



Kompira

**運用を開発しよう**

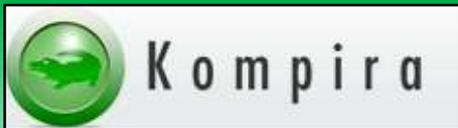


# 自己紹介

三角 正樹(みすみ まさき)    twitter:@micchan\_jp

運用に携わって15年

- ・ホスティングのサーバ構築・運用
- ・MSPのサービス企画、運用設計、システム構築、運用
- ・MSP会社立ち上げ(1500ノードの監視・運用)
- ・株式会社フィックスポイント立ち上げ(運用自動化ツール)



# 運用って？

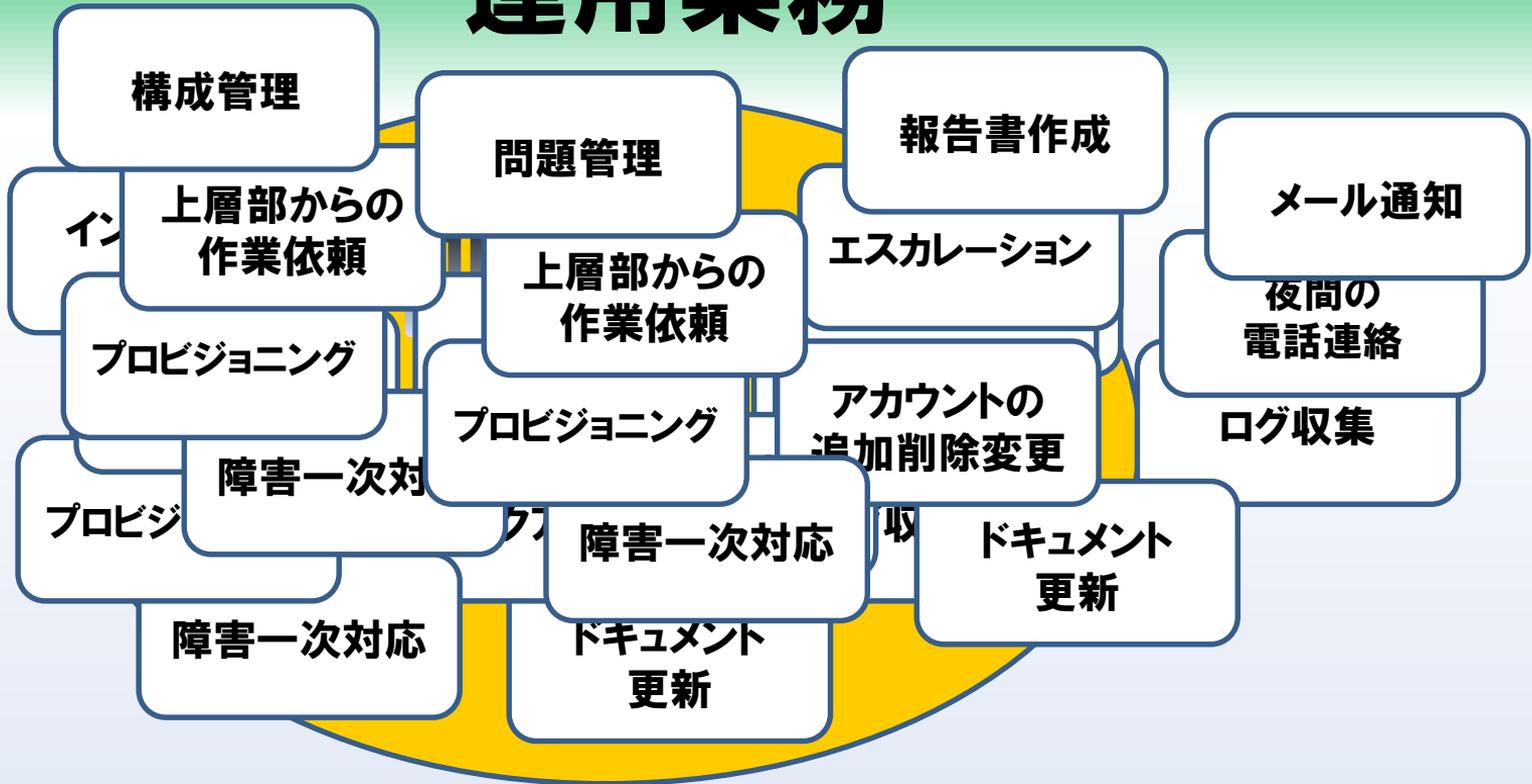
サービスイン後のもろもろ



導入、保守、障害対応、定常運用などなど



# 種々雑多な 運用業務



# 運用業務

## 維持業務

- ・ 障害一次対応
  - ・ アラート通知受け
  - ・ 確認
  - ・ ログ収集
  - ・ メール通知
  - ・ エスカレーション
- ・ 定常運用
  - ・ バックアップ
  - ・ リストア
  - ・ SSL証明書更新
  - ・ ユーザの追加、変更、削除
  - ・ 構成管理情報の取得
  - ・ ドキュメントの更新
  - ・ セキュリティアップデート

## 改善業務

- ・ サービス改善
- ・ パフォーマンス改善
- ・ 業務改善
- ・ 予防保守



Kompira

# 運用マンの憂鬱

## 運用維持業務は減点方式

トラブルがなくて（常に100点  
点が）当たり前、  
誉められない、評価されない

普段どんなに完璧にやっ  
ていても、トラブルが発生  
すると何をやっているんだと  
怒られる（減点）



担当者のモチベーションは保て  
ない。前向きな仕事をしたがら  
ない。新しい技術についていけ  
ない

モチベーション高い人はこな  
い、来ても離れていく



Kompira

# どうやって 解決するか？



人が足りてないので  
人を増やしましょう！



Kompira

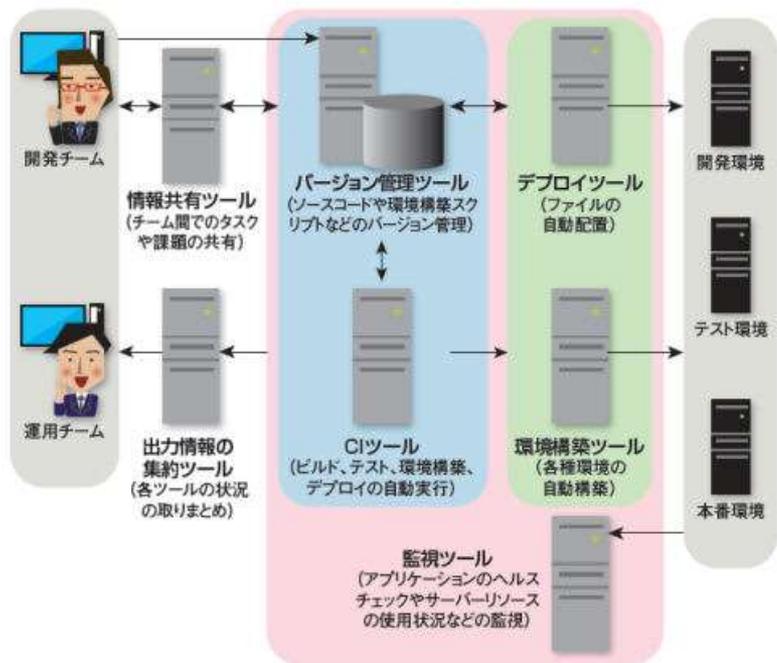
# どうやって 解決するか？



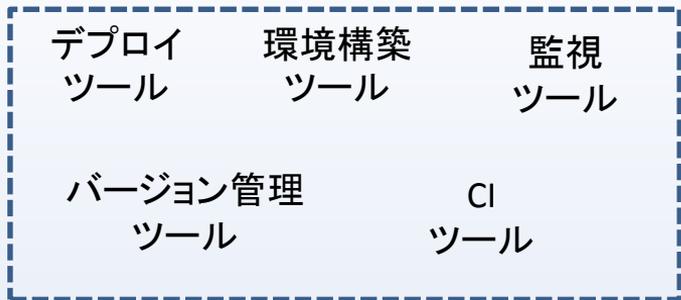
用をシステムの出来  
来のように開発して効率  
化しよう！

# DevOps

最近では、DevとOpsが協力しながら、  
ビジネスのために継続的に成果を出すDevOpsが人気



# DevOps

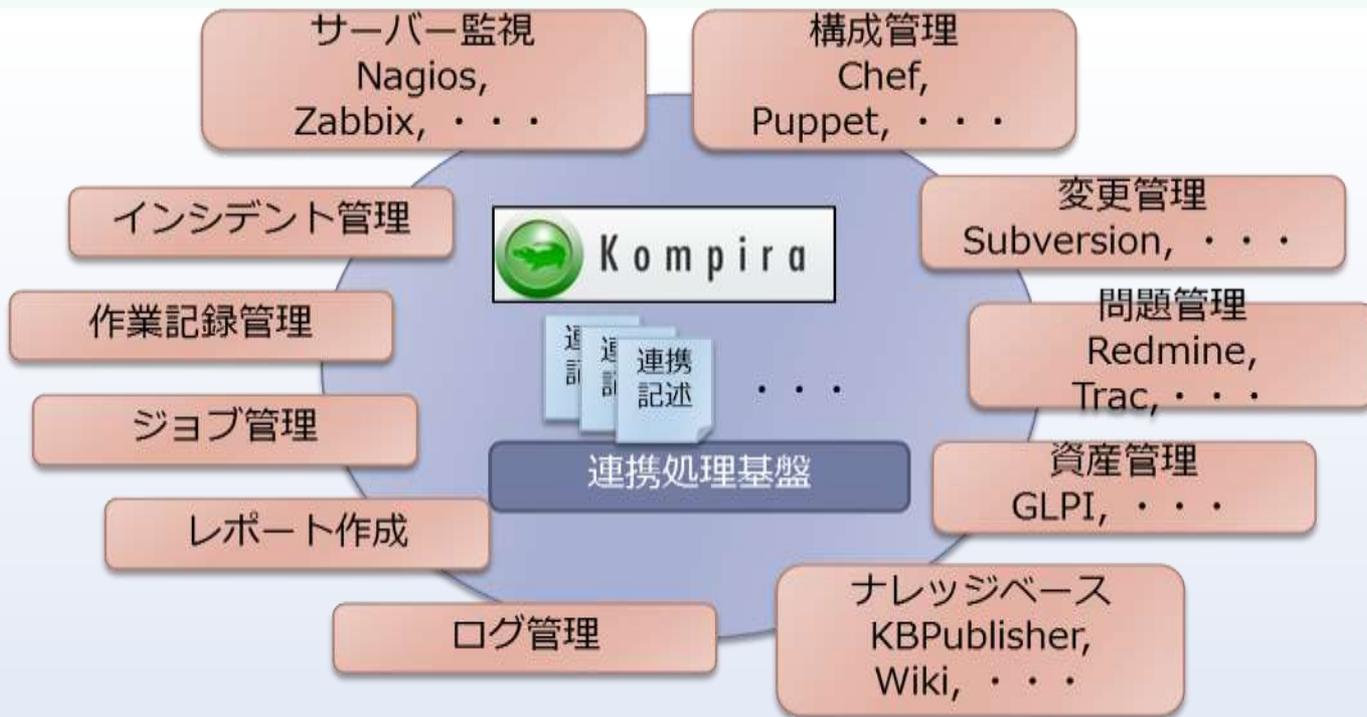


構成管理ツール 問題管理ツール ログ管理ツール ....

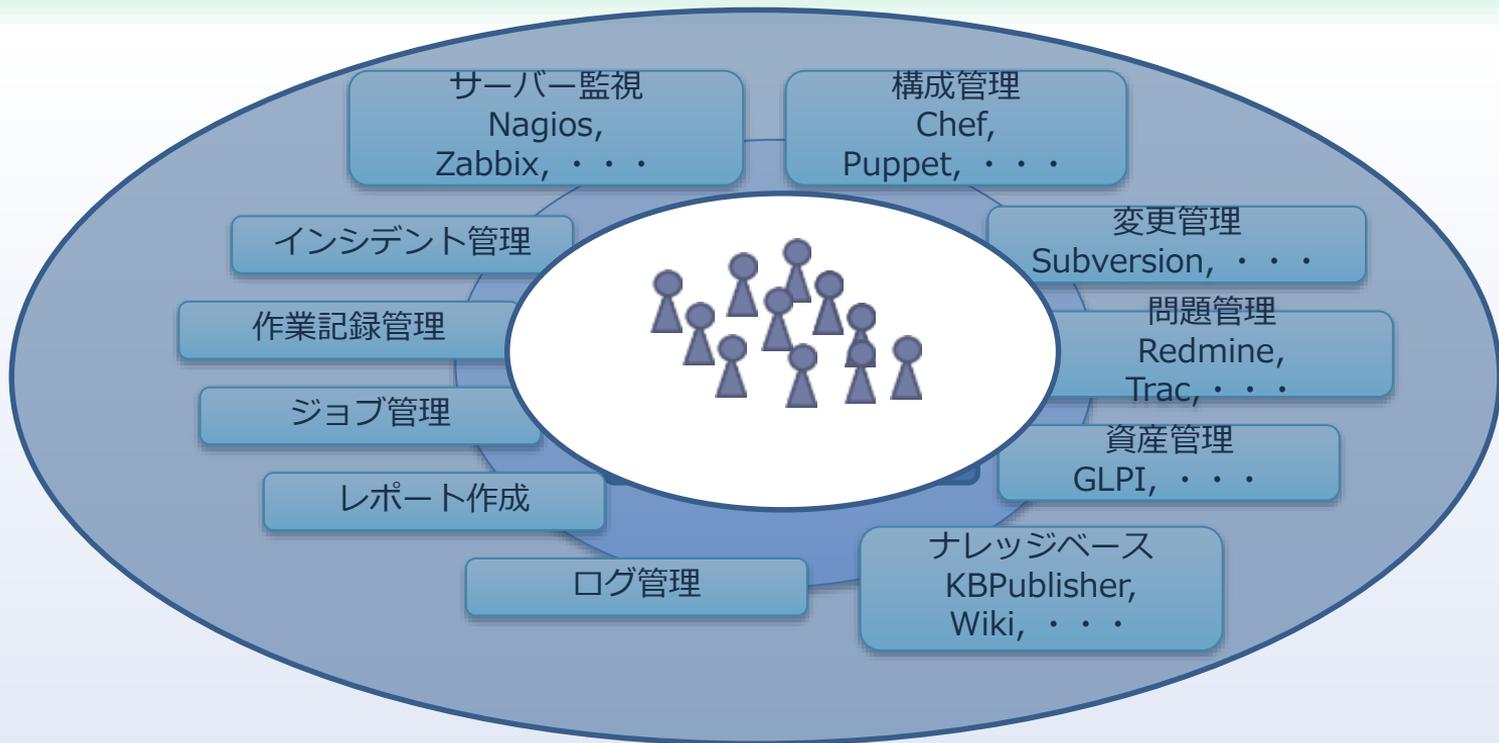
前頁の図のツールの連携範囲

こちら側の連携はまだまだ

# Kompira



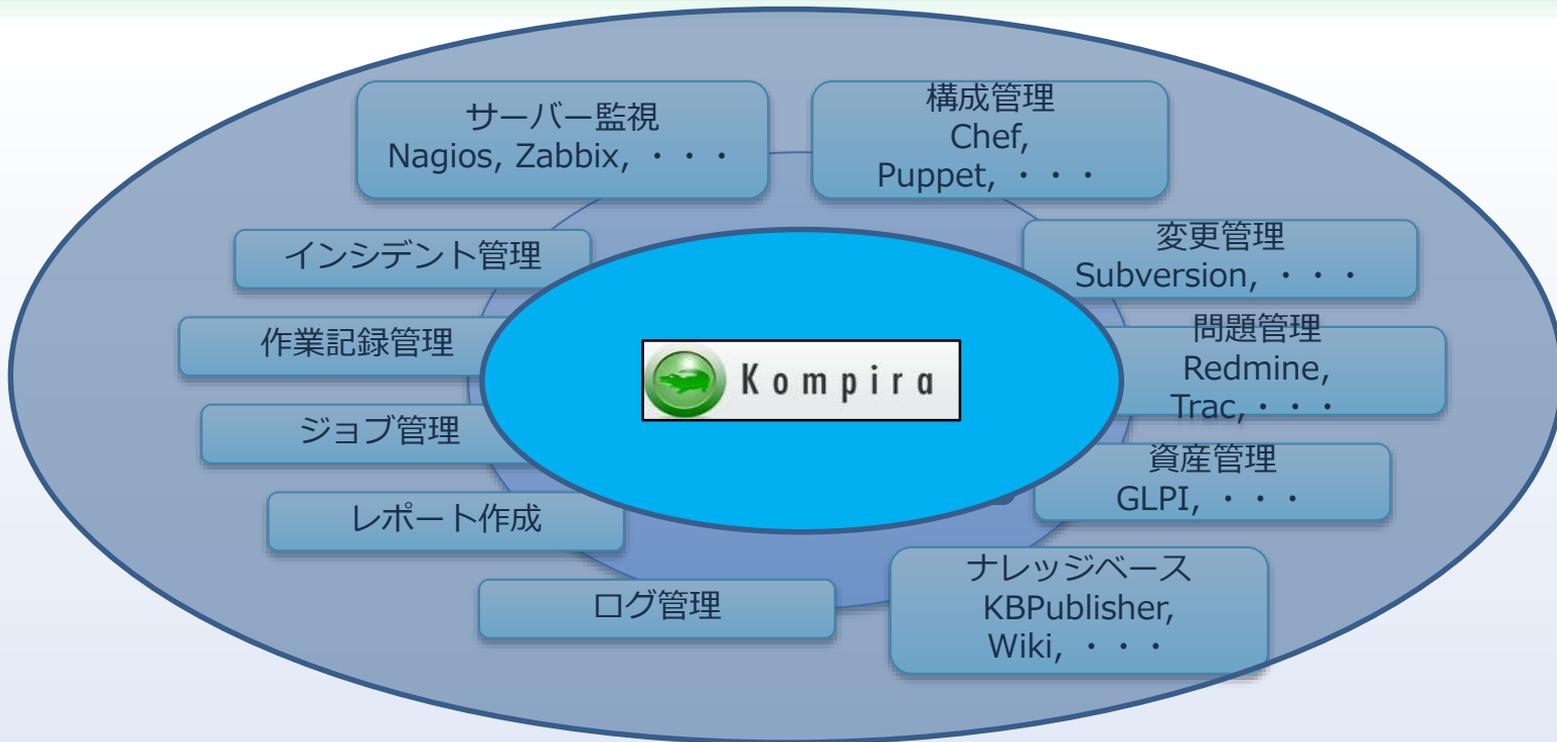
# Kompira



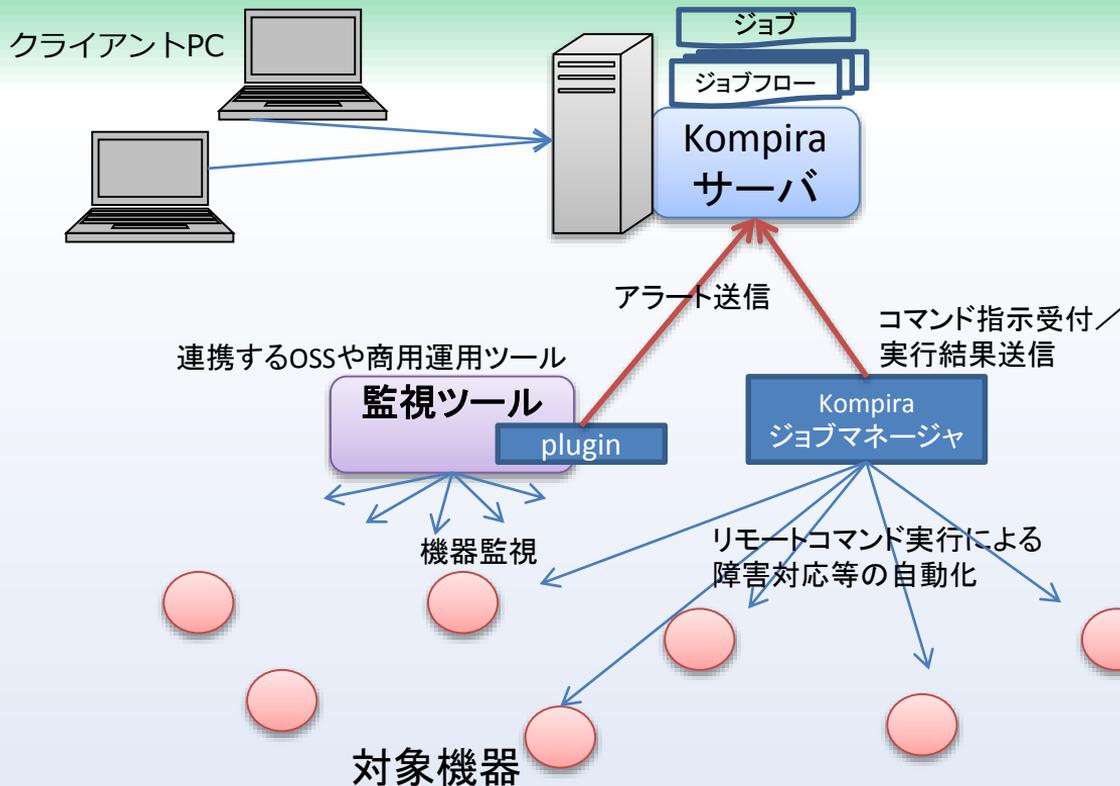


# Kompira

## バーチャルオペレータ



# Kompira構成



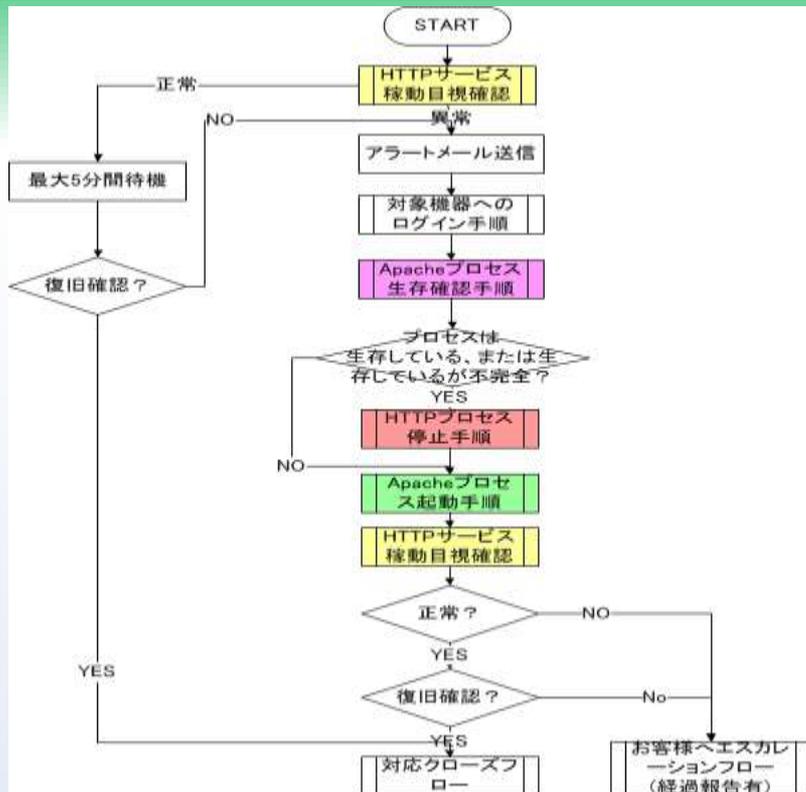
## 自動化の流れ

1. 自動化用のジョブ、ジョブフローをKompiraサーバにて作成
2. 既定のトリガーにてジョブやジョブフローが実行
3. Kompiraジョブマネージャへコマンド指示
4. Kompiraジョブマネージャから対象機器にリモートコマンド実行
5. 実行結果をジョブマネージャからKompiraサーバに結果送信

## 動作環境

アーキテクチャ  
X86/64bit  
オペレーティングシステム  
Linux CentOS 6.0以上  
Pythonバージョン  
Python 2.6系

# ジョブフロー例



# URL確認する

`[./URLチェック: URL=ENV.url, BODY=ENV.body, incident=incident]`

`-> { case |`

`'OK': [./インシデント出力: '問題ないのでクローズします', incident]`

`-> [incident.update: status='CLOSED'] -> exit()`

`'NG': [./アラートメール送信: ENV, incident] }`

`-> [./インシデント出力: 'Apacheプロセスの生存確認を行います', incident]`

`-> { apache=9 | # フラグの初期化`

`[__host__ = ENV.host, __user__ = ENV.user, __password__ = ENV.password]`

`-> ['ps auxw | grep httpd | grep -v grep'] => [apache=$STATUS]`

`-> { case '${apache}' '${tomcat}' '${db}' |`

`'1 0 0': [./インシデント出力: 'Apache存在しない', incident]`

`-> [./復旧シナリオ/シナリオ1: from='start httpd', ENV=ENV, incident=incident]`

`}`

`-> [./URLチェック: URL=ENV.url, BODY=ENV.body, RETRY=10, incident=incident]`

`-> { case |`

`'OK': [./リカバリメール送信: ENV, incident] -> return()`

`'NG': [./インシデント出力: 'リカバリを確認できません', incident]`

`-> [./エスカレーションフロー: ENV] }`

# 運用を開発する課題



運用の事はわかりません

開発の人



プログラミングわかりません

運用の人

# 運用を開発する課題

1970年代、メインフレームの時代に、それまで一つの体制だった開発と運用を分けた事に始まる運用マンの悲哀。もはや40年解決できなかった積年の課題に。

DevOpsも出てきて、そろそろ積年の課題を解決する時期では？  
運用を開発する事で、解決していきましょう！

# 事例 (障害通知の自動化)

性能情報取得、アラート通知

- ・仮想サーバーの監視
- ・ネットワーク機器の監視
- ・障害予兆検知
- ・各種閾値監視
- ・ポートのUP/DOWN検知
- ・応答監視 (ping/http/tcp等)

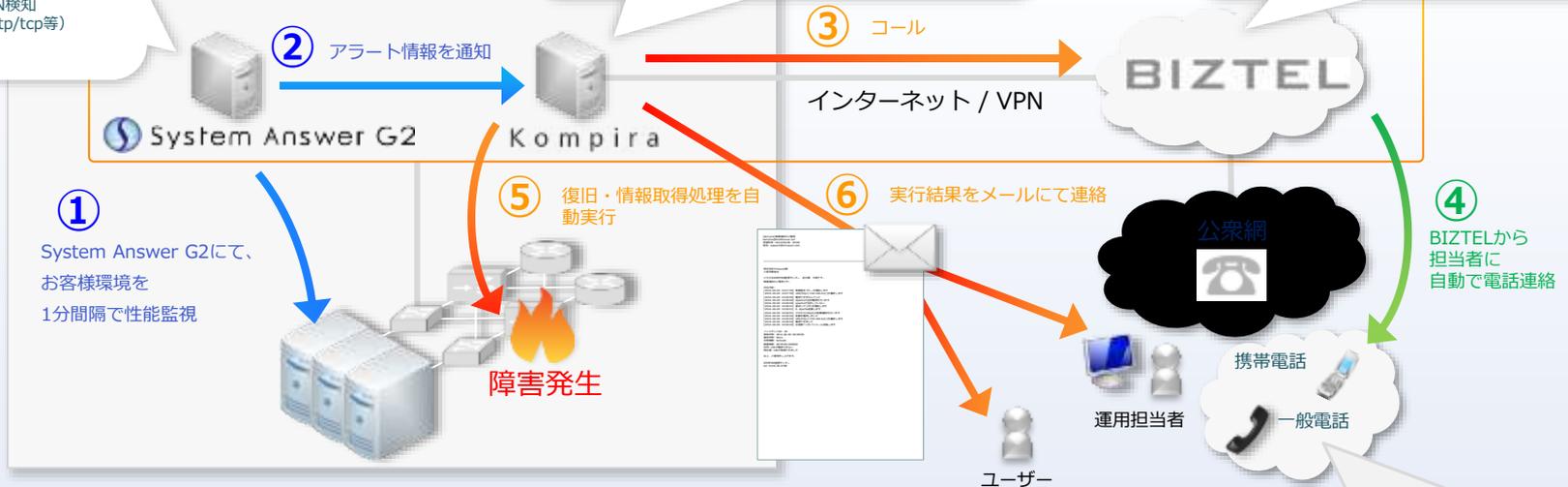
お客様環境

取得したアラート情報を基に自動コントロール

- ・シナリオによる自動処理
- ・メール・電話・FAXによる通知
- ・電話での自動音声入力により処理を分岐

クラウド型IPテレフォニーサービス

- ・電話での音声アラート通知
- ・電話でのIVRにより処理を実施



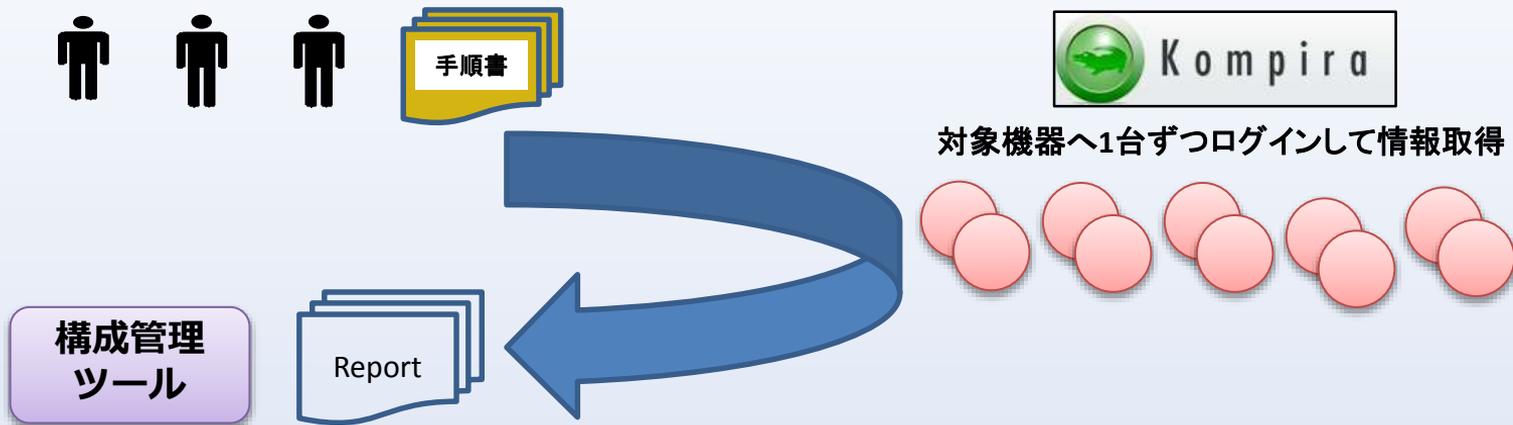
「自動音声」

ウェブサーバに障害が発生しています。自動復旧処理を実行する場合は「1」を、他の担当者にエスカレーションする場合は「2」を押してください。



# 事例 (構成情報管理)

定期的に行う必要のある構成情報の収集作業を自動化する。セキュリティ情報、資産管理情報と連携させることで、安定的な運用を実現できる。





# 事例 (運用の統合管理)

インフラエンジニア  
2名で24時間365日の  
サービス提供

株式会社アイダック CROMCA

