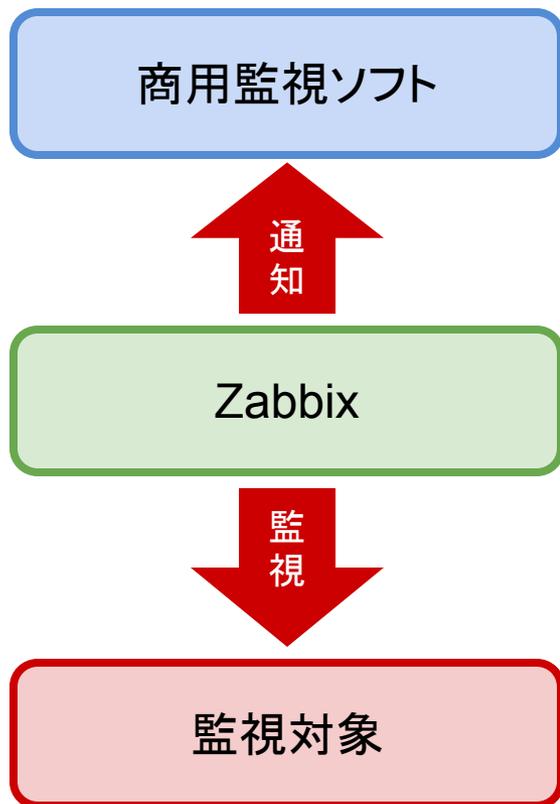
Do the Next, Open your Window

MIRACLE

# インテージ様での使い方



# Zabbixと商用監視製品の併用



- ・ 大量の監視対象はZabbixで監視
- ・ 障害検知時は既存のツールに通知
- ・ 普段は既存の監視コンソールを監視



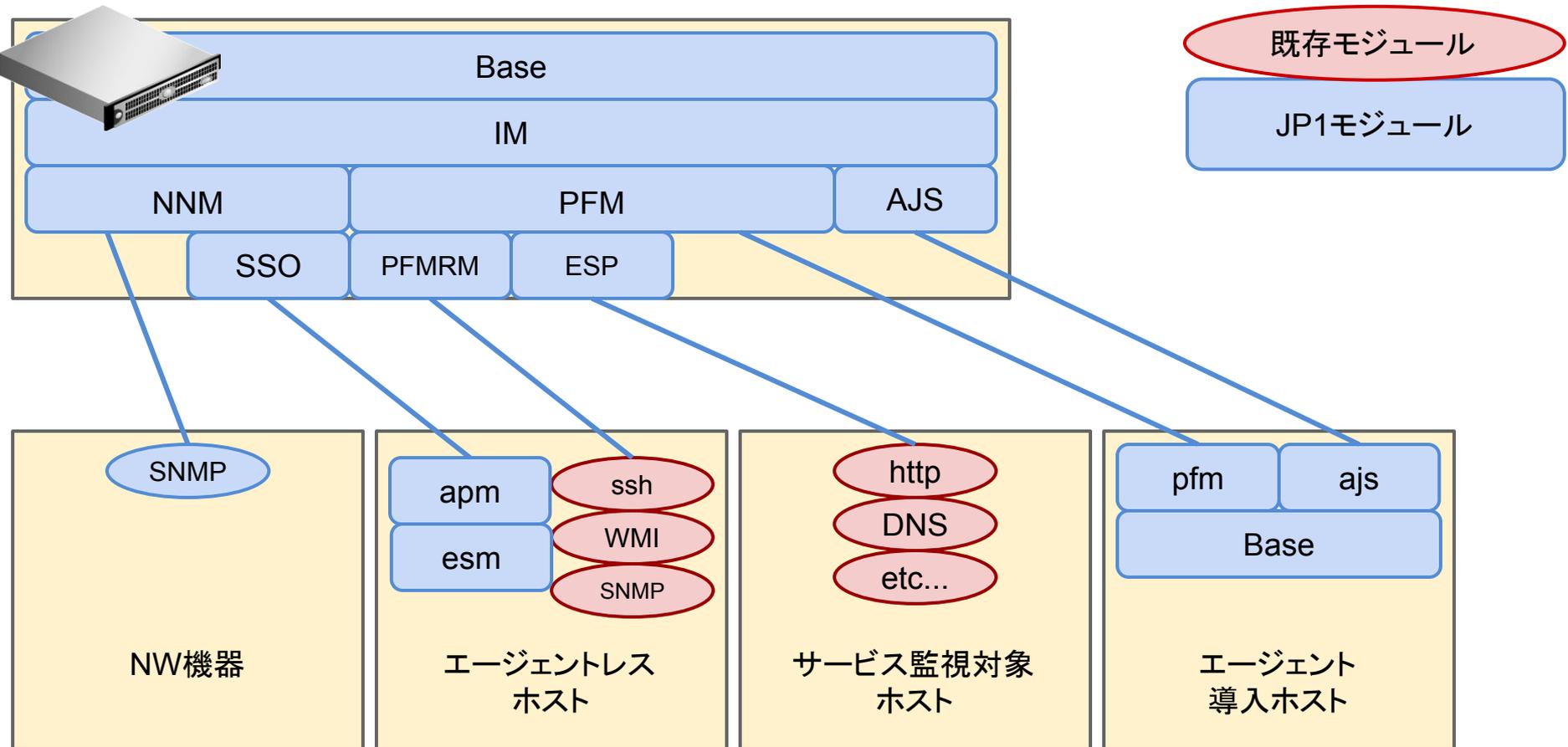
## メリット

- エージェント費用が削減できる
- 監視台数が変動してもコストは一定
- 監視画面の変更がない
- 段階的にOSSへの移行ができる



# 例えば、JP1の構成は・・・

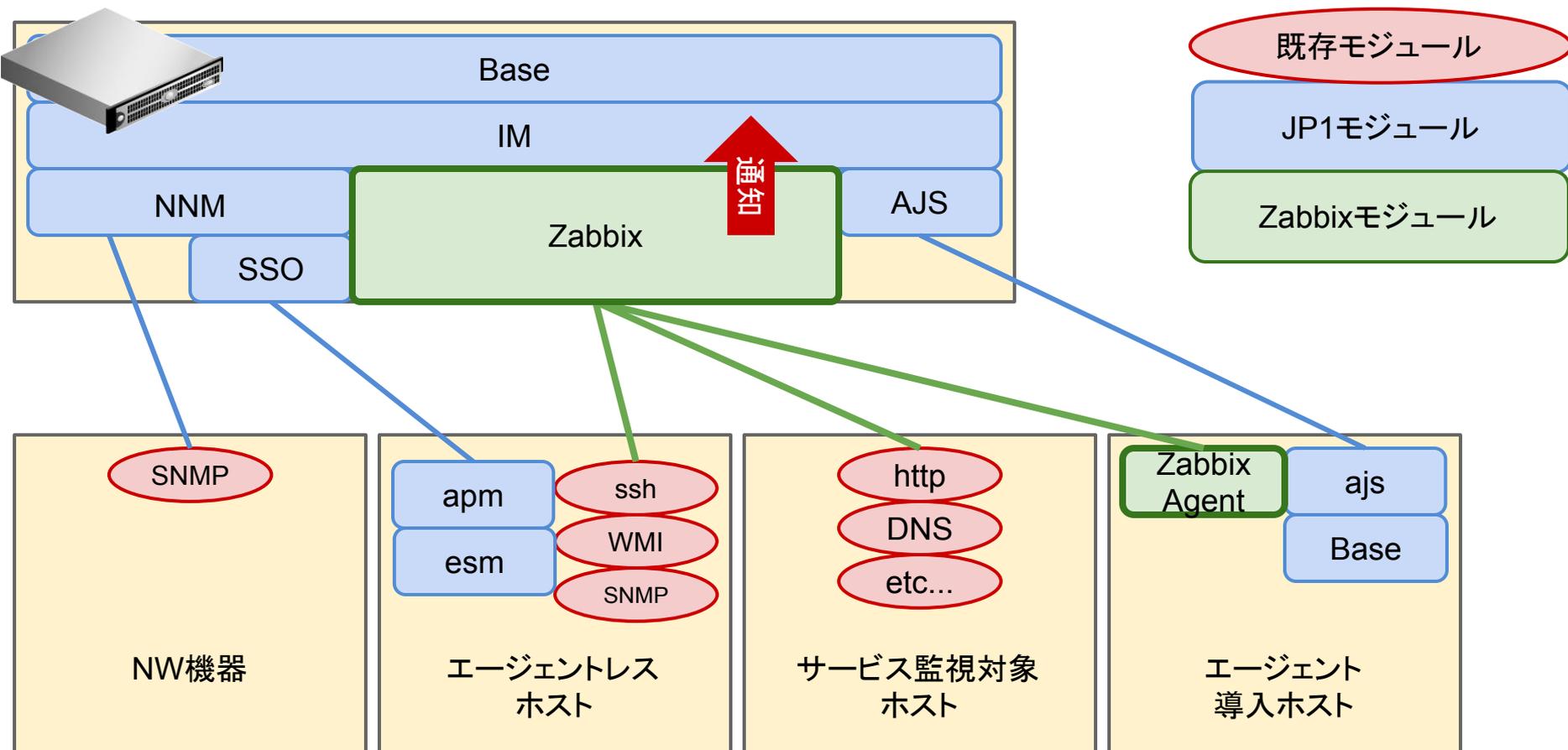
- ・多くの商用監視製品は監視機能x監視対象台数で課金
- ・モジュールによってライセンスポリシーが異なる場合が多い



# JP1とZabbixの共存例



- ・ Zabbixを監視エンジンとして利用し、イベントをJP1に通知
- ・ このパターンで多くのエージェント費用を削減できる
- ・ JP1はIM、NNM、AJSを利用



# Zabbixの主な監視機能



分類	監視項目	エージェント
Zabbixエージェント	<ul style="list-style-type: none"><li>・リソース監視 (CPU、メモリ、ディスク、ネットワークなど)</li><li>・ファイル監視 (更新、サイズ、存在)</li><li>・プロセス監視</li><li>・ログ監視</li><li>・Windowsサービス監視</li><li>・Windowsイベントログ監視</li><li>・Windowsパフォーマンスカウンター監視</li><li>・スクリプト実行 (エージェントサイド) など</li></ul>	必要
SNMP Polling / Trap	V1、V2c、V3 プロトコルに準じる	不要
ICMP、TCP	接続監視、応答速度	不要
IPMI、JMX、SQL	プロトコルに準じる	不要
HTTP	ダウンロードスピード 文字列監視 HTTPステータス 応答時間	不要
その他	リモートシェル (SSH / Telnet) スクリプト実行 (サーバサイド) トラップ (任意のプログラムに組み込み可能な任意のデータ送信)	不要

# NNM (Network Node Manager) はどうする？



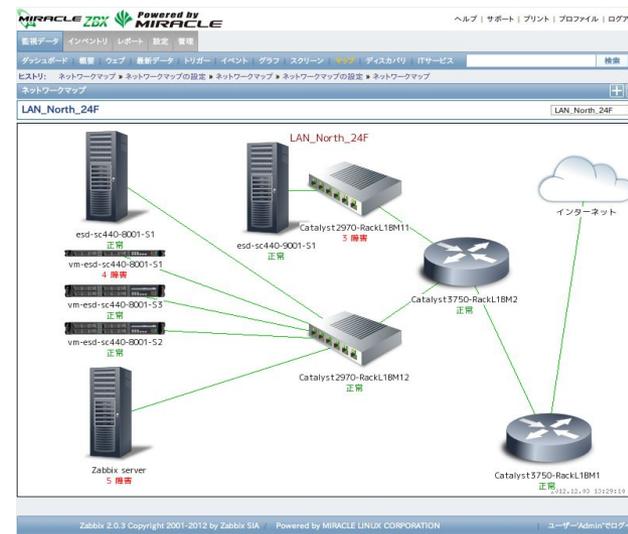
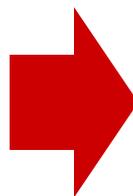
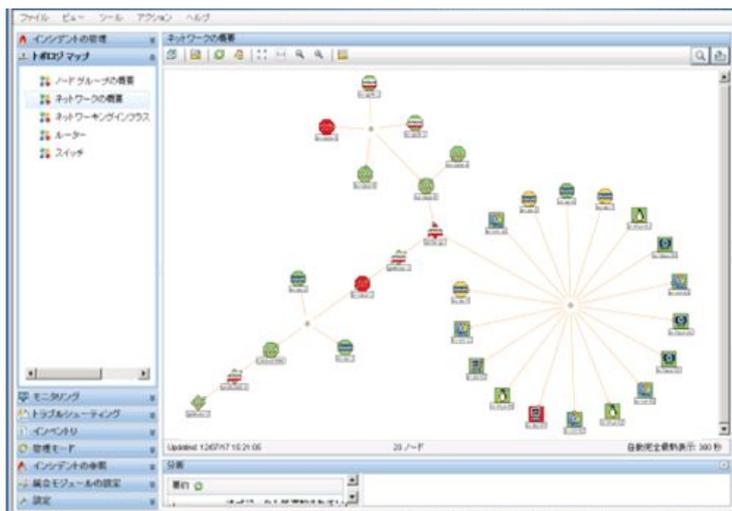
## NNMのZabbixへの置き換えは使い方によって検討が必要

Zabbixでは以下が可能

- ・ SNMP監視／SNMPトラップ監視は可能
- ・ 手動でのマップ作成

ただし、以下はできない または 苦手

- ・ インテリジェントな自動マップ作成
- ・ 高度な解析機能
- ・ 大量ノードを管理してのマップ表示



Do the Next, Open your Window

MIRACLE

# 商用監視ツール連携の仕組み

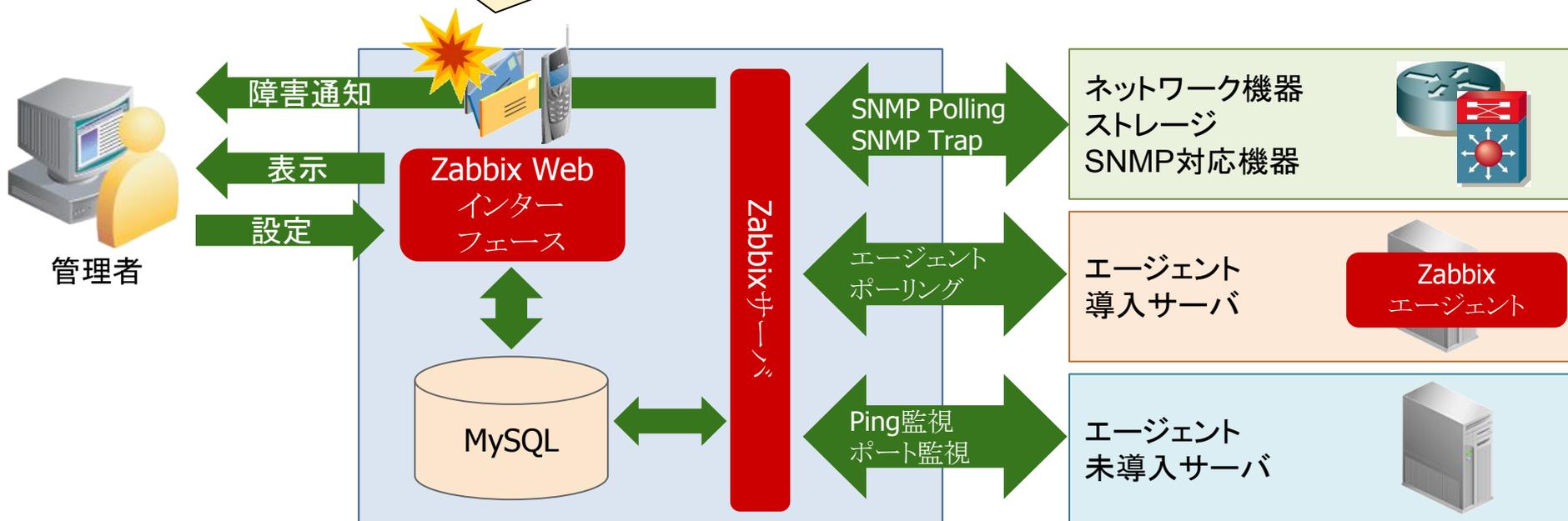


## Zabbixでの通知方法

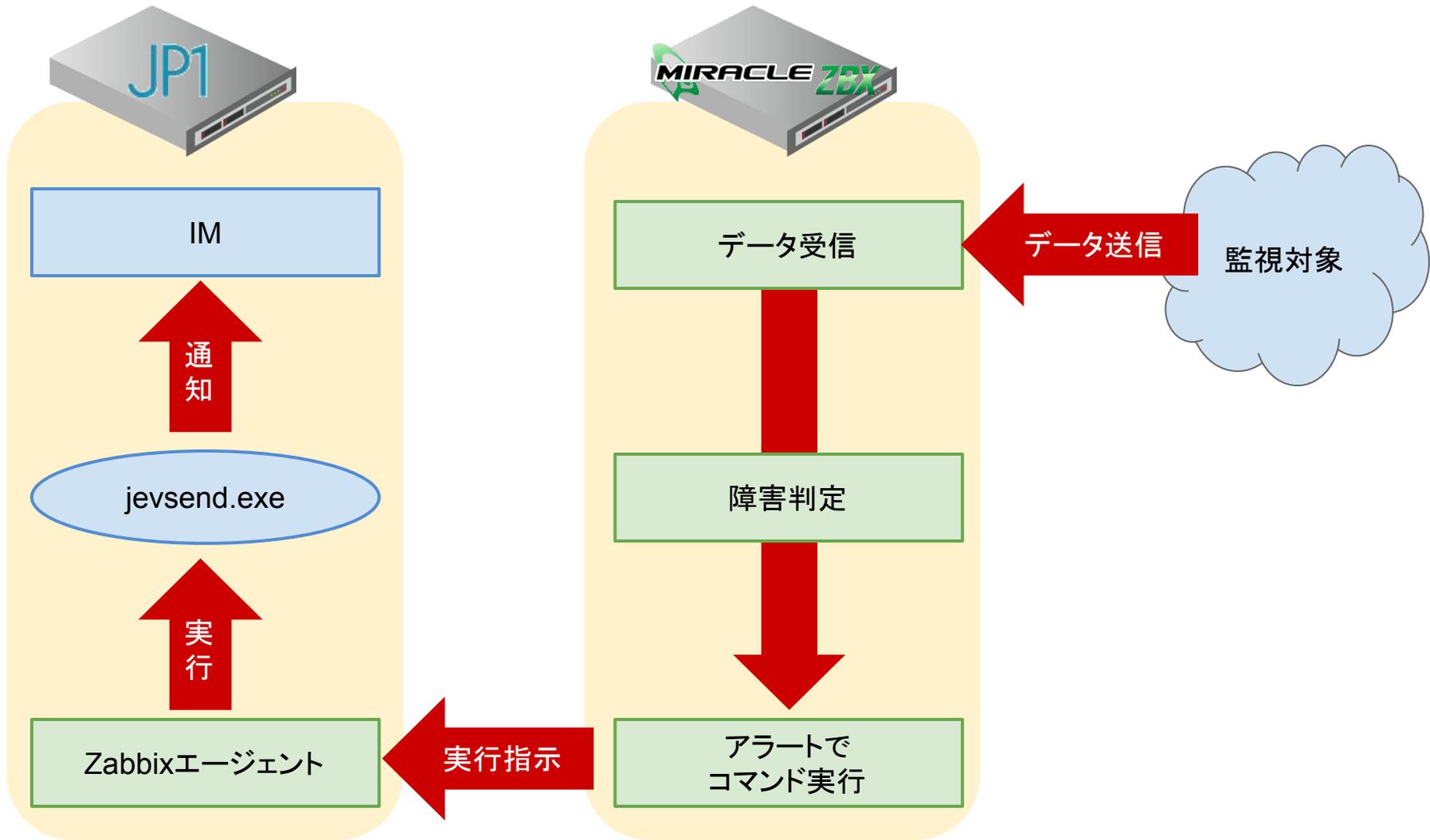
- メール
- コマンド実行(サーバーサイド)
- コマンド実行(エージェントサイド)



この仕組みを利用し、  
商用製品付属の通知コマンドを実行



# JP1とZabbix連携の方法





# JP1-Zabbixの重要度の紐付け

- ・Zabbixで定義された障害の重要度にJP1での障害の重要度を紐付けてIM上で障害の重要度が表示できるよう設定を行う。
- ・以下に関しては紐付けを行わない
  - Zabbixの『不明』  
監視が正常に行えていない状態のため
  - JP1の『緊急』  
ユーザ割り当て用のため

C:\Program Files (x86)\Hitachi\JP1Cons\confchsev\jcochsev.conf

```
DESC_VERSION=バージョン情報
def 定義名1
  cnd
    イベント条件
  end-cnd
  sev 重大度
end-def
```

例)

```
DESC_VERSION=1
def 重大度変更1
  cnd
    B.MESSAGE SUBSTR【ZBX_Severity:Not_Classified】
  end-cnd
  sev Notice
end-def
def 重大度変更2
  cnd
    B.MESSAGE SUBSTR【ZBX_Severity:Information】
  end-cnd
  sev Information
end-def
```

jevsend.confについては以下を参照

<http://itdoc.hitachi.co.jp/manuals/3020/30203K0660/BASE0105.HTM>

JP1 イベントSeverity	Zabbix障害重要度
	不明(Unknown)
通知(Notice)	未分類(Not_Classified)
情報(Information)	情報(Information)
警告(Warning)	警告(Warning)
エラー(Error)	軽度の障害(Average)
致命的(Critical)	重度の障害(High)
警戒(Alert)	致命的な障害(Disaster)
緊急(Emergency)	

# プロセスダウンの通知例



JP1 Central Console(jp1admin@jp1-server)

ファイル 編集 表示 オプション ヘルプ

Integrated Management

表示範囲: 12/03 11:12:35 - 12/03 11:13:02

表示フィルター編集... フィルター名: 既存のフィルター条件  表示フィルター

アク...	重大度	登録時刻	発生元ホスト名	イベントID	メッセージ	オブジェクトタイプ	オブ
<input checked="" type="checkbox"/>	警戒	12/03 11:12:35	jp1-server	00001000	[ZBX_Severity:Disaster] 11:12:22:httpdプロセスがダウンしました:PROBLEM		
<input checked="" type="checkbox"/>	警戒	12/03 11:13:02	jp1-server	00001000	[ZBX_Severity:Disaster] 11:12:49:httpdプロセスがダウンしました:OK		

イベント詳細 - イベントコンソール

編集 前イベント 次イベント

属性名	属性値
イベントDB内通し番号	1320
イベントID	00001000
発行元プロセスID	1972
登録時刻	12/03 11:20:35
到着時間	12/03 11:20:35
発行元ユーザーID	-1
発行元グループID	-1
発行元ユーザー名	SYSTEM
発行元グループ名	
発生元ホスト名	jp1-server
発行元イベントサーバ名	jp1-server
発行元IPアドレス	192.168.10.10

メッセージ

[ZBX\_Severity:Disaster] 11:20:22:httpdプロセスがダウンしました:PROBLEM

メモ

モニター起動 アクション結果 閉じる ヘルプ

イベント詳細 - イベントコンソール

編集 前イベント 次イベント

属性名	属性値
イベントDB内通し番号	1322
イベントID	00001000
発行元プロセスID	2636
登録時刻	12/03 11:21:47
到着時間	12/03 11:21:47
発行元ユーザーID	-1
発行元グループID	-1
発行元ユーザー名	SYSTEM
発行元グループ名	
発生元ホスト名	jp1-server
発行元イベントサーバ名	jp1-server
発行元IPアドレス	192.168.10.10

メッセージ

[ZBX\_Severity:Disaster] 11:21:34:httpdプロセスがダウンしました:OK

メモ

モニター起動 アクション結果 閉じる ヘルプ

# エラーログ検知、SNMPトラップ受信の通知例



JP1 Central Console (jp1admin@jp1-server)

ファイル 編集 表示 オプション ヘルプ

Integrated Management

表示範囲: 11/29 16:30:39 - 12/04 11:54:26

表示フィルター編集... フィルター名: 既存のフィルター条件  表示フィルター

アク...	重大度	登録時刻	発生元ホスト名	イベントID	メッセージ	オブジェクトタイプ
<input checked="" type="checkbox"/>	致命的	12/04 08:52:48	jp1-server	00001000	[ZBX_Severity:High] 08:52:32:snmptrapを受信しました:OK	
<input checked="" type="checkbox"/>	警戒	12/04 08:53:09	jp1-server	00001000	[ZBX_Severity:Disaster] 08:52:53:snmptrapを受信しました:OK	
<input checked="" type="checkbox"/>	通知	12/04 08:54:24	jp1-server	00001000	[ZBX_Severity:Not classified] 08:54:08:snmptrapを受信しました:PROBLEM	
<input checked="" type="checkbox"/>	情報	12/04 09:52:03	jp1-server	00001000	[ZBX_Severity:Information] 09:51:48:snmptrapを受信しました:OK	
<input checked="" type="checkbox"/>	警告	12/04 09:52:24	jp1-server	00001000	[ZBX_Severity:Warning] 09:52:09:snmptrapを受信しました:OK	
<input checked="" type="checkbox"/>	エラー	12/04 09:52:45	jp1-server	00001000	[ZBX_Severity:Average] 09:52:30:snmptrapを受信しました:OK	
<input checked="" type="checkbox"/>	致命的	12/04 09:52:48	jp1-server	00001000	[ZBX_Severity:High] 09:52:33:snmptrapを受信しました:OK	

イベント詳細 - イベントコンソール

編集 前イベント 次イベント

属性名	属性値
イベントDB内通し番号	1331
イベントID	00001000
発行元プロセスID	2124
登録時刻	12/03 11:52:32
到着時間	12/03 11:52:32
発行元ユーザーID	-1
発行元グループID	-1
発行元ユーザー名	SYSTEM
発行元グループ名	
発生元ホスト名	jp1-server
発行元イベントサーバ名	jp1-server
発行元IPアドレス	192.168.10.10

メッセージ: [ZBX\_Severity:High] 11:52:19:snmptrapを受信しました:PROBLEM

メモ

モニター起動 アクション結果 閉じる ヘルプ

イベント詳細 - イベントコンソール

編集 前イベント 次イベント

属性名	属性値
イベントDB内通し番号	1188
イベントID	00001000
発行元プロセスID	4620
登録時刻	12/03 01:07:33
到着時間	12/03 01:07:33
発行元ユーザーID	-1
発行元グループID	-1
発行元ユーザー名	SYSTEM
発行元グループ名	
発生元ホスト名	jp1-server
発行元イベントサーバ名	jp1-server
発行元IPアドレス	192.168.10.10

メッセージ: [ZBX\_Severity:Disaster] 01:07:22:エラーログを検知しました:PROBLEM

メモ

モニター起動 アクション結果 閉じる ヘルプ



# メッセージの整形方法

マクロを使ってjevsendに引数を渡す

※マクロとは: Zabbixで利用できる変数

マクロ名称	展開内容
{EVENT.DATE}	障害発生日付
{EVENT.TIME}	障害発生時刻
{HOSTNAME}	障害発生ホスト
{IPADDRESS}	障害発生ホストのIPアドレス
{ITEM.VALUE}	障害の引き金となった値
{ITEM.NAME}	障害ソースアイテム名
{TRIGGER.COMMENT}	トリガーのコメント
{TRIGGER.SEVERITY}	障害重要度
{TRIGGER.STATUS}	ステータス(OK、PROBLEM)

コマンド指定例

```
"jevsend.exe" -i 1000 -m "【ZBX_Severity:{TRIGGER.SEVERITY}】{TRIGGER.NAME}:{TRIGGER.STATUS}"
```

マクロの詳細は以下の URLを参照。

[https://www.zabbix.com/documentation/2.0/manual/appendix/macros/supported\\_by\\_location](https://www.zabbix.com/documentation/2.0/manual/appendix/macros/supported_by_location)

# アラートバーストの検証



## ◆ 検証内容

スクリプトを使って 100件のSNMPトラップを連発

## ◆ 検証結果

- 1秒以内に100件全てのSNMPトラップの送信が完了
- Zabbixへの登録は約2秒で完了
- JP1への登録は7秒で完了

## ◆ 備考

本検証中に右記のメッセージが JP1上で出力された。  
本メッセージはJP1へ大量のイベントを通知したため、  
イベント生成が滞留しデフォルトのしきい値 10件を超過したため発生したメッセージとなる。  
なお、全SNMPトラップがイベントとして登録された後、回復していることを確認できた。

## ◆ 検証環境

CPU	Intel Xeon X3430 2.40GHz
Memory	8GB

The screenshot shows two instances of the 'Event Details' window. The top window shows event ID 00003FA6 with a message indicating that the number of commands exceeded the limit of 10. The bottom window shows event ID 00003FA6 with a message indicating that the number of commands is 0.

属性名	属性値
イベントDB内通し番号	1789
イベントID	00003FA6
発行元プロセスID	3368
登録時刻	12/04 12:45:00
到着時間	12/04 12:45:00
発行元ユーザーID	-1
発行元グループID	-1
発行元ユーザー名	SYSTEM
発行元グループ名	
発生元ホスト名	jp1-server
発行元イベントサーバ名	jp1-server
発行元IPアドレス	192.168.10.10

メッセージ

KAVB2071-W jp1-serverでjp1-serverから要求されたコマンドの先行入力数が閾値[10件]を超えました。

# 制限事項／注意事項



## 1. イベントの登録順序

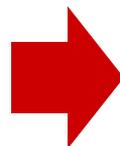
JP1サーバーがメッセージ受信順に  
イベント登録される



{EVENT.TIME} マクロを使い  
メッセージ内に時刻を記載

## 2. SNMPトラップやログ監視の回復イベント

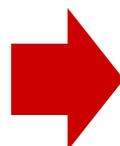
Zabbixでは閾値にマッチしないメッセージを  
受信すると回復してしまう



JP1の設定でイベント登録を除外  
Zabbixの設計で対処

## 3. ログの内容や、受信したSNMPトラップのメッセージの通知

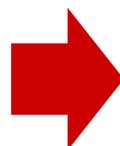
"(ダブルクォート)などで  
メッセージがきれってしまう



スクリプトで整形

## 4. イベントコンソールの『登録ホスト名』

jevsendではソースホストを指定できない



{HOST.NAME}マクロを使い  
メッセージ内で障害発生元ホストを記載

# 他のOSSでマネージャーとジョブコンを代替



さらに他のOSSを利用することでJP1/IMとJP1/AJSの代替も可能

Hatohol



Zabbix



JobDes



Job実行  
サーバー

監視対象サーバー

◆ JP1/PFMなど  
OSS監視ツール『Zabbix』を利用してIMに通知

◆ JP1/NNM  
用途に応じて検討

◆ JP1/AJS  
ジョブコントローラー『JobDes』を利用

◆ JP1/IM  
OSSの監視コンソール『Hatohol』を使って代替

# 監視ツールにジョブ管理機能をプラス！ HTL-JobDes



JobDesは、Hinemos®のジョブ管理機能をWEBブラウザ上から管理/設定することを可能としたブラウザ操作型のジョブコントロールツールです。

他監視ツールと連携させることで、御社の運用管理にこれまでの「監視」機能に加え、「ジョブ管理機能」を手軽にプラスすることができます。

## ■ JobDes管理コンソール TOP画面

The screenshot shows the JobDes management console interface. At the top, there's a navigation bar with 'Holon Technology' and 'ログアウト'. Below it, there are tabs for 'ジョブ', 'カレンダー', 'リポソトリ', 'アカウント', and 'メンテナンス'. The main area is divided into two panes. The left pane shows a tree view of job groups, including 'ジョブ連携正常サンプル (Z\_99\_SAMPLE\_JOBS)', 'ジョブ連携エラーサンプル (Z\_99\_SAMPLE\_JOBS\_ERR)', and 'Zabbix通知 (ZABBIX\_ERROR\_NOTICE)'. The right pane shows a table of jobs with columns for 'ジョブID', 'ジョブ名', '種別', 'ファシリティID', 'スコープ', '待ち条件', and 'オーナーロールID'. Below the tree view, there are sections for 'ジョブ[ジョブ詳細]' and 'ジョブ[スケジュール予定]', each with a table of job details and schedules.

ジョブID	ジョブ名	種別	ファシリティID	スコープ	待ち条件	オーナーロールID	編集
JOB_101	JOB_101	ジョブ	zbxvm20-vmware	zbxvm20-vmware	いいえ	ALL_USERS	hine
JOB_102	JOB_102	ジョブ	zbxvm20-vmware	zbxvm20-vmware	はい	ALL_USERS	hine

ジョブの作成、編集も可能  
カレンダーの利用や、手動によるジョブの再実行などもおこなえます。

他監視ツールとの連携が可能  
Zabbixとの連携実績有り！これまでのZabbix運用にジョブ管理をプラス！

WEBブラウザで一元管理  
管理コンソールはWEBブラウザからご利用頂くことができます。

**Hinemos®のジョブ管理機能で制御できる機能をJobDesで設定/実行することができます！  
ジョブコントロールツールに高額な商用製品を追加導入する必要はもうございません！**

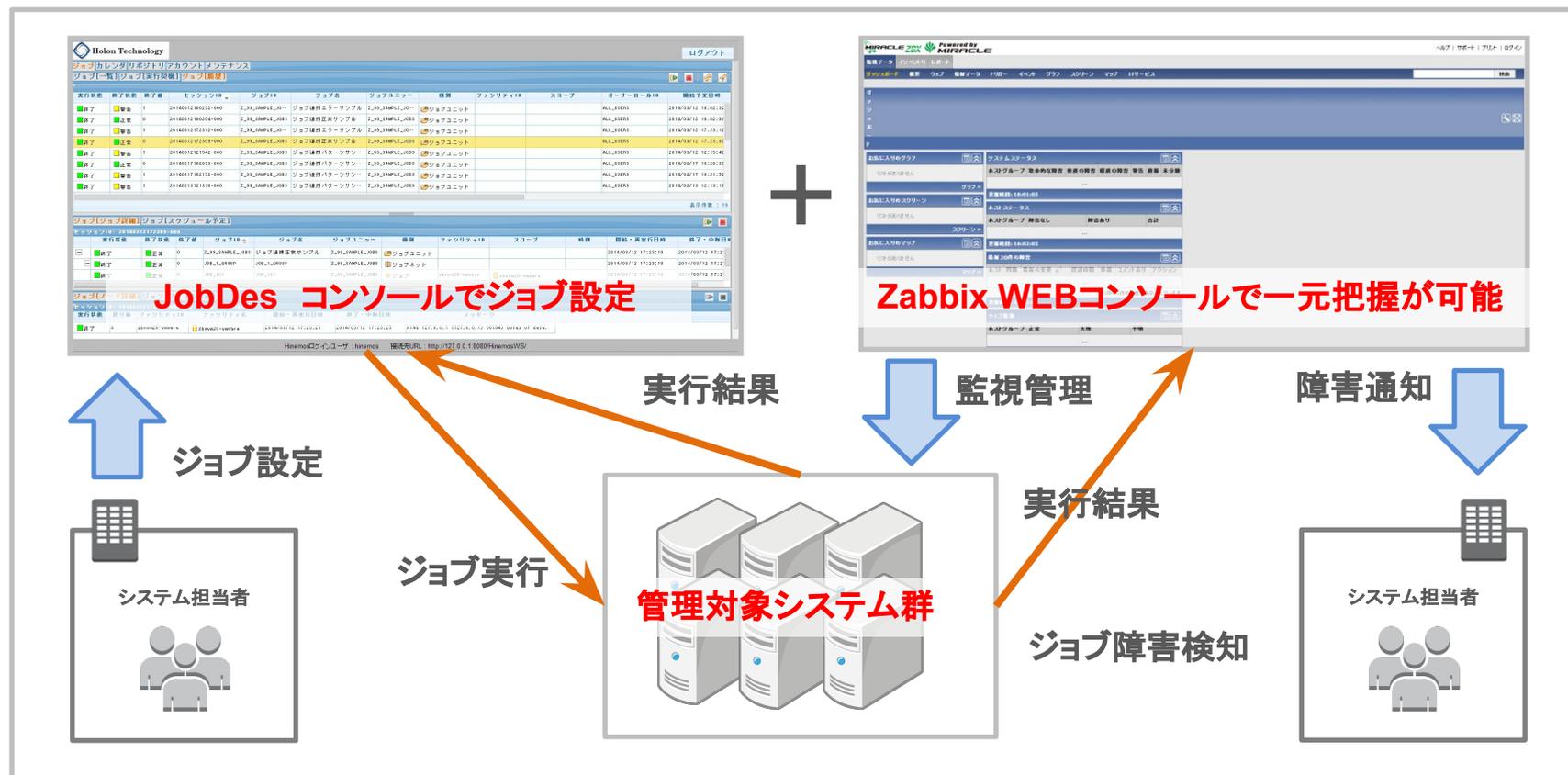
Do the Next, Open your Window

MIRACLE

# MIRACLE ZBXとの連携も可能



既に監視ツールを導入されている場合でも、JobDesなら容易にジョブ管理機能をプラスできます。ホロンテクノロジーではZABBIXとの連携実績により、ZABBIXコンソール画面からJobDesでのジョブ実行結果の確認を実現しています。



Do the Next, Open your Window

MIRACLE

複数の運用管理ソフトを集約できる監視コンソール  
OpenStack対応！！

## OSS運用統合ソフト Hatohol

The screenshot shows the Hatohol web interface. At the top, there's a navigation bar with 'Hatohol' logo and menu items like 'ダッシュボード', '概要トリガー', '概要アイテム', '最新データ', 'トリガー', 'イベント', '設定', and 'ヘルプ'. On the right, there's a user profile 'admin' and a '監視中' (Monitoring) status. Below the navigation, the main content area is titled 'イベント' (Events). On the left, there's a sidebar with 'サマリー' (Summary) section, including a dropdown for '全て(31日間)', buttons for 'すべてのイベント', '未対処の重要イベント' (with a red badge '8'), and '重要イベント'. The main table lists events with columns: '対処' (Action), 'ステータス' (Status), '深刻度' (Severity), '期間' (Period), '監視サーバー' (Monitoring Server), 'ホスト名' (Host Name), 'メッセージ' (Message), '現在のトリガーステータス' (Current Trigger Status), '継続時間' (Duration), and '優先度' (Priority). Callouts highlight: '表示のフィルタリング機能が充実' (Rich filtering), '重要度に応じた色の変更が可能' (Color change by severity), '対処のマーキングが可能' (Marking actions), and '未対処の重要イベント数を表示' (Displaying number of unhandled important events). A '設定' (Settings) button is circled in red.

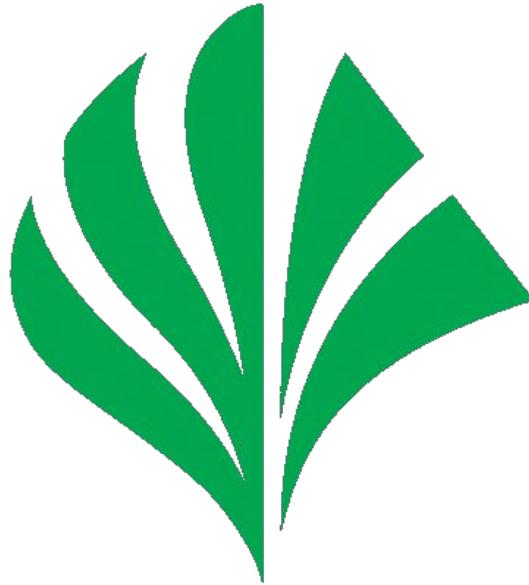


運用管理のコストダウンにはOSS

既存ツールとOSSの連携で  
運用負担を最低限に段階的な移行

製品によってライセンスポリシーが  
異なるため注意が必要





# MIRACLE

本セミナーおよびミラクル・リナックス製品に関する  
お問い合わせは以下までお願い致します

**MIRACLE**  
**DIRECT** 



tel: **03-6205-9502**



fax: **03-6205-9501**



mail: **info@miraclelinux.com**