

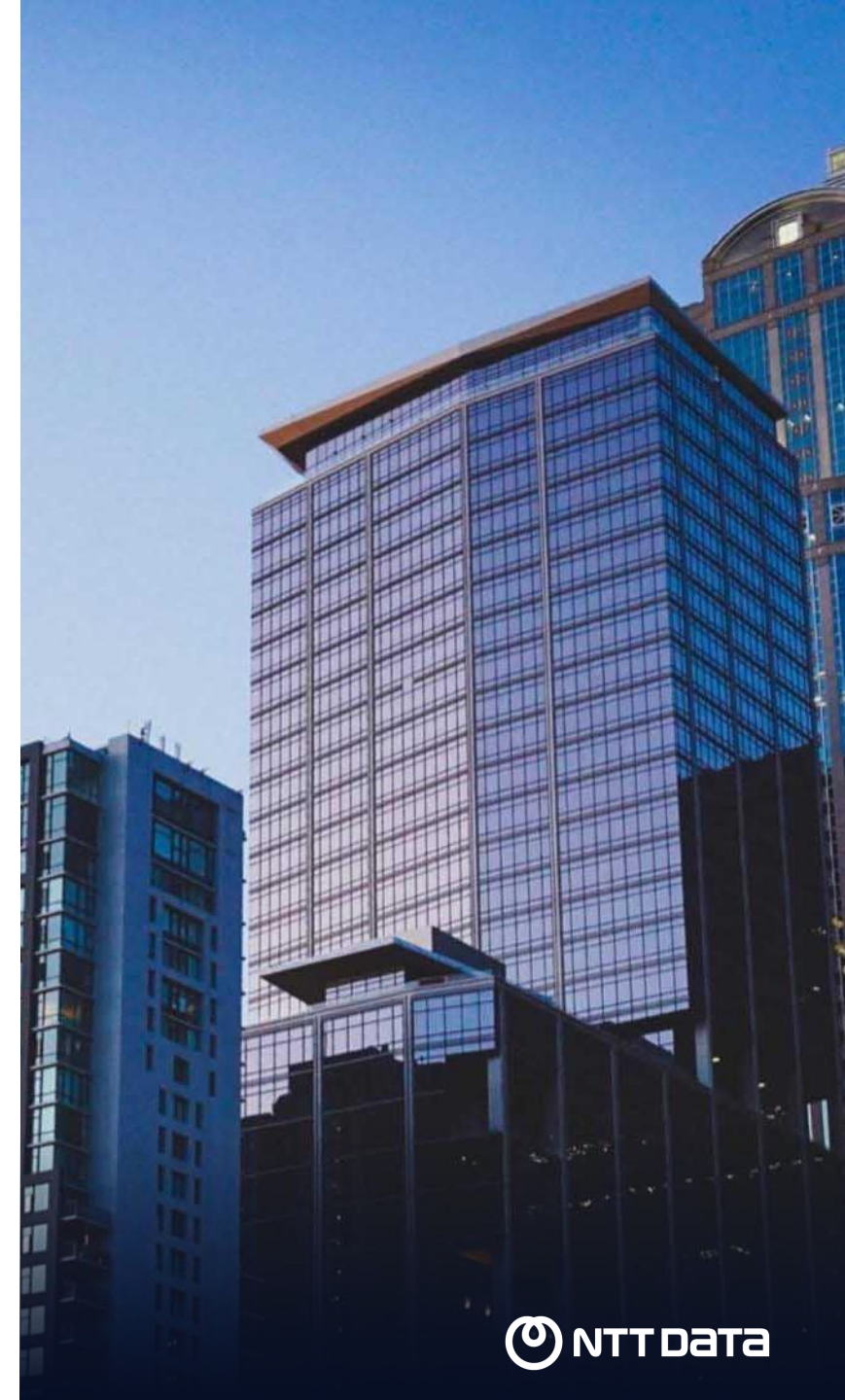
Hinemos AIエージェントと運用AI



Hinemos

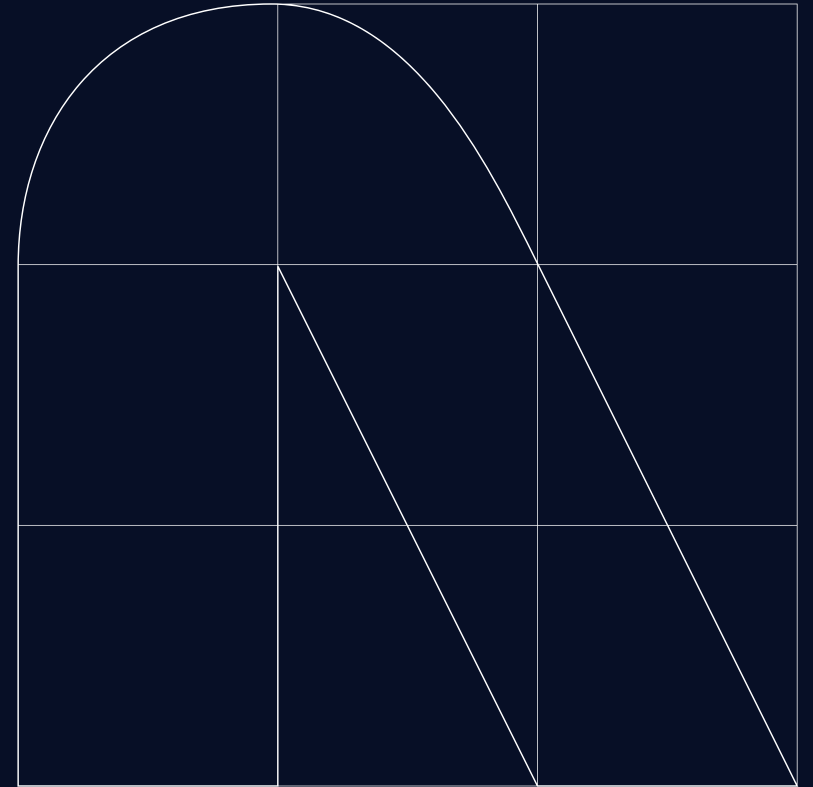
INDEX

1. 昨今のAIOpsのトレンド
2. AIドリブン運用
3. Hinemos AIエージェント
4. まとめ



01

昨今のAIOpsのトレンド



昨今のAIOpsのトレンド

異常検知と ログ解析の効率化

大量のログデータやシステムメトリクスを分析し、異常を検知する際に人が理解しやすい形で説明を生成

トラブルシューティングにかかる時間を短縮し、運用担当者の負担を軽減

予測とシナリオ生成

過去のデータを基に未来のシナリオを予測し、運用計画を自動生成

リスクを事前に回避し、システムの安定性を向上

インシデント管理と対応

インシデント発生時の適切な対応策の自動提案や、チケット管理システムに報告内容を自動生成

インシデント対応のスピードが向上し、運用チームの負荷が軽減

コードやスクリプトの 自動生成

運用管理に必要なスクリプトやコードを自動生成し、効率的なシステム管理を支援

手動でのスクリプト作成の負担を減らし、エラーのリスクを低減

ナレッジベースの 自動生成と更新

過去のインシデントや運用データを基にした、ナレッジベースの構築や更新を自動化

ナレッジの共有が容易になり、新しい担当者でも迅速に運用に対応できる

運用レポートの 自動生成

運用状況やシステムパフォーマンスに関するレポートを自動的に生成し、意思決定を支援

レポート作成にかかる時間を削減し、分析結果を迅速に提供

チャットボットによるサポート

運用チームやエンドユーザーからの問い合わせに対して、適切な回答をリアルタイムで提供

問い合わせ対応の負担を軽減し、運用チームの時間を節約

セキュリティ対応の効率化

セキュリティの脅威が検出された際に、対応フローを自動生成

セキュリティ対応の迅速化と正確性の向上

HinemosのAIOps対応

① 将来予測監視／変化量監視

② AI基盤による異常検知

③ HinemosとES/1 Sheltyの
連携ソリューション

④ Azure OpenAI Service基盤の
運用管理

⑤ Hinemos メッセージフィルタ

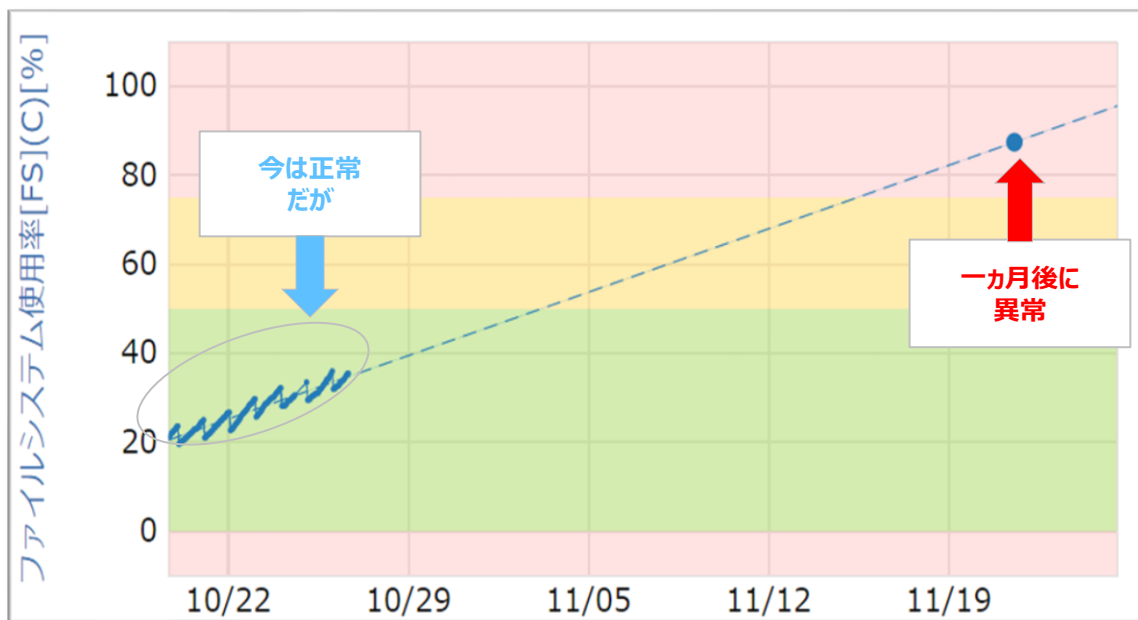
⑥ Hinemos AIエージェント

① 将来予測監視／変化量監視

収集した各種データから将来取り得る傾向(変化量)を予測、算出して予兆を監視

将来予測監視

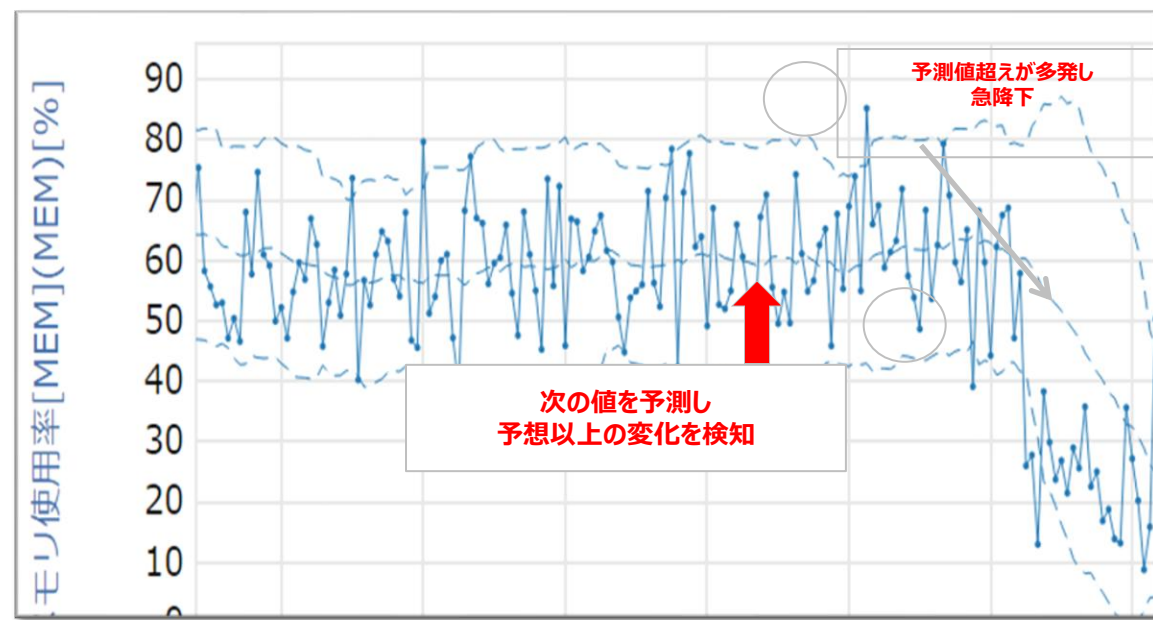
現在の傾向から将来的な閾値越えを監視



将来的な異常をいち早く検知

変化量監視

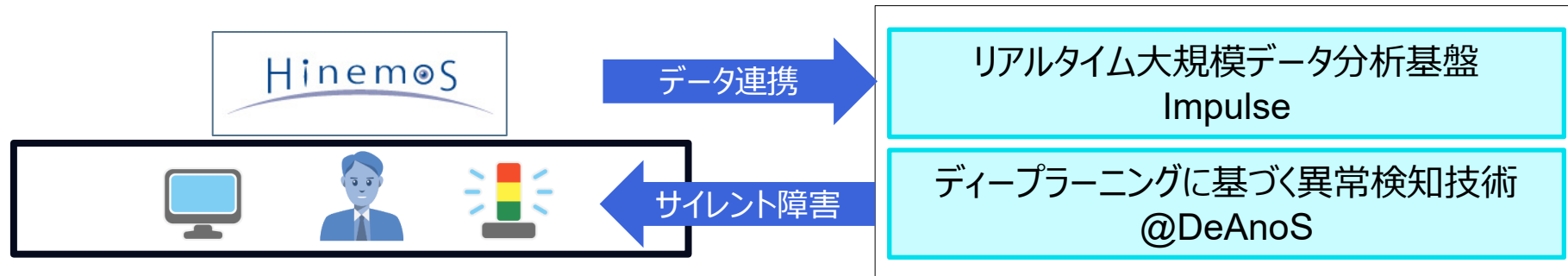
普段の傾向と外れたことを監視



傾向という曖昧な情報からサイレント障害を検知

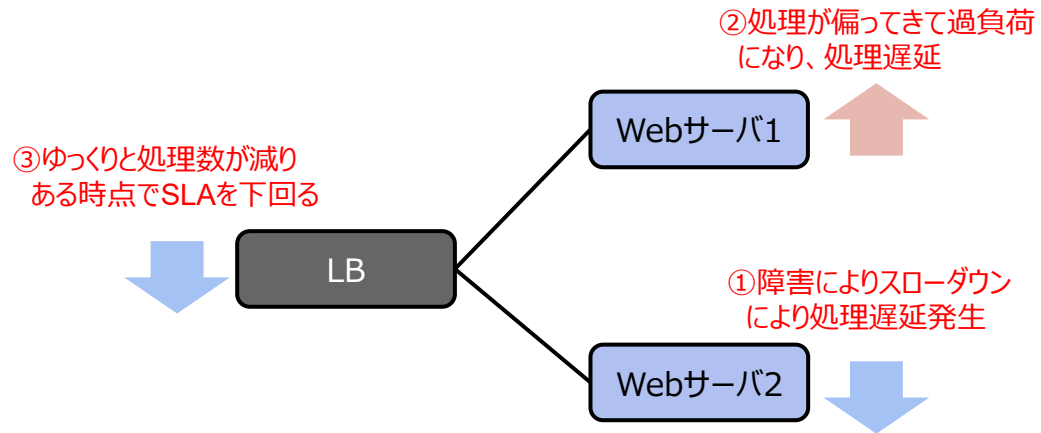
② AI基盤による異常検知

HinemosとAI基盤（Impulse／@DeAnoS）が連携することでITシステムの予兆検知から要因分析を実現



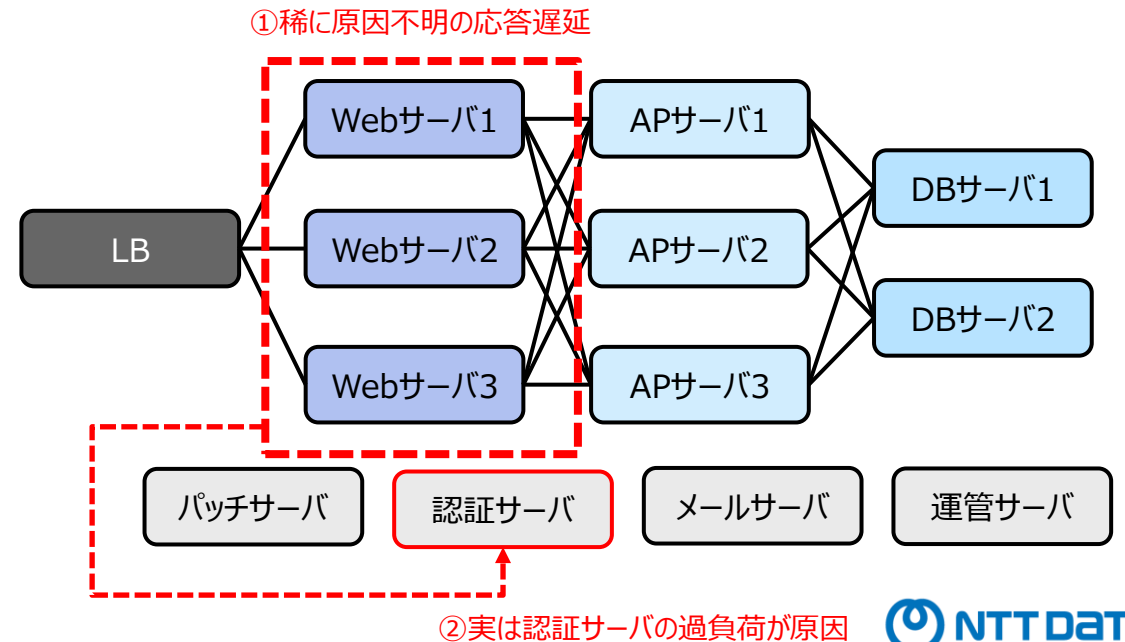
予兆検知

例) LBスローダウンの早期検知



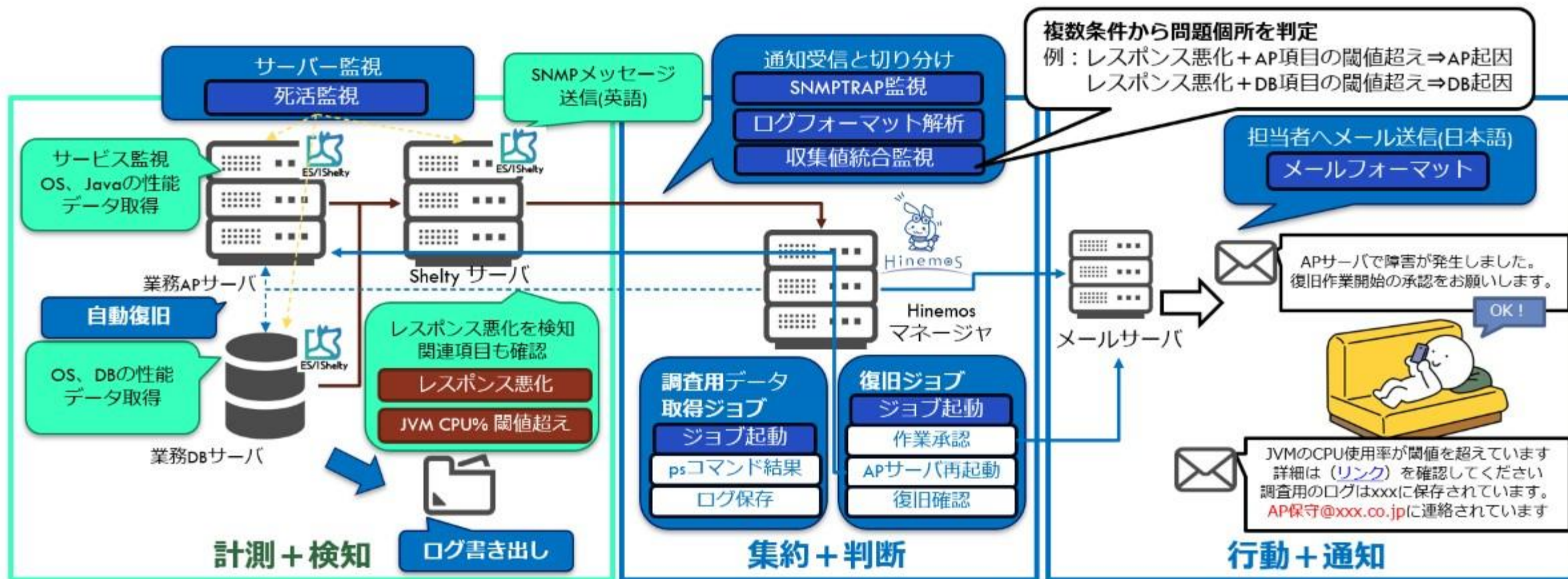
要因分析

例) Webサーバ応答遅延の要因分析



③HinemosとES/1 Sheltyの連携ソリューション

Hinemosと日本市場にベストマッチするAPM製品「ES/1 Shelty」の連携



オペラビリティと障害検知からの一連の対応を自動化

④ Azure OpenAI Service基盤の運用管理

AI基盤に特有の管理項目

AI基盤の管理の課題

ITシステム全体の運用管理の一元化

Hinemosによる、Azure OpenAI Serviceの統合管理

Azure OpenAI Serviceの監視

稼働状況の監視
サービスの死活状態の監視

サービスの正常性

インスタンスの監視

キャパシティの監視
クォータ制限の監視

呼び出し数

トークン数

セキュリティの監視
不正な利用の監視

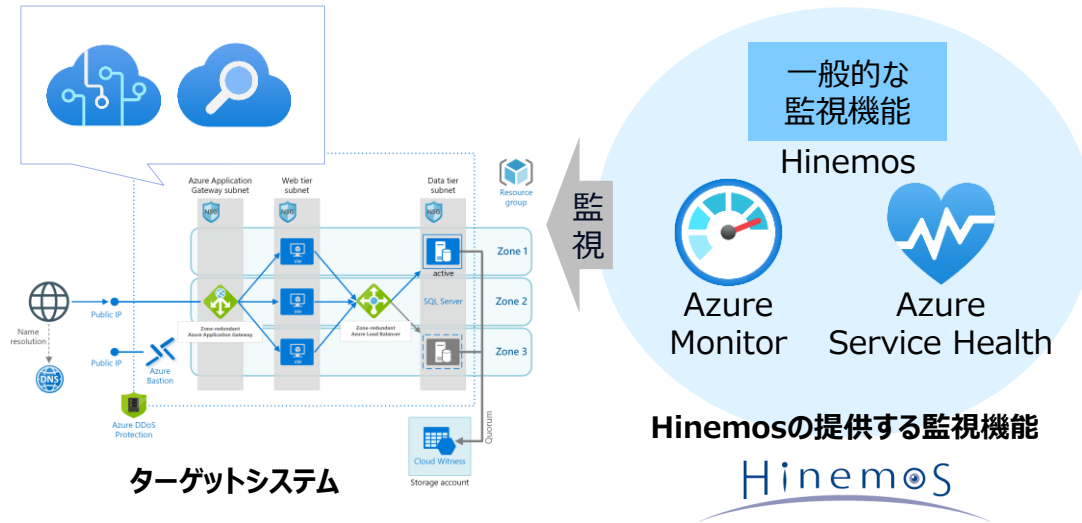
監査ログ

リクエスト/レスポンス

課金状況の監視
サービス利用状況の監視

コスト

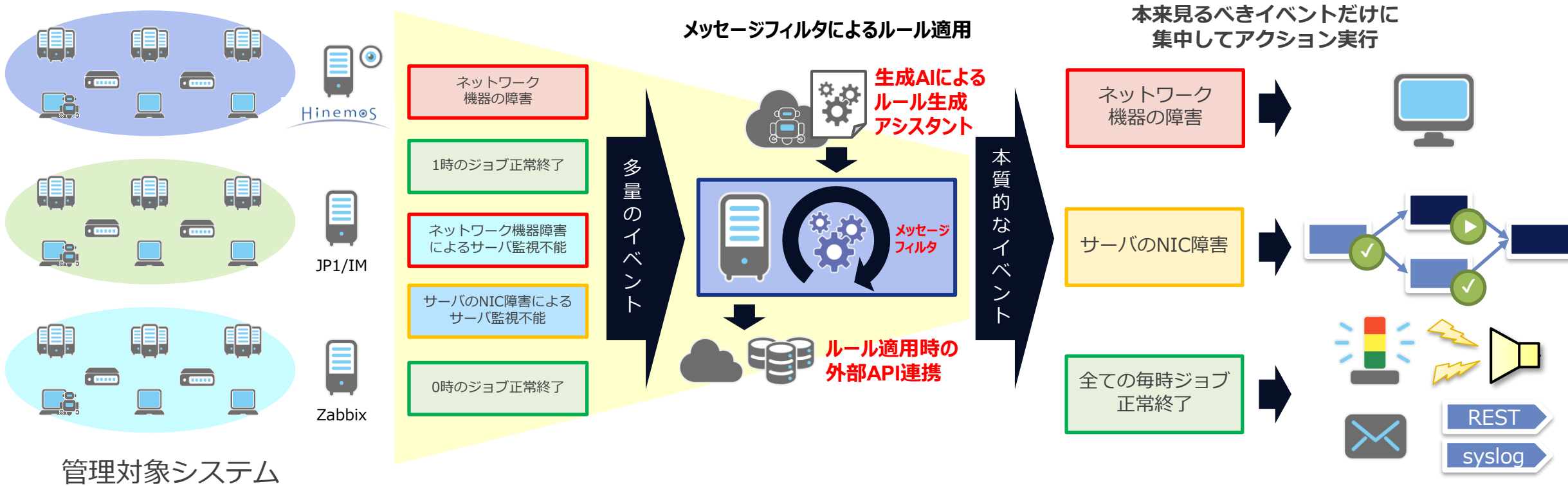
Azure環境の統合監視



Hinemosで、Azure OpenAI Serviceを含む **ITシステム全体の統合的な運用管理**を実現

⑤Hinemos メッセージフィルタ

Hinemosメッセージフィルタが本質的なイベントを発見し、運用の効率化と自動化を実現します



メッセージフィルタの6つの特長

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

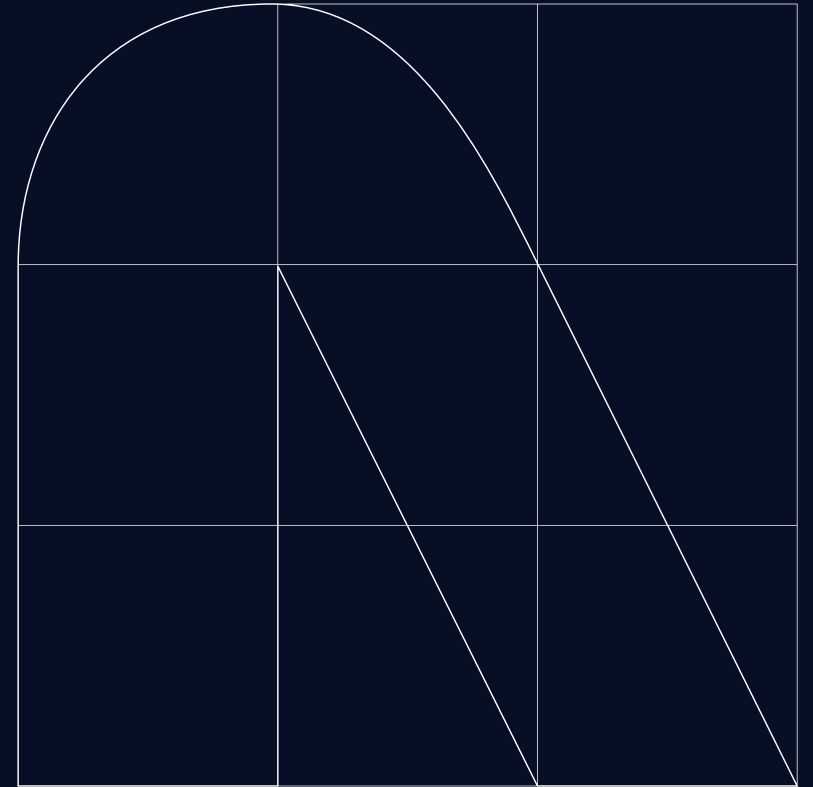
④Hinemosからのシームレスな導入

⑤簡易なルールの開発／管理

⑥簡易なルール運用

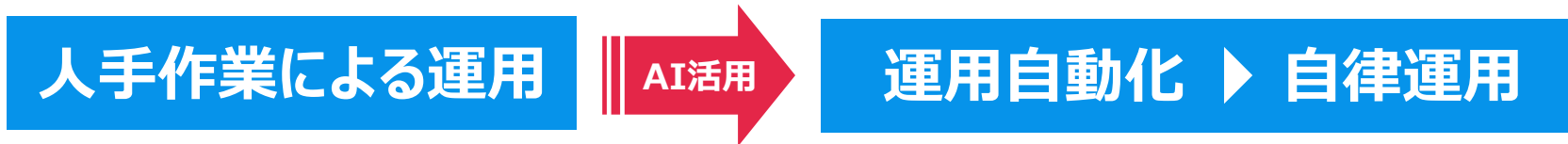
02

AIドリブン運用



AIドリブン運用とは

これまで人手作業で実施していたものをAIベースにすることで
飛躍的に運用を効率化し、生産性の向上を実現



得られる効果

- ① システム運用品質の向上
- ② 故障対応・運用改善のアジリティの向上
- ③ システム運用コストの低減

AIドリブン運用による運用課題の解決

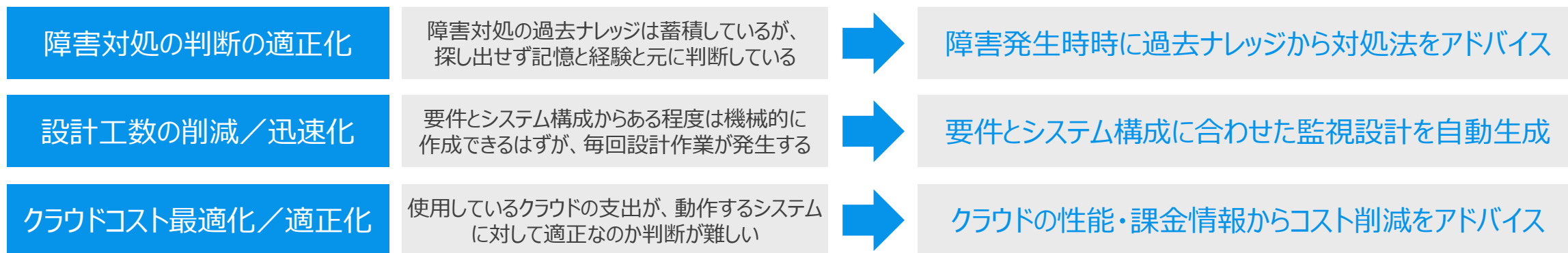
AIを運用に活用するAIドリブン運用により ITシステム運用の飛躍的な「品質向上」「アジリティ」「コスト低減」を実現

解決すべき課題：

- DX推進によるシステムの増加、複雑さに伴い運用品質の確保が難しい
- 人手に頼った運用では新たなシステムの受入れ、運用改善に迅速に対応できない
- クラウド活用等により不要なコスト原因の特定が困難

具体的な運用現場の課題例

生成AI活用による解決



AIドリブン運用によりもたらされる効果

人手ベースの課題

可用性・性能の担保

- ・全パターンの事前網羅は困難
- ・事後対応

イベント対応

- ・発生したイベントの静観処理
- ・人の判断による関連付け

故障対応

- ・人手で故障対応
- ・一部の高スキルなメンバ頼み

インシデント管理

- ・メンバによる知識の偏り
- ・過去対応の利活用が進まない

運用改善

- ・日々の作業優先で後回し
- ・改善点の洗い出しが困難

コスト管理

- ・システムを稼働させることが最優先コスト分析は後回し

AI活用

予防保全

- ・予兆検知による故障未然対応
- ・性能分析とキャパプラ

インテリジェントなイベント対応

- ・不要なイベントのフィルタリング
- ・必要なイベントを集約

故障対応の迅速化/自動化

- ・影響範囲特定
- ・故障原因分析

ナレッジベースのインシデント管理

- ・根本原因の解析
- ・分類/傾向分析

アセスメントによる運用改善

- ・生産性の分析
- ・改善ポイントの特定

データ分析によるコスト最適化

- ・クラウド課金情報分析
- ・リソース最適配置

システム運用品質の向上

人では得られなかった気づき
情報整理、分類、判断の精度の向上
ヒューマンエラーの撲滅
継続的な運用改善
予防保全

運用品質の向上による安定的な
サービス提供

サービス停止による機会損失の回避

故障対応・運用改善のアジリティの向上

新たなシステムリリース、システム改修
時の運用受入れの迅速化
運用業務の作業時間短縮
保守業務・故障対応の復旧迅速化

新たなITサービス拡充の迅速化

BCP(事業継続計画)の強化

システム運用コストの低減

非生産的な人手作業の低減
・繰り返し作業を減らす
・運用レポートの自動生成
インフラ管理の高度化/自動化

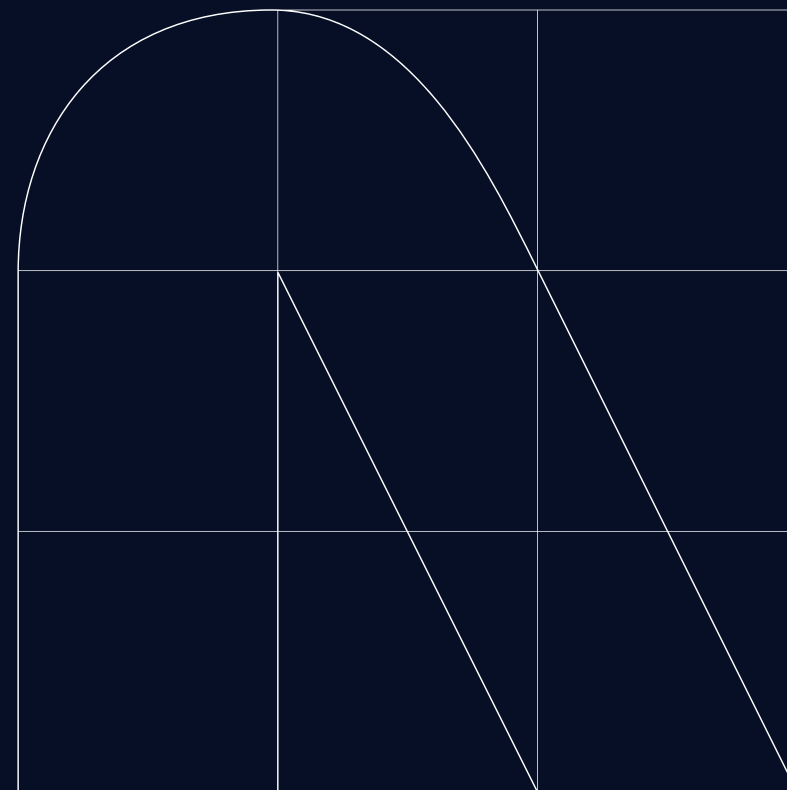
人的リソースの最適配置

不要なコスト原因の特定と排除

適切なリスクコントロールによる
コスト最適化

03

Hinemos AIエージェント



Hinemos AIエージェントとは

ver.7.2新機能

システム運用の確認から分析、調査、対処までの自動化をサポート

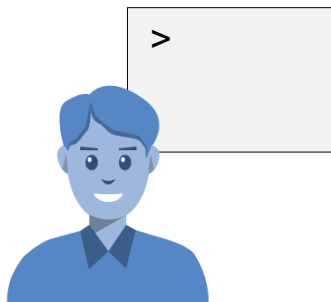
大規模言語モデル（LLM）
の活用

インターネット等の
一般的なナレッジの活用

Hinemosの蓄積した
運用イベントの活用

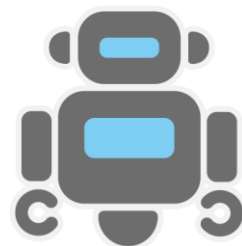
自動化エンジンとしての
Hinemosの活用

自然言語による問い合わせ



オペレータ・SE

AIとHinemosを連携



Hinemos AIエージェント

運用タスクの自動化をサポート・推進

確認

分析

調査

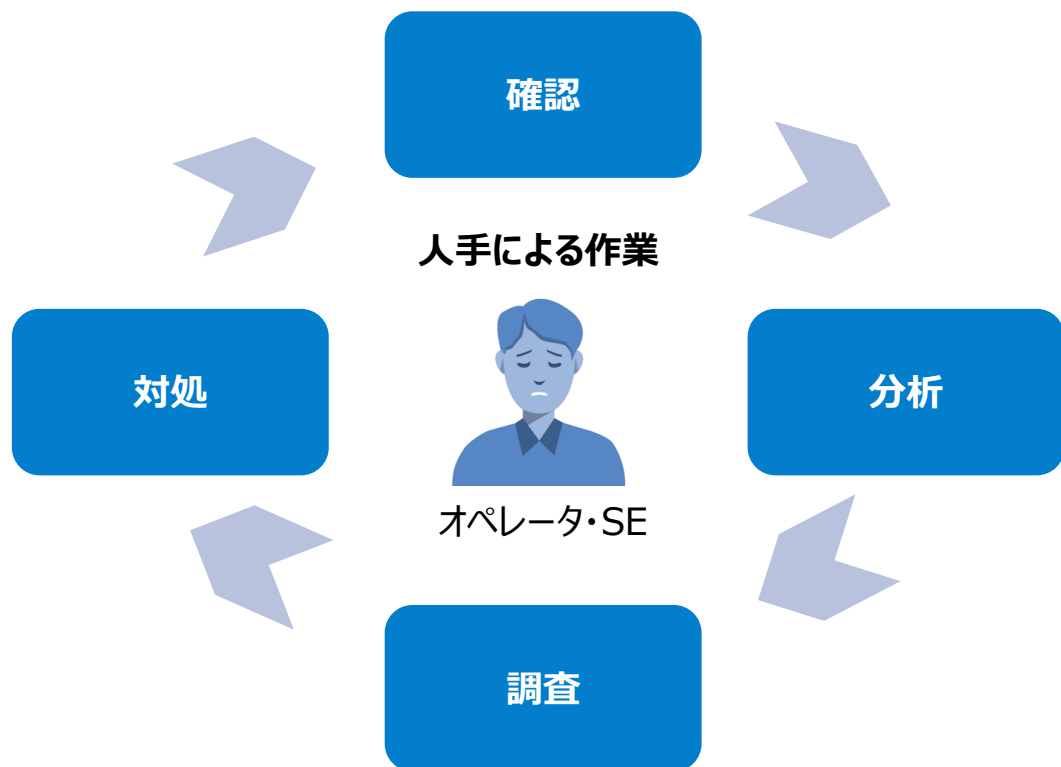
対処

自動化

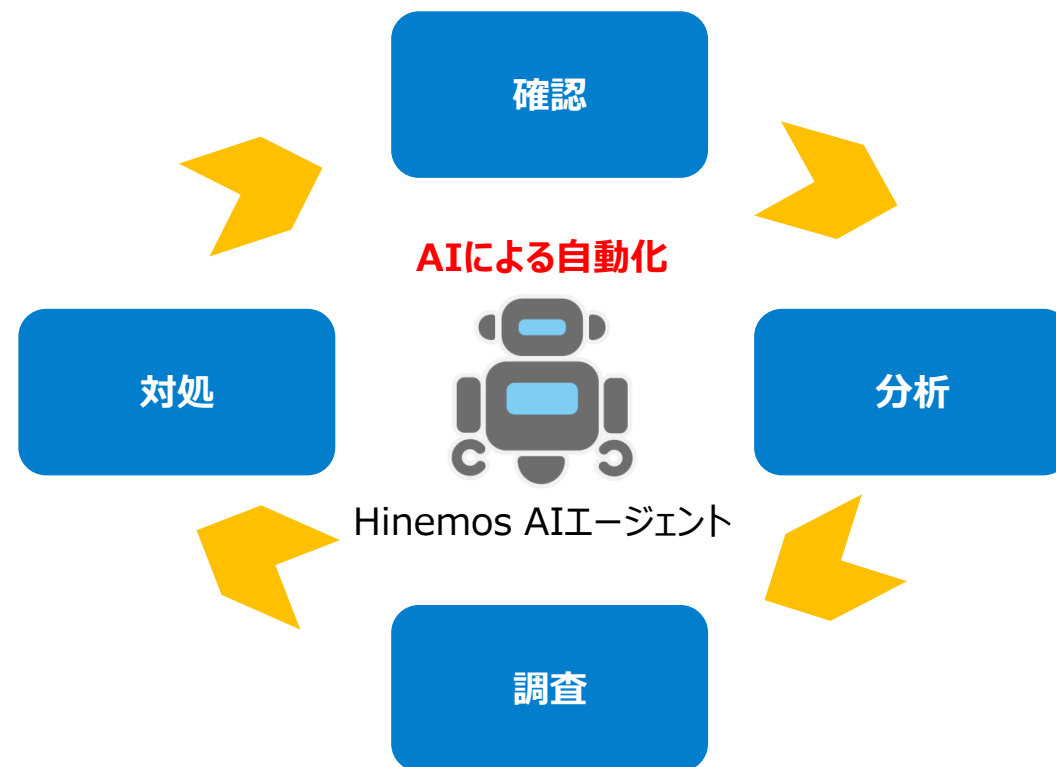
Hinemos AIエージェントで何ができる？

通常運用から障害イベント発生時の対処を、様々なナレッジを活用してより安全に実現

これまでの運用



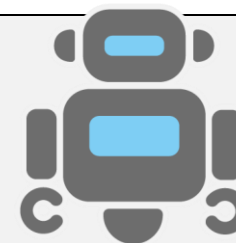
Hinemosが提供する運用



ユースケース 確認 異常の発生有無を知る

今のシステムの状態はどう？何かクリティカルなアラートは出ている？

今のシステムの状態はどう？何かクリティカルなアラートは出ている？



現在、Hinemos上ではクリティカル(危険)なステータスが1件出ています。

- 監視項目ID: `Ping03`
- プラグイン: PING監視 (`MON_PNG_N`)
- 対象ノード(ファシリティID): `Node03`
- 重要度: 危険 (CRITICAL)
- 最終更新日時: 2026-03-02 11:36:41.009
- メッセージ:
`Packets: Sent = 1, Received = 0, Lost = 1 (100% loss)`

意味としては、`Node03` に対するPING疎通が取れておらず、100%パケットロスしている状態です。

****運用上の次のアクション (提案) :****

1. `Node03` へのネットワーク疎通確認
 - 他サーバから `ping Node03` / ホスト名 or IP で確認
 - 必要に応じて traceroute などで経路調査
2. `Node03` 自体の状態確認
 - 物理/仮想マシンの電源・ハイパーバイザ側コンソール確認
 - OSがハングしていないか、シャットダウンしていないか
3. 想定された停止かどうかの確認
 - メンテナンスや計画停止中であれば、監視一時停止やメンテナンスカレンダの適用を検討

必要であれば、このノードの詳細情報や、同ノードに関する他のイベント履歴 (WARNINGなど) も取得して状況を整理しますが、確認しますか？

ユースケース 分析 発生した事象をサマライズする

最近、どのスコープでアラートが多い傾向にある？

最近、どのスコープでアラートが多い傾向にある？

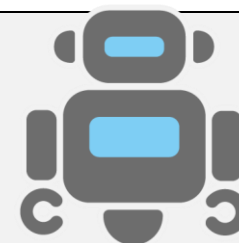


スコープごとのアラート傾向を出すには、「イベント履歴（監視履歴[イベント]）」を集計する形になります。

まず、どの範囲で確認したいか教えてください。

1. 期間
2. 対象の重要度
3. 対象スコープ
 - 無指定なら全スコープを対象

この3点を教えていただければ、その条件でイベント履歴を取得し、スコープごとのアラート件数を集計して、「どのスコープでアラートが多いか」を一覧でお出します。



条件は任せます

直近7日間のイベント履歴（危険／不明／警告）を全件検索して集計した結果、次の通りです。

- アラートが多いスコープ（ファシリティID）
 - ****Node03****：少なくとも ****100件以上****（危険）
 - 監視項目ID: `Ping03`
 - メッセージ: `Packets: Sent = 1, Received = 0, Lost = 1 (100% loss)` が連続発生

取得できたデータでは、****ほぼ全てのアラートが Node03（Ping監視）に集中している****状態です。
つまり、最近アラートが多いスコープは ****「Node03」**** になります。

推測としては、**Node03** へのネットワーク疎通が継続的に失敗している、もしくは Ping監視の閾値／リトライ設定によって短間隔で危険イベントが蓄積している**可能性**があります。

必要であれば：

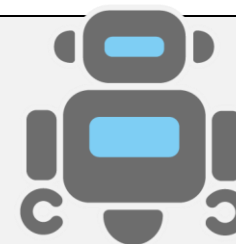
- Node03 への疎通確認（`repositoryPing` 相当、もしくは実環境で ping 実行）
- `Ping03` 監視の設定（監視間隔・リトライ回数・閾値）の見直し

と一緒に確認していきましょう。

ユースケース 調査 障害の事象を知る (1/4)

『Ping03』のエラー、具体的に何が原因か教えて

『Ping03』のエラー、具体的に何が原因か教えて



1. 事実として分かっていること

先ほどのイベントは、いずれも以下の内容です。

- プラグインID: `MON_PNG_N` (PING監視)
- 監視項目ID: `Ping03`
- ファシリティID: `Node03`
- メッセージ:
`Packets: Sent = 1, Received = 0, Lost = 1 (100% loss)`

Hinemos の PING監視(MON_PNG_N)は、監視対象ノードのIPアドレスに対してOSレベルの ping コマンドを実行し、その成否で死活を判定します。
このメッセージが出ている =

> Hinemosマネージャから Node03 に送った ICMP Echo (ping) に ****一切応答が返ってこなかった****

というところまでは、確定した事実です。

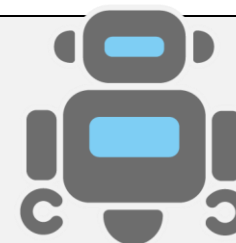
また、この「100% loss」のイベントが数分間連続しているので、

> 一時的な1パケット落ちではなく、****その間ずっと疎通不能状態が続いていた****

ことも事実と言えます。

ユースケース 調査 障害の事象を知る (2/4)

『Ping03』のエラー、具体的に何が原因か教えて



2. しかし、「具体的な1つの原因」はイベントだけでは特定不可

Hinemosのイベント（PING監視の結果）は「疎通できなかった」という結果だけを示しており、

- 電源断なのか
- OSハングなのか
- NIC故障なのか
- FW/セキュリティ製品によるICMP遮断なのか
- 経路上のスイッチ/ルータ障害なのか
- IPアドレス設定ミスなのか

といった「物理/論理的な原因」までは持っていません。

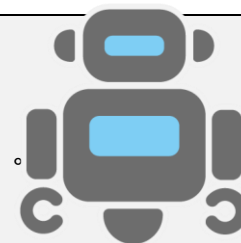
したがって、**イベント情報だけから「これが唯一の原因です」と断定することはできません。**
分かるのは「Node03に対して ping が通らない状態が続いていた」という事実までです。

ユースケース 調査 障害の事象を知る (3/4)

『Ping03』のエラー、具体的に何が原因か教えて

3. イベント内容から見て、現実的な原因候補

「Ping03 だけが NG」「Node01 / Node02 の Ping は OK」という状況から、以下のような可能性が高いと考えられます（推測です）。



1. **Node03 自体の障害**

- 電源断
- OS停止 / ハング
- 再起動中
- NICダウン など

2. **Node03 周辺のネットワーク障害**

- Node03 が接続されているスイッチポートのダウン
- VLAN/経路設定の変更ミスで、Hinemosマネージャからの経路だけ切れた
- ケーブル抜け・断線

3. **ICMP を遮断する設定・機器の影響**

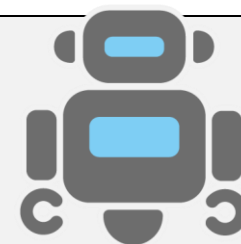
- Node03 のOSファイアウォール設定で ICMP (echo request/reply) が拒否されるようになった
- 間に入っているFW/セキュリティ製品で、Node03宛の ICMP がブロック、またはレート制限されている

4. **設定ミス**

- Hinemos の Node03 のIPアドレスが誤っている
- Node03 側のIP変更があったが、Hinemos側未更新

ユースケース 調査 障害の事象を知る (4/4)

『Ping03』のエラー、具体的に何が原因か教えて



4. 具体的な切り分け手順 (次にやると良いこと)

もしまだ原因調査中であれば、以下の順で確認するをお勧めします。

1. **現在の疎通確認**

- Hinemosマネージャが動いているサーバから、Node03 のIPアドレスへ手動で `ping` 。
- 可能なら `ssh` / `RDP` / 管理ツール (iLO/iDRAC等) でログインできるか確認。

2. **Hinemosからの疎通確認**

- Hinemosクライアントから Node03 (ファシリティID: `Node03`) に対して「ノードに対するPING実行」を試す。
- 同じく 100% loss になるか、今は復旧しているかで、**一時的障害か継続中か**が分かります。

3. **Node03側の状態チェック**

- 電源状態 (コンソールやリモート管理ツール)
- OSログ (システムログ、ネットワークログ)
- NICリンク状態 (link up/down)

4. **ネットワーク機器/FWのログ・設定確認**

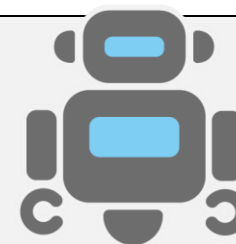
- Node03 の接続スイッチのポート状態
- 間にあるFWで、該当時間帯に ICMP のブロックログやエラーがないか
- 経路変更やACL変更の履歴

5. 必要なら手順を一緒に詰められます

ユースケース 対処 一次的な対処を実施する (1/4)

不安定なノードはある？ある場合、メンテしたいから監視を全て止めて

不安定なノードはある？ある場合、メンテしたいから監視を全て止めて



不安定なノードはあります。

1. 不安定なノード (監視ステータス)

- DB01
 - 監視項目ID: `PingDB` (PING監視 / MON_PNG_N)
 - 重要度: 危険
 - メッセージ: `Packets: Sent = 1, Received = 0, Lost = 1 (100% loss)`
 - 最終変更日時: 2026-03-02 17:52:32.008
- Node03
 - 監視項目ID: `Ping03` (PING監視 / MON_PNG_N)
 - 重要度: 危険
 - メッセージ: `Packets: Sent = 1, Received = 0, Lost = 1 (100% loss)`
 - 最終変更日時: 2026-03-02 17:53:11.010

※上記は監視履歴[ステータス]から、「警告以上 (CRITICAL/UNKNOWN/WARNING)」で絞って取得した結果です。

ユースケース 対処 一次的な対処を実施する (2/4)

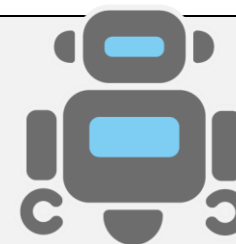
不安定なノードはある？ある場合、メンテしたいから監視を全て止めて

2. 停止対象になりそうな監視設定

該当ノードに紐づく監視設定を確認すると、以下のようになっています (抜粋) :

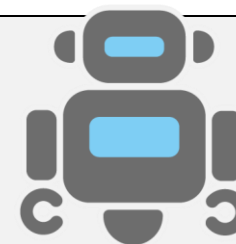
- DB01 向けの個別監視
 - `PingDB`
 - monitorId: `PingDB`
 - monitorTypeId: `MON_PNG_N` (PING 監視 / 数値監視)
 - facilityId: `DB01`
 - 監視有効: `true`
- Node03 向けの個別監視
 - `Ping03`
 - monitorId: `Ping03`
 - monitorTypeId: `MON_PNG_N` (PING 監視 / 数値監視)
 - facilityId: `Node03`
 - 監視有効: `true`

同じ一覧の中に `ANDROID_*` や `WINDIRECTOR_*` など、`facilityId = REGISTERED` の監視も多数ありますが、これは「登録ノードすべて」に共通で効くテンプレートです。これらが無効化すると**他のノードの監視まで止まる**ため、メンテ対象ノードだけ止めたい場合は外した方が安全です。



ユースケース 対処 一次的な対処を実施する (3/4)

不安定なノードはある？ある場合、メンテしたいから監視を全て止めて



3. 監視停止の方針について確認させてください

実行系（監視有効フラグ変更）は影響が大きいので、次のどちらにするか教えてください。

A. メンテ対象ノードのPING監視だけ止める（推奨・影響小）

- 無効化する監視:
 - `PingDB` (facilityId: DB01)
 - `Ping03` (facilityId: Node03)
- 影響: DB01 / Node03 の死活 (PING) 監視だけが止まります。他ノードの監視や共通監視には影響しません。

B. メンテ中は、DB01 / Node03 に関する監視を全部止めたい

- `PingDB` / `Ping03` に加えて、「REGISTERED 向けの共通監視」も止めるかどうかを決める必要があります。
- 共通監視を止める場合、**全登録ノードに対する監視が止まる可能性**があり、システム全体に影響します。

どちらの方針にしますか？

例:

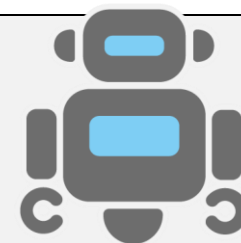
- 「AでOK。今すぐ `PingDB` と `Ping03` の監視を止めて」
- 「Bで、全ノードの監視が止まっても構わないので全部止めて」

方針と「実行してよい」という意思表示をいただければ、対象と影響内容を明示した上で、`monitorId` を指定して監視無効化（`monitorsettingsetStatusMonitor`）を実行します。

ユースケース 対処 一次的な対処を実施する (4/4)

不安定なノードはある？ある場合、メンテしたいから監視を全て止めて

AでOK



次の2つの監視を「無効化」しました (monitorFlg = false) :

1. DB01 向け
 - 監視項目ID: `PingDB`
 - 種別: 数値監視 (PING監視 / MON_PNG_N)
 - facilityId: `DB01` / scope: `DB01`
 - 更新日時: 2026-03-02 17:56:10.929 (updateUser: hinemos)
2. Node03 向け
 - 監視項目ID: `Ping03`
 - 種別: 数値監視 (PING監視 / MON_PNG_N)
 - facilityId: `Node03` / scope: `Node03`
 - 更新日時: 2026-03-02 17:56:10.940 (updateUser: hinemos)

【事実】

- 上記2件の監視設定は、Hinemos上で「監視有効 = false」に変更されています。
- これ以降、DB01 / Node03 については PING 監視によるアラートは上がりません。

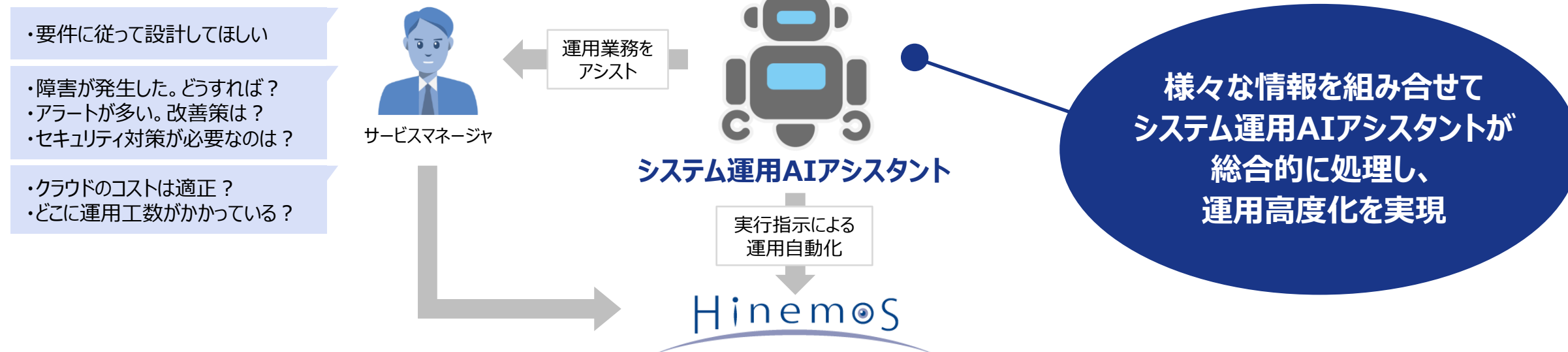
【推測される影響】

- メンテ中に DB01 / Node03 が停止・疎通不可になっても、PING監視のイベントは発生しないため、意図しないアラートは抑止できていると考えられます。
- 他ノードの監視設定 (`REGISTERED` 向け共通監視など) は変更していないため、他ノードへの影響はありません。

メンテが終わったら、同じ2つの監視 (`PingDB`, `Ping03`) を再度有効化する必要があります。
終わったタイミングで「監視を元に戻して」などと指示してもらえれば、再度有効化します。

運用現場とAIの協調 システム運用AIアシスタント

属人的・エンジニアスキルが必要・広範囲な分析が必要な情報をシステム運用AIアシスタントがカバーし運用高度化を実現



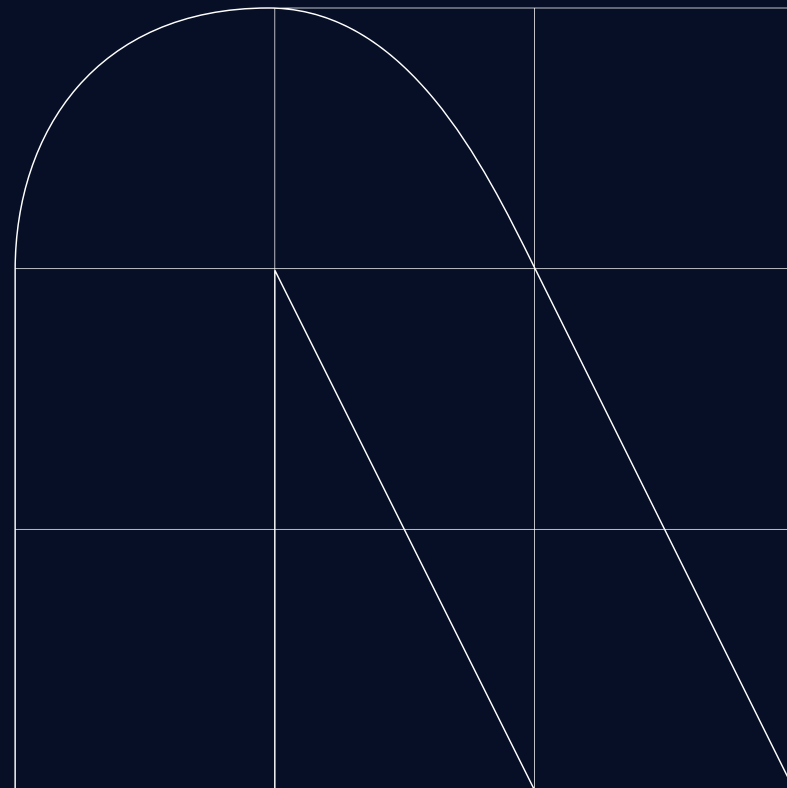
設計の迅速性、生産性を向上

判断の迅速性、正確性を向上

コストの適正管理、削減を実現

04

まとめ



まとめ

昨今のAIOpsのトレンド

昨今は生成AIを活用するAIOpsの動きが活発
Hinemosは昔から予兆検知、要因分析、オブザーバビリティなど様々なAIOpsの取り組みを推進

生成AIを使ったAIドリブン運用の世界観

人手作業をAIベースにすることで、運用の迅速性・生産性を向上

生成AIの取り組み

- **Hinemos AIエージェント**

システム運用の確認から分析、調査、対処までを自動化をサポート
通常運用から障害イベント発生時の対処を、様々なナレッジを活用してより安全に実現

- **運用現場とAIの協調**

Hinemos AIエージェントを中心にシステム運用AIアシスタントを推進
属人的・エンジニアスキルが必要・広範囲な分析が必要な情報をシステム運用AIアシスタントがカバー

AIドリブン運用に
興味をお持ちの方は
ぜひお問い合わせください！



