ジョブ管理の技 20

ジョブ異常時の障害連絡を自動化し迅速な障害復旧を行いたい

現状の課題

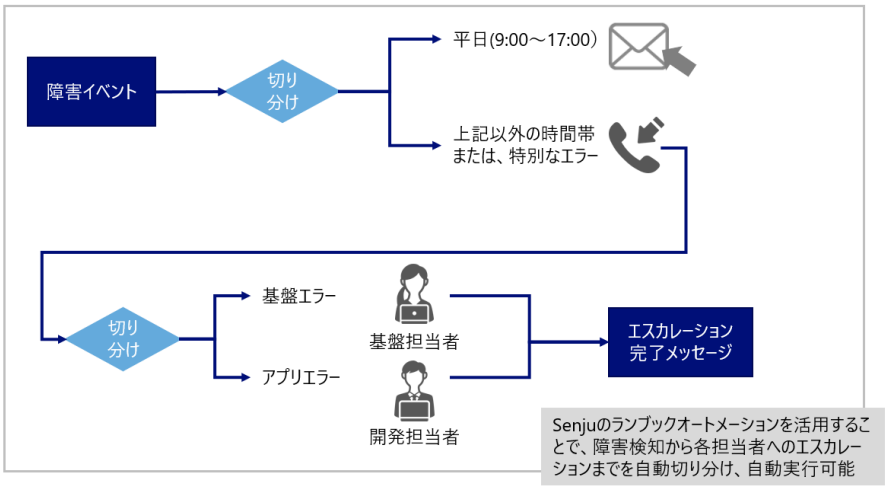
ジョブの異常をゼロにするのは現実的ではない。しかし、ジョブ異常時のビジネスへの影響を最小限に抑える必要がある。そのためには、ジョブが異常になることを前提にし、迅速に対応できる対策が必要である。迅速な対応を実現するにはジョブの状態を常時監視し、いかに素早くエスカレーションするかがポイントになる。障害が発生すると、オペレーションエンジニアは開発・基盤担当者やシステム利用者などにメールや電話などで異常を通知する。このとき、システムや時間帯によって連絡先が異なったり、相手に応じて内容を変えて連絡したりしているケースが多い。

そこで問題となるのが、「時間のロス」と「コミュニケーションミス」である。電話では不在時に連絡が取れず、メールでは読まれているかが確認できない。口頭での連絡では詳細が伝わらず、メールでの連絡では背景が伝わらない。そのため、念押しの連絡や差し戻しなどの無駄な作業が発生してしまう。

解決策

障害連絡を自動化することにより、時間ロスを無くし、連絡漏れや報告内容の間違いも無くなる。さらに、相手が確認したことまでシステム化できれば人による確認連絡も必要なくなる。単純に全ての障害イベントを自動的にエスカレーションしてしまうと、受け側の負荷が増し本末転倒である。イベントを見極め、必要な情報を的確に伝える必要がある。

さらに、連絡方法も工夫が必要である。メールでの連絡では、相手が見ているか確認が取れない。メールに加え、自動電話による連絡も有効である。担当者不在の場合の 2 次対応の方法も決めておく必要がある。



Senju Family での実践方法

Senju DevOperation Conductor では「自動メール送信」機能を実装している。あらかじめ登録されているメールテンプレートに従って、自動でメール送信を行う。テンプレートには環境変数を指定することができ、メール送信時に自動的に環境変数を組み立てて送信される。メール送信を行うきっかけになったイベント情報（イベント ID、日時、サーバー、内容、ジョブ名など）を環境変数として指定すれば、発生したイベントに応じて異なる内容のメールを柔軟に送信することが可能である。

また、「自動電話発信」も実装している。あらかじめ登録されている連絡先グループに、自動で電話コールを行う。コール先について、ジョブの終了コードで切り分けを行うことで基盤エラーであれば基盤担当者へ、アプリエラーであれば開発担当者へ連絡を行うことが可能である。また、連絡先グループには複数の電話番号を登録でき、誰かが電話に出るまで、繰り返し電話を掛け続けることができる。