PagerDuty

AIでシステム運用は楽になるのか？

PagerDuty

Product Evangelist

Kazuto Kusama @jacopen

Kazuto Kusama
@jacopen

Product Evangelist	@PagerDuty
代表理事	@一般社団法人クラウドネイティブイノベーターズ協会
理事	@一般社団法人SREコネクト

AIエージェントによって激変した開発

開発がAIエージェントに取って代わられた
コーディング作業の大半を代行してくれることにより、アプリケーション開発が格段に高速化。

少数精鋭体制への転換

開発に必要な人数が減少、少数精鋭によるチーム構成に。AIに適切な指示を下せる2, 3人のチームが最適と言われる時代に

開発の民主化
知識を持つ人に限られていたアプリケーション開発が、非エンジニアまで拡大

生み出される
アプリケーションが
激増

もはや開発は
ボトルネックではない
運用が新たなボトルネックに

障害のほとんどはデプロイによって引き起こされる

~~“障害のほとんどはデプロイによって引き起こされる。~~
したがって、デプロイが増えると障害も増え、結果としてインシデント管理、軽減策、ポストモーテムが必要となる”

エレガントパズル　エンジニアのマネジメントという難問にあなたはどう立ち向かうのか より引用
https://www.amazon.co.jp/dp/4296070916

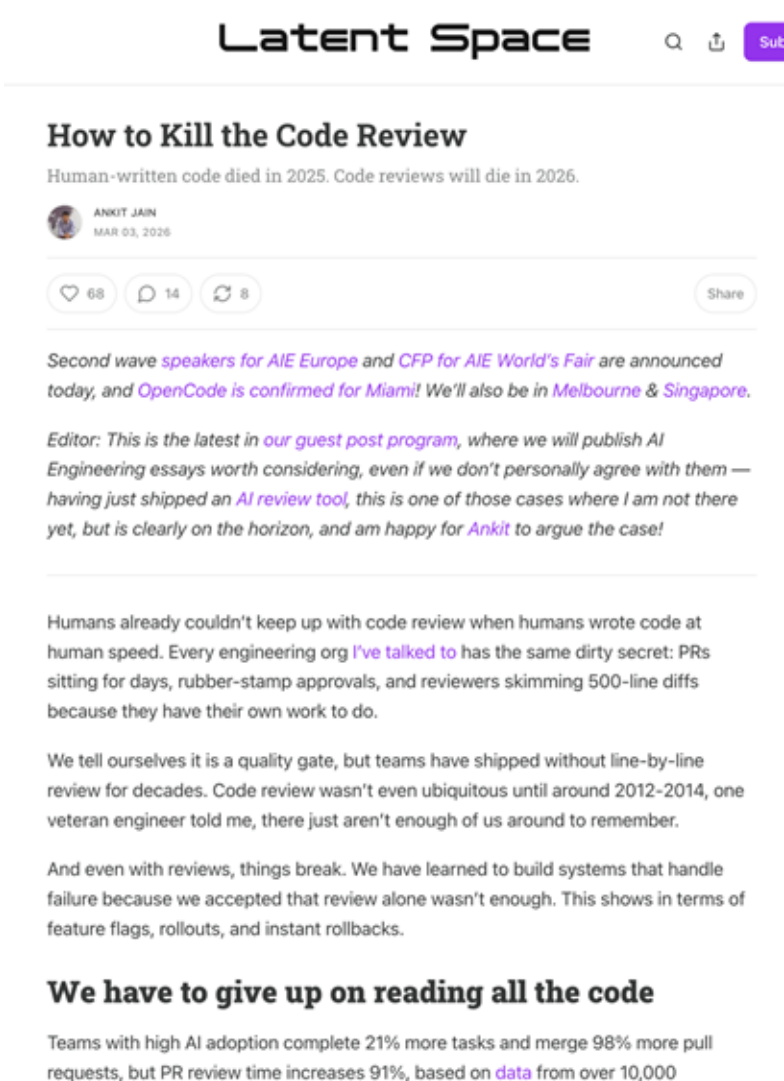
⇨ 開発が高速化するとデプロイも増え、インシデントが増える

コードレビューもなくなる時代

Human-written code died in 2025. Code reviews will die in 2026.
人間が書くコードは 2025 年に死んだ。コードレビューは 2026 年に死ぬ。

> 人間が人間の速度でコードを書いていた時代でさえ、コードレビューに追いつけていませんでした。
> この戦い（手動レビューで全部チェックする）に、私たちが勝つ道はありません。コードレビューは、もはや仕事の「形」に合っていない歴史的な承認ゲートです。

How to Kill the Code Review - https://www.latent.space/p/reviews-dead



Latent Space

How to Kill the Code Review

Human-written code died in 2025. Code reviews will die in 2026.

ANKIT JAIN
MAR 03, 2026

68 14 8 Share

Second wave speakers for AIE Europe and CFP for AIE World's Fair are announced today, and OpenCode is confirmed for Miami! We'll also be in Melbourne & Singapore.

Editor: This is the latest in our guest post program, where we will publish AI Engineering essays worth considering, even if we don't personally agree with them — having just shipped an AI review tool, this is one of those cases where I am not there yet, but is clearly on the horizon, and am happy for Ankit to argue the case!

Humans already couldn't keep up with code review when humans wrote code at human speed. Every engineering org I've talked to has the same dirty secret: PRs sitting for days, rubber-stamp approvals, and reviewers skimming 500-line diffs because they have their own work to do.

We tell ourselves it is a quality gate, but teams have shipped without line-by-line review for decades. Code review wasn't even ubiquitous until around 2012-2014, one veteran engineer told me, there just aren't enough of us around to remember.

And even with reviews, things break. We have learned to build systems that handle failure because we accepted that review alone wasn't enough. This shows in terms of feature flags, rollouts, and instant rollbacks.

We have to give up on reading all the code

Teams with high AI adoption complete 21% more tasks and merge 98% more pull requests, but PR review time increases 91%, based on data from over 10,000

AIにコードレビューさせたら障害は起きない？

全然起きる。

The future is ship fast, observe everything, revert faster.

未来は「素早く出荷し、すべてを観測し、より早く戻す」世界です。

We’re not going to outread the machines. We need to outthink them—upstream, where the decisions actually matter.
私たちは機械より多くのコードを読むことでは勝てません。

上流で、意思決定が行われる場所で、
機械よりもよく「考える」ことで勝つ必要があります。

How to Kill the Code Review - https://www.latent.space/p/reviews-dead

インシデントは必ず起きる。
起きる前提で物事を設計する

インシデント起きたとしても
AIで対処すればいいのでは？

インシデントも
AIで対処すればいい

という考えは甘え

想像してみてください

あなたは人気ECサイトのエンジニアです。

ある日、サイトが突然重くなり、監視アラートが飛びました。

「DBのCPU使用率95%超」

調査を始めると、営業から電話が入ります。

「クライアントが注文できないって言ってる。いつ直る？」

電話を切った直後、今度はカスタマーサポートからチャットが来ます

「問い合わせ急増中。お客様に何と説明すれば？」

SNSでは「またこのサイト落ちてる」と拡散が始まっています。

システムからはアラートが次々と発報され、チャットには「状況報告まだ？」が積み上がる。焦るほど、ミスしそうになります。

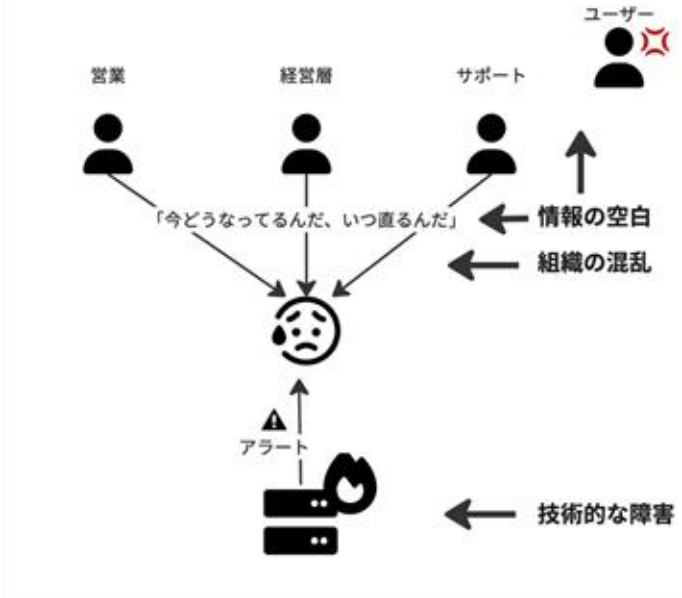
ここで起きた問題は、「システム障害」だけ？

確かに障害がトリガー、でも組織の混乱の本当の原因は

- 不十分な情報連携
- コミュニケーションの不足
- 対応ルールの欠如

エンジニアは殺到するアラートと苦情に押しつぶされ、新たなミスが起きかねない状態

障害の復旧だけでは根本解決にならず、システム障害がもたらすビジネスへの影響という問題は残ったまま。



ステークホルダーごとの「恐れ」

👔 経営層(エグゼクティブ)

必要なのは「ビジネスインパクト」

- いくらの損失が出るのか
- 法的リスクはあるのか
- メディア発表は必要か

情報がなければ、適切な判断を下せない

💼 セールス・営業

商談中の顧客から「御社のサービス大丈夫？」と言われたら･･･

たった一度の障害が、数ヶ月かけて築いた信頼関係を崩壊させ、契約を白紙に戻す可能性がある。

📣 広報・マーケティング

SNSでの炎上は秒単位で拡散する。
沈黙は「隠蔽」と受け取られかねない。

適切なタイミングでの発信が、ブランドイメージを守る鍵。

🎧 カスタマーサポート

顧客との最前線にいる人たち。システムが止まると、怒った顧客からの電話やチャットが殺到。

彼らにとって凄まじいストレスであり、離職の原因にもなり得る。

技術面だけでなく、各ステークホルダーの不安を解消しなくてはいけない

インシデントを「管理」することが重要

技術的な対応を行う障害対応だけでなく、組織やビジネスの視点を含めて問題を解決していくためのインシデント管理が、今後の世の中では必要不可欠。

障害対応

🔧 視点：　システム（技術）

🎯 目的：　壊れたものを直す（復旧）

⚡ アクション：

- サーバーの再起動
- コードのロールバック
- ログの調査
- データベースのフェイルオーバー

など。

インシデント管理

🏢 視点：　組織（ビジネス）

👥 主語：　組織全体

🎯 目的：　ビジネスへの影響を最小化し、信頼を維持する

⚡ アクション：

- 状況の指揮
- ステークホルダーへの報告
- 役割分担
- 顧客対応
- 再発防止策の策定

など。

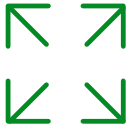
まずは基本をしっかりと

オブザーバビリティ

メトリクス、ログ、トレースが整備され、異常の検知と原因追跡が可能な状態

インシデント対応プロセス構築

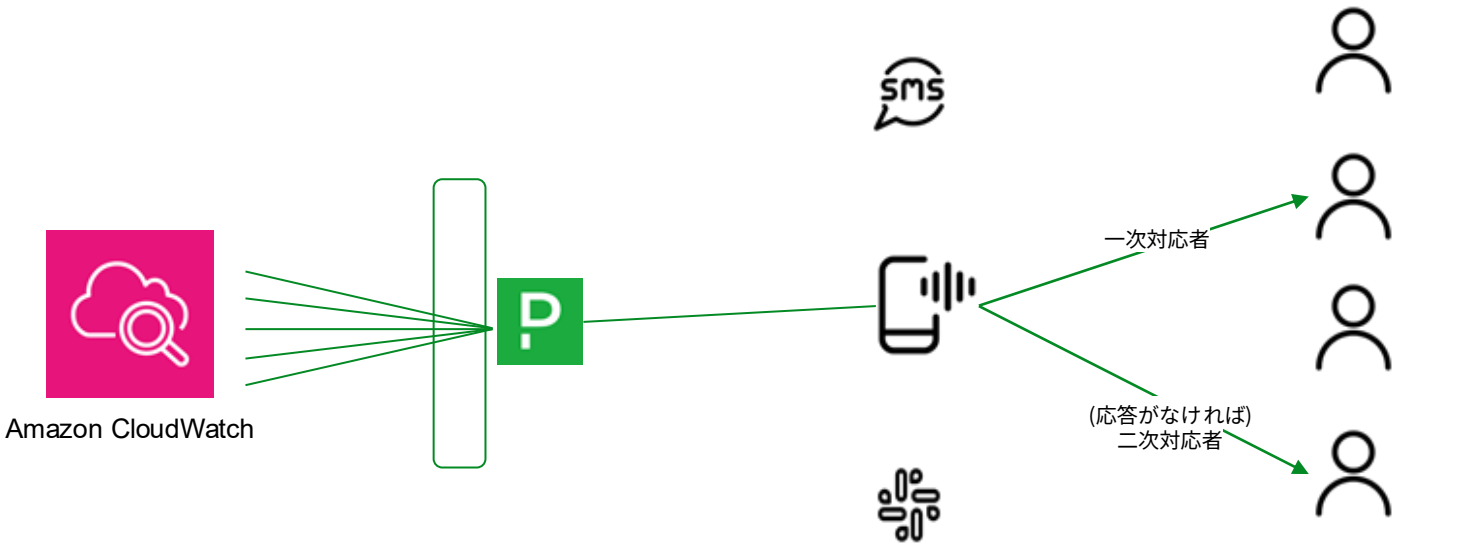
初動対応から復旧、事後分析までの標準化されたフロー



情報共有と連絡手段

コミュニケーションチャネル整備、ステークホルダーとの定期報告や広報プロトコル

インシデントが起きたときには



必要なアラートだけに絞り込み

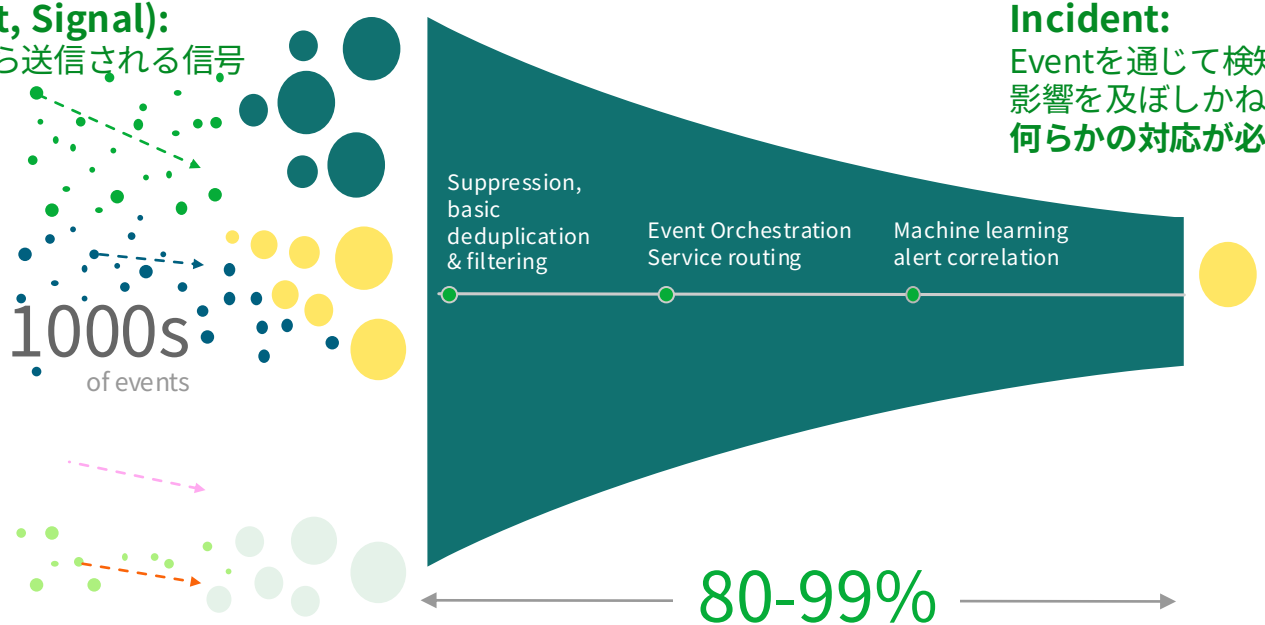
電話やSMS、プッシュ通知、
Slackなど、人それぞれ適した
通知

オンコールの
ローテーション

大量のアラートの中から、インシデントを特定する

Event (= Alert, Signal):
監視ツール等から送信される信号

Incident:
Eventを通じて検知した、サービスに影響を及ぼしかねない課題。
何らかの対応が必要なもの。



⚠ 大量のアラート

⚠ 解決に時間を要する

アラート数を約1/10に削減

迅速にインシデントを解決

PagerDuty

インシデント対応プロセスの確立

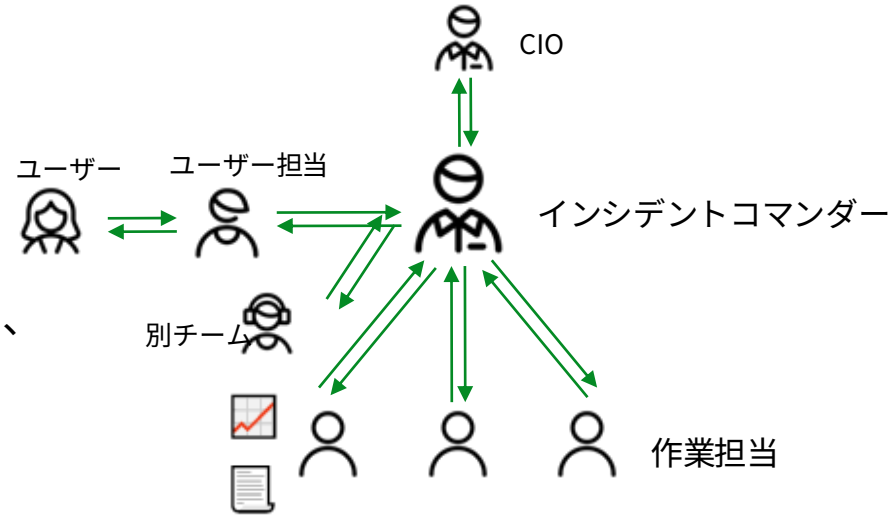
責任の所在と
意思決定の経路を明確にし
情報の流れをコントロールする

インシデントコマンダー(IC)を
中心とした命令指揮系統を構築

ICはインシデント対応の指揮者。
重大インシデントを解決に導くことを目的とし、
意思決定を行う。

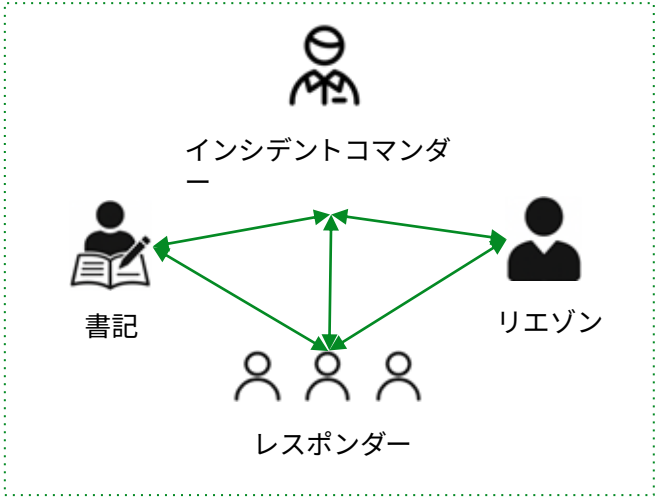
このフローがない場合

アラート検知 → ？？？
AIは「次に何をすべきか」を提案できず、
各人がバラバラに動く

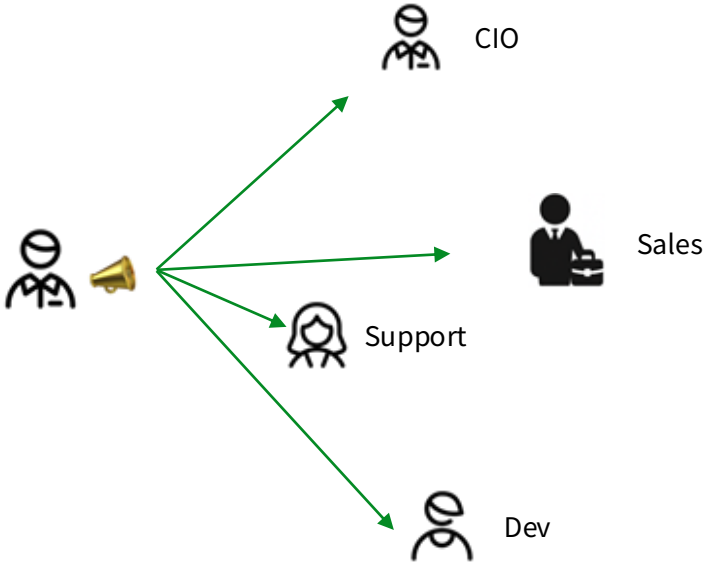


適切なコミュニケーション

インシデント対応チーム内
ICを中心とした、意思決定と指揮を
行う密な連携



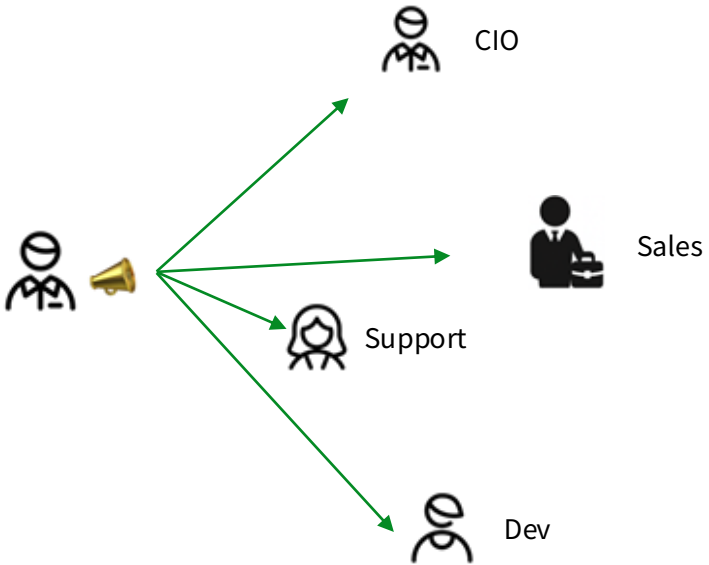
ステークホルダーとのコミュニケーション



ステークホルダーとのコミュニケーション

経営的な意思決定、対外的な情報発信のために重要。3つの「適切」を意識

- 適切な粒度
  - 技術的な詳細までは不要
  - 何が起きているか、今何をしているか、今後の見通し　を伝える
- 適切な方法
  - ブロードキャスト型を徹底
  - 戦時における1:1のコミュニケーションは、取り返しの付かない遅延を招く
- 適切なタイミング
  - 定期的 (1時間おき、など)
  - ステータスに変化があったとき



AIエージェント時代だからこそ、振り返りが必要

起きてしまったインシデントは、きちんと分析して改善に繋げる

AIエージェントによる問題？人為的なミス？ インフラの障害？

障害によって対処すべき方法が異なる。きちんと振り返り「インシデントが繰り返し起きないよう対処」することが重要

AIじゃない部分の
インシデント管理

AIによる
インシデント管理

PagerDuty AI エージェント
- インサイトからアクションまで -

エージェントがより良く、早く、スマートに業務を支援

PagerDuty AI エージェントに含まれるエージェント一覧

Insight エージェント

組織内で使われているツール全体のデータを分析し、戦略的な運用判断に必要な情報を特定し、運用手順とビジネスの効率を継続的に改善

インシデント
対応プロセスの改善

Shift エージェント

オンコールシフトを動的に調整して、スケジュールや空き時間の競合を未然に防ぎ、インシデント担当者のカバレッジを確保することで、迅速なインシデント解決を促進し、運用コストの削減と顧客影響の最小化を図る

On-call 対応
スケジュールの調整

SRE エージェント

運用上の問題を特定して分類し、関連する過去のインシデントなどの重要なコンテキストを浮き彫りにし、対応者に解決を早めるための推奨事項を提示することにより、業務の中断によって引き起こされるビジネスリスクを軽減し、顧客体験を向上

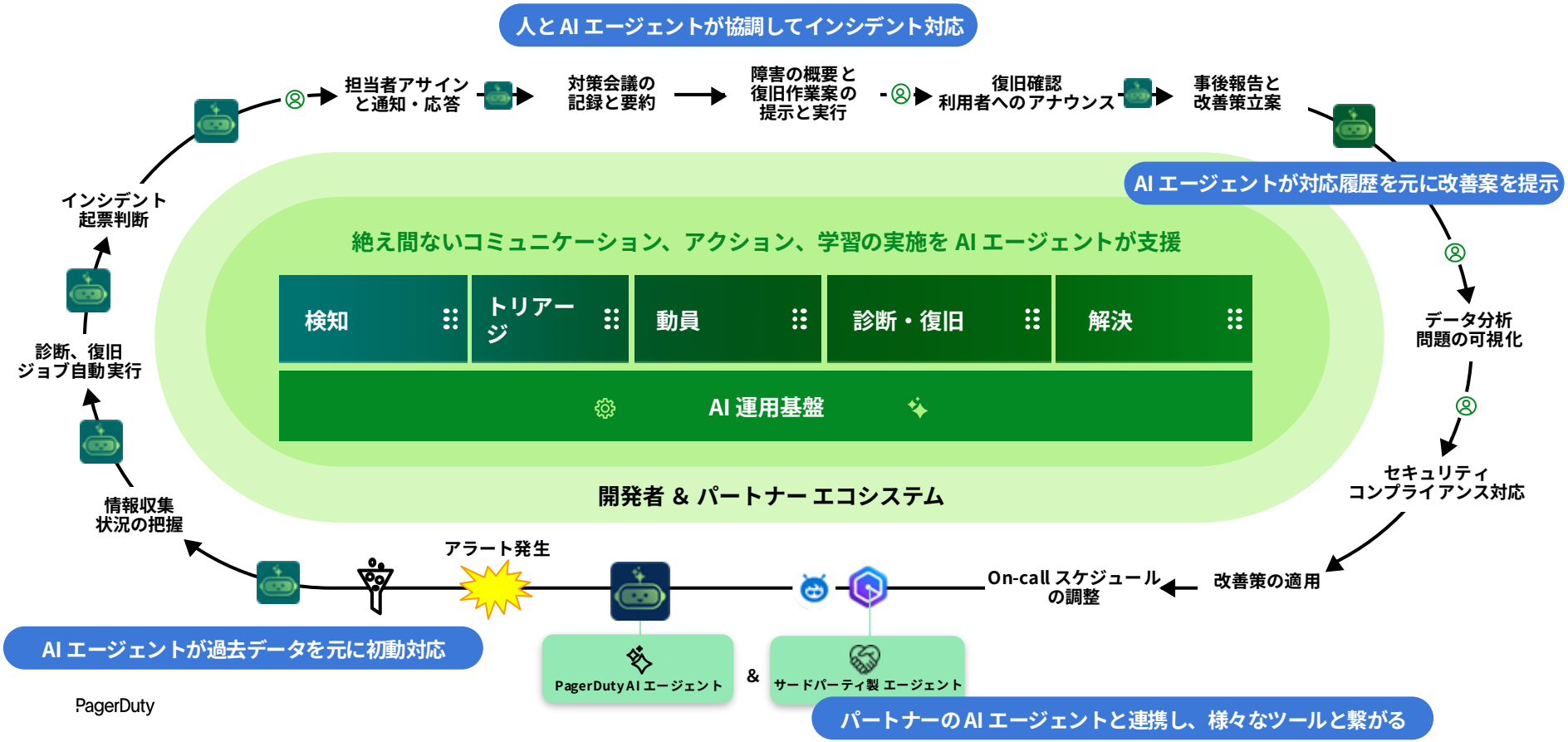
インシデント
対応中の右腕役

Scribe エージェント

Web 会議での会話内容をリアルタイムに整理、分析し、必要なアクションを特定し、内容をサマリーして提供することにより、インシデント解決の迅速化と関係者への情報共有を促進

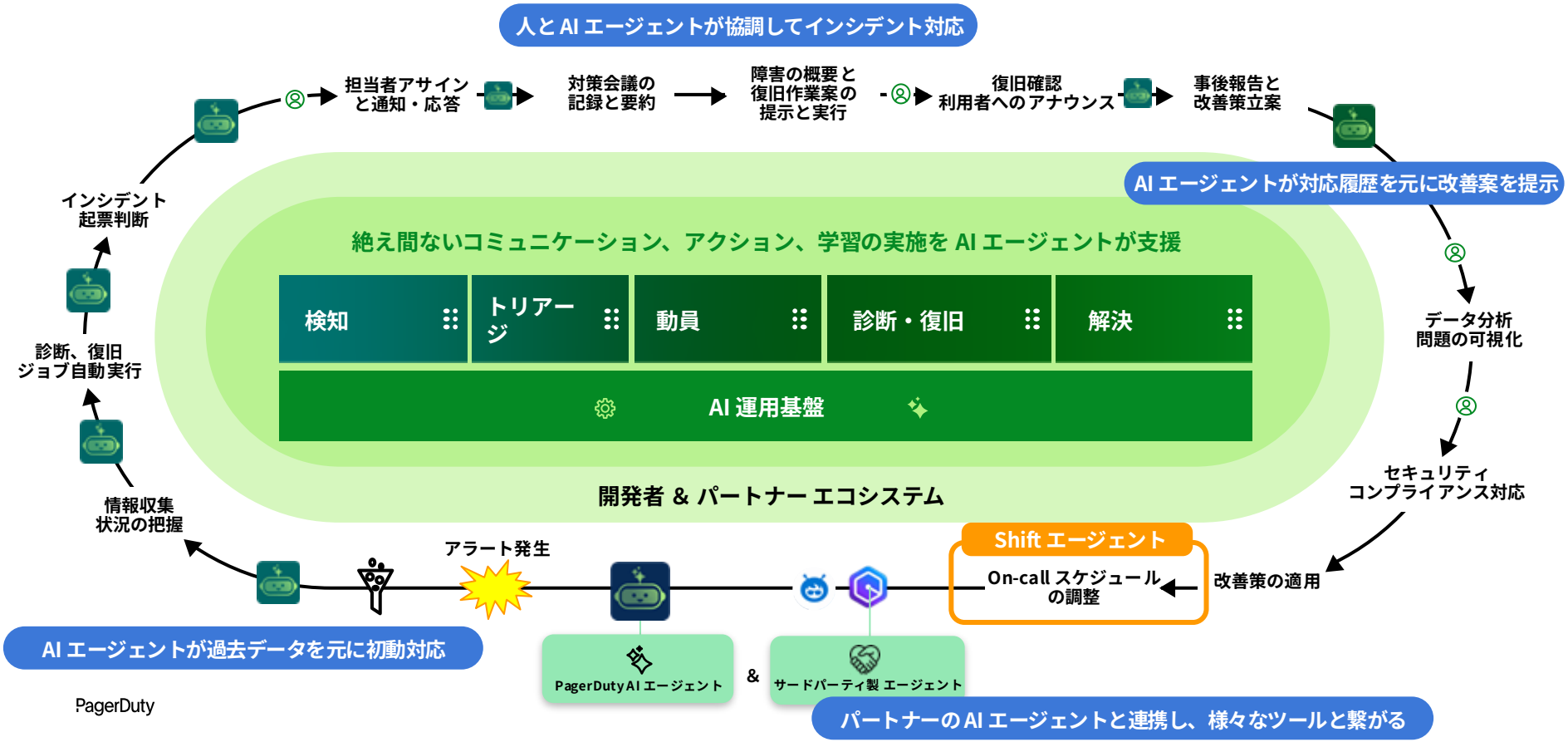
インシデント対応の
筆記担当者

PagerDuty AI エージェントを活用したシステム運用のライフサイクル



PagerDuty

PagerDuty AI エージェントを活用したシステム運用のライフサイクル

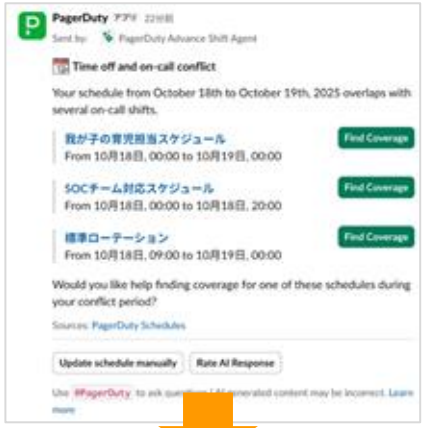


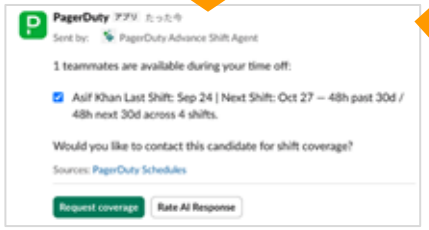
PagerDuty

Shift エージェントができること

On-call スケジュール、エスカレーションポリシーに関する問い合わせ対応
Google カレンダーや Outlook 予定表※で不在が検出される or 不在の予定をエージェントに伝えると、
交代メンバー候補の提示と打診、Override スケジュールの登録を実施

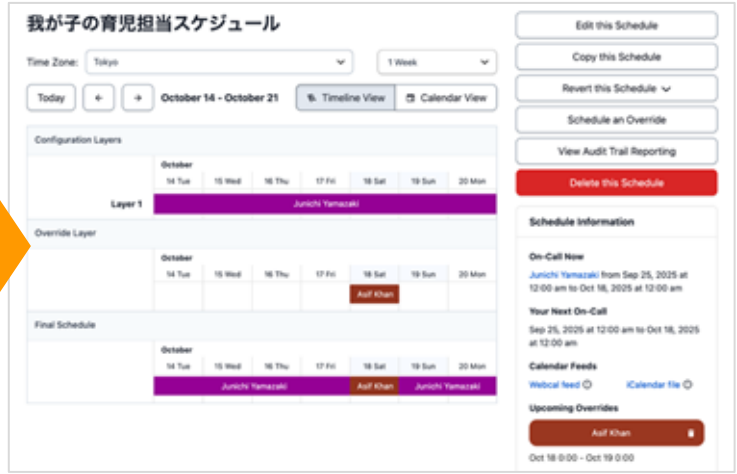
個人予定と On-call 予定との競合の通知





On-call 対応の交代候補者の提示

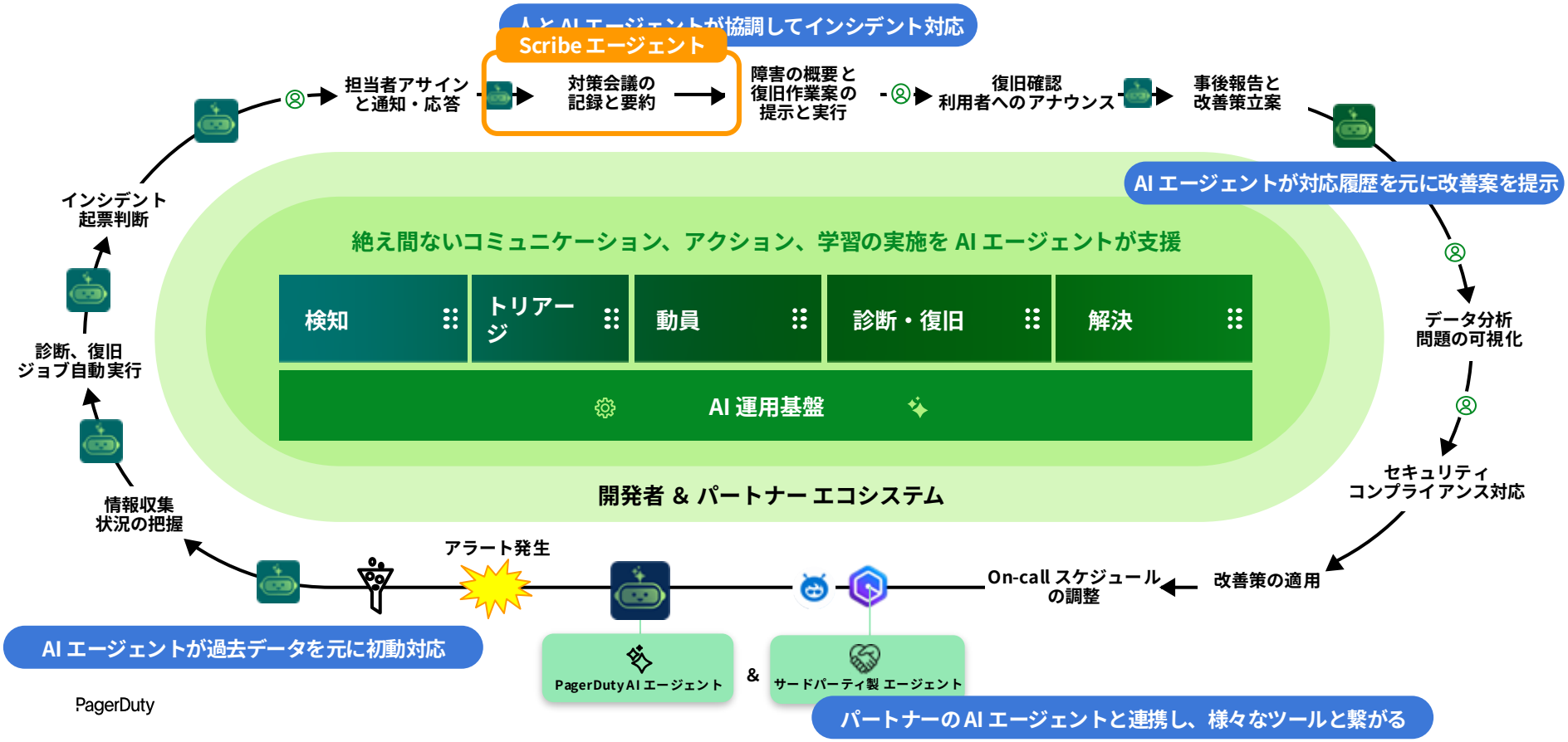
候補者へ交代の打診



On-call スケジュールへの反映

※ Googleカレンダーは対応済み。Outlook予定表は2026年夏頃までの対応を予定

PagerDuty AI エージェントを活用したシステム運用のライフサイクル

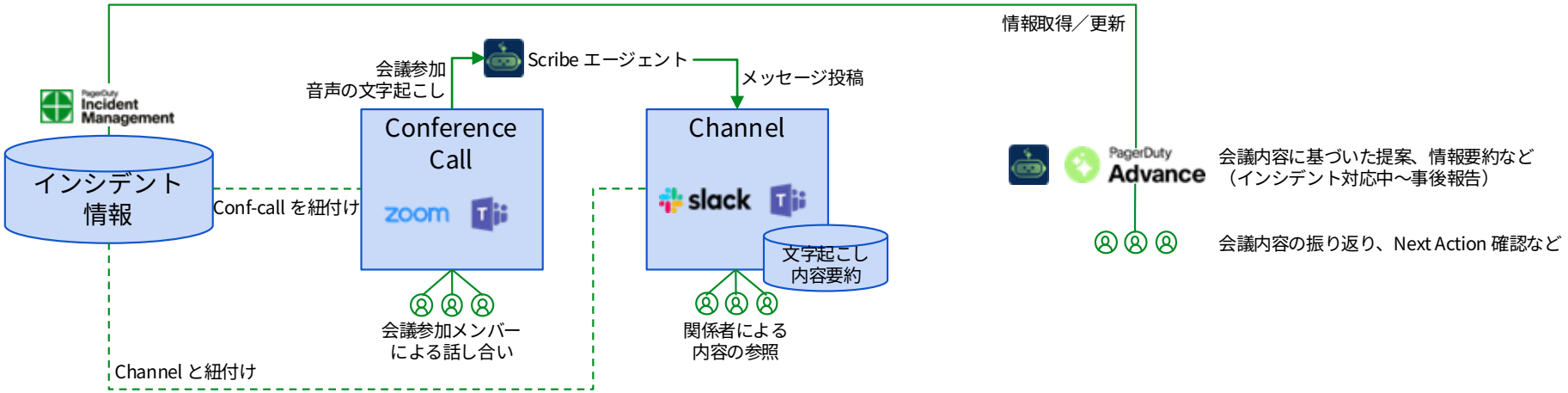


PagerDuty

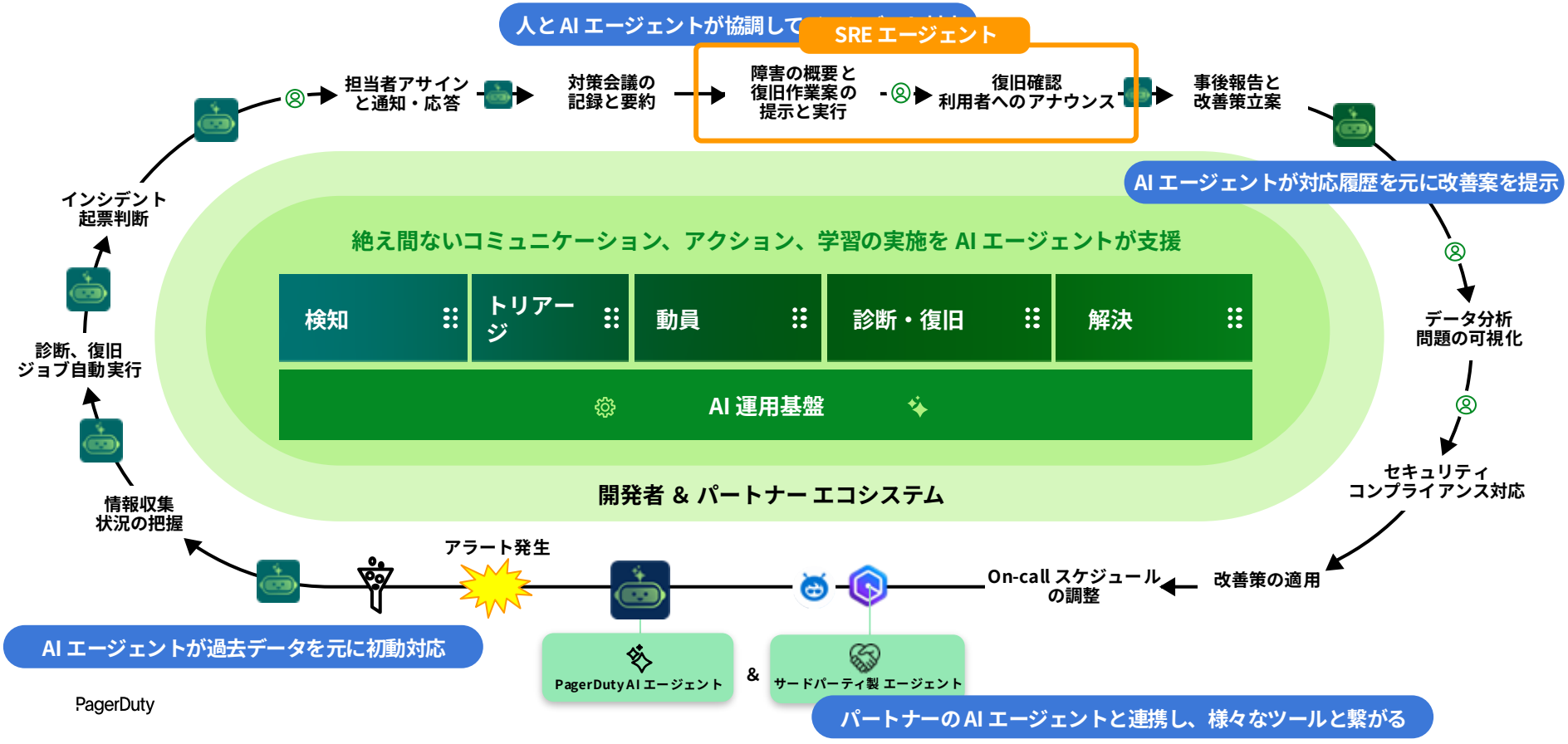
Scribe エージェントができること

インシデントと紐付けられたWeb 会議のライブ文字起こし
会議終了後の内容要約と Next Action の提示

PagerDuty 上のインシデント情報と紐付け、以降の対応で
人やPagerDutyが参照し担当者を支援



PagerDuty AI エージェントを活用したシステム運用のライフサイクル



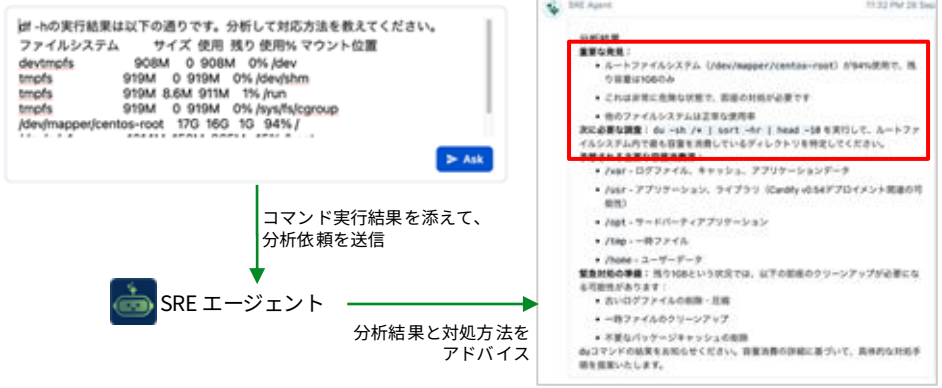
PagerDuty

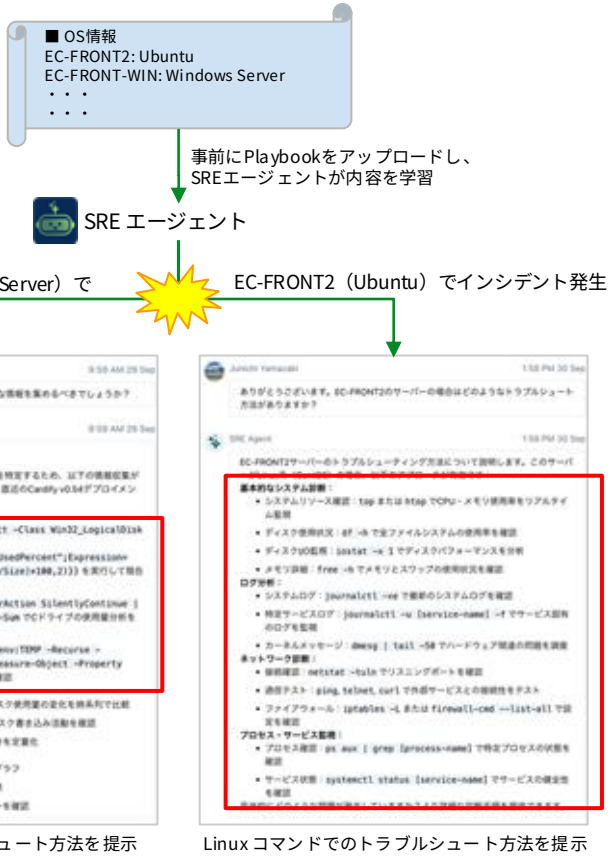
SRE エージェントができること

インシデントの発生～解決までのフェーズで支援する AI アシスタント。**担当者がインシデントを迅速に診断、トラブルシューティング、修復できるよう**設計。

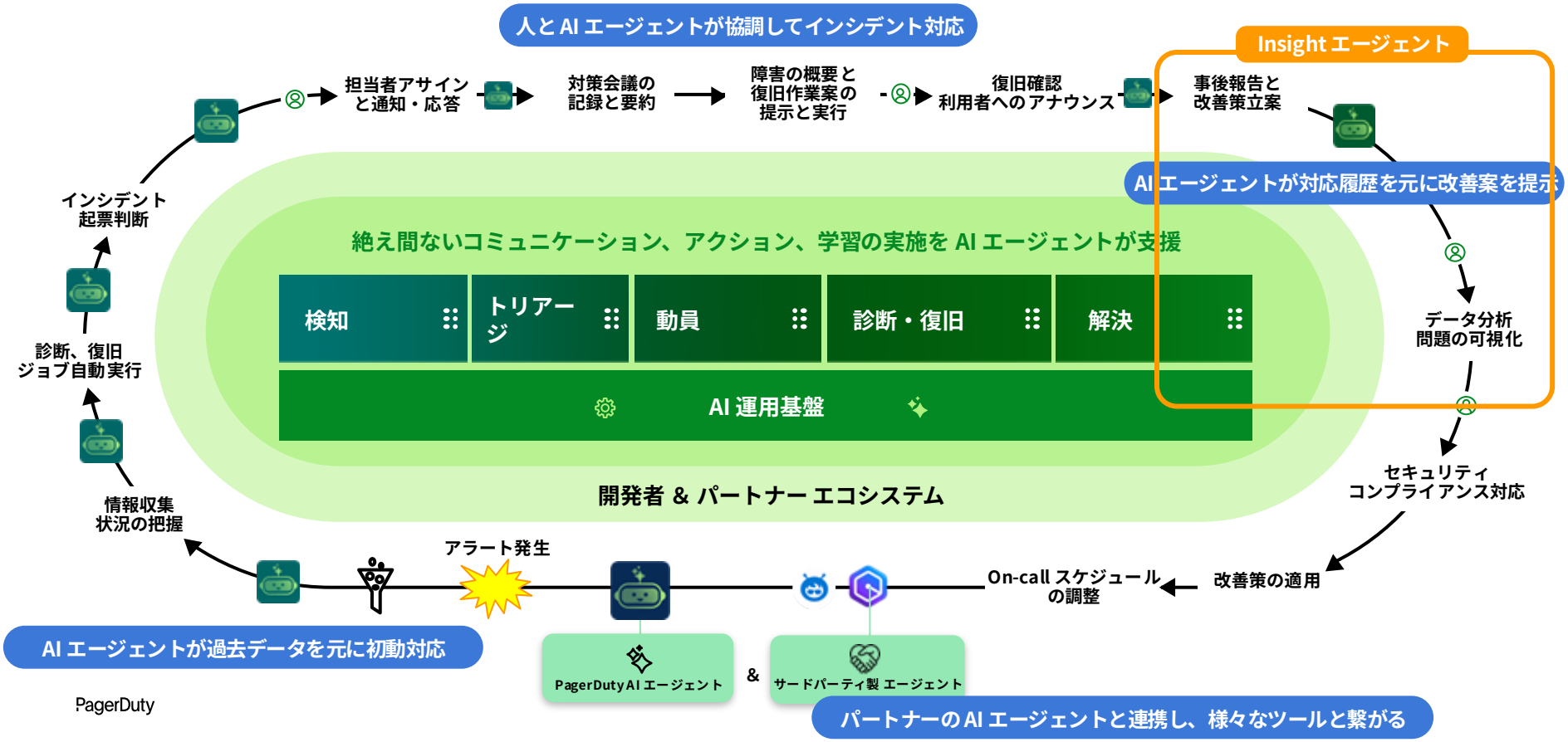
主なメリット

- インシデントのトリアージを加速
- 重大なインシデント発生時の対応負荷を軽減
- 継続的な学習を通じて問題の再発を防止
- 過去の修復パスとユーザーフィードバックを元に改善





PagerDuty AI エージェントを活用した システム運用のライフサイクル



PagerDuty

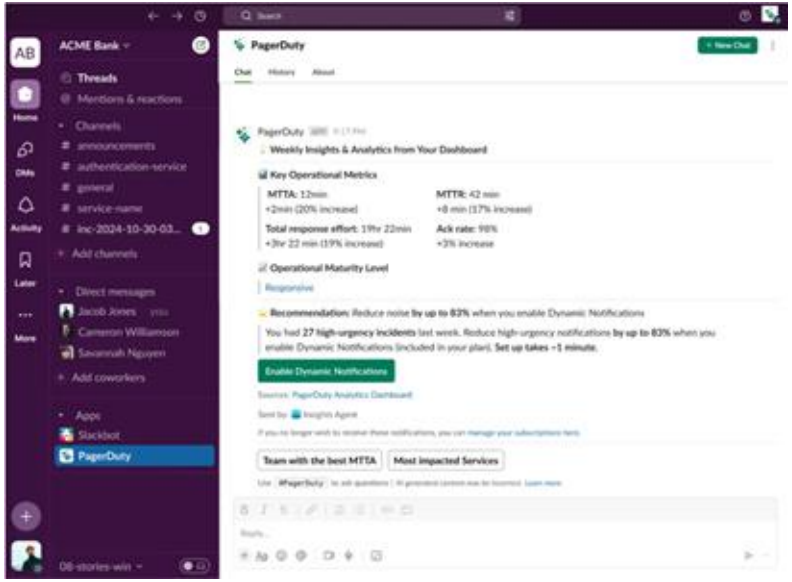
Insight エージェントができること

PagerDuty の定型レポートだけでは読み取れない、独自の分析軸／条件でのデータ提示
定期的に PagerDuty のデータと設定をスキャンし、有効化することでより便利に使える機能を提案

特定 Team のインシデント解決時間の平均値を提示

Q2 と Q3 の深夜時間帯のインシデント対応件数の推移を提示
（「深夜時間帯」の定義も回答）

定期（Weekly）レポートと機能有効化の提案



Thank you

PagerDuty