JANOG39
Meeting

RICOH
imagine. change.

障害ありきで運用自動化をやってみた

2017年1月18日

株式会社リコー　後藤芳和

きっかけ

RICOH
imagine. change.

- JANOG38 で、IIJ松崎氏が「色々なトラフィック制御の利点と欠点」というお題で発表をしてくださいました
  - 「国内だと切り替わりに 270 秒」
- 「え？そんなに？」と
  - こうなったら障害対応も自動化だ！
- やってみた
  - 障害への対策と自動化、独自実装も
  - ワールドワイドにサービスしているからこそできたこと
  - 未だ残る悩み

自己紹介

RICOH
imagine. change.

- 経歴
  - 計測機器メーカー
    - L3 スイッチのファームウェア開発
    - ネットワークアナライザの IPv6 プロトコルスタック開発
    - 2.5GHz 帯移動通信の仕様策定と実験機のファームウェア開発
  - ISP
    - お客様向けサーバーとネットワーク機器のお守り
  - 現在：リコー
- 個人的にやっていること
  - Twitter：@goto_ipv6
  - http://togetter.com/id/goto_ipv6
  - http://www.slideshare.net/goto_ipv6

サービス概要

RICOH UCS

- 正式名称は「RICOH Unified Communication System」
  - 以下、本発表では「RICOH UCS」とさせてください
- どんなサービス？
  - ビデオ会議
- どの地域に提供？
  - ワールドワイドに
- 弊社の特徴
  - ハードウェアもソフトウェアも作ることができる
    - 専用端末
    - 各種 OS 用の会議アプリケーション
      - Windows, macOS, iOS, Android
    - SDK

ユーザー識別子

RICOH
imagine. change.

- CID(Contact ID)
  - 電話番号みたいなもの
  - 国番号や市外局番のような、どの「制御ドメイン」に属するかを示す桁も
    - JP制御ドメイン（アジア、オセアニアなど）
    - US制御ドメイン（米国、欧州、中東、アフリカなど）

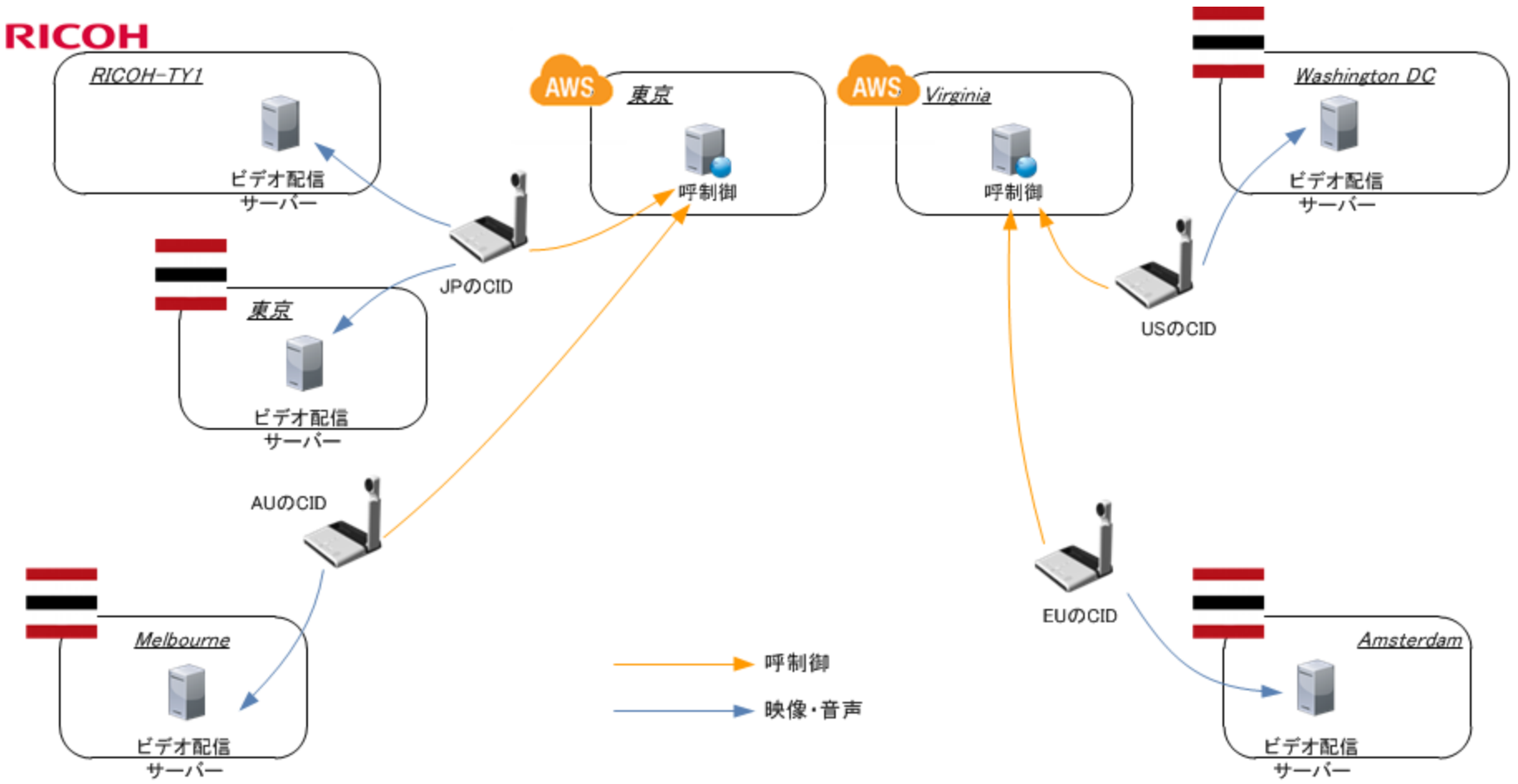
ビデオ配信サーバー

RICOH
imagine. change.

- クライアント同士が直接、音声や映像を送り合うことはしない
  - 多拠点会議も可能なので
- 映像・音声を配信する機能を持つ
  - 複製、転送処理があるので CPU 負荷がそれなりに高い
    - 物理サーバー（ベアメタル）の方が良い
- 品質に影響するので、お客様に近いところに配置したい
  - 遅延をなるべく少なく
    - あっちこっちに設置
- 「制御ドメイン」の中で「系統」わけ
  - 冗長化やメンテナンス性を考慮して
  - 管理サーバーが複数台のビデオ配信サーバーを管理

システム配置

RICOH
imagine. change.



ビデオ会議までの流れ

RICOH
imagine. change.

- ログイン
  - 専用端末はクライアント証明書で
- 会議相手一覧の取得
- 会議開始
  - 呼制御（接続先を教えてもらう）
  - 接続先（ビデオ配信サーバー）へ接続
- 会議終了
- ログアウト

想定される事態(1)

RICOH
imagine. change.

- 小規模な障害
  - デーモンの再起動や OS の再起動を自動化したり
    - Zabbix のリモートコマンド実行機能を利用
  - サーバーを冗長構成にしたり自動再構築されるようにしたり
    - AWS では CloudFormation や Auto Scaling, ECS など
      - Kumogata を利用し、CloudFormation テンプレートを Ruby DSL にして処理をひとまとめに
      - 開発環境用と本番環境用を切り替えて構築するための起動コマンド (CLI)
      - 設定情報は YAML で記述し、必要に応じて S3 へ転送
      - Docker を積極的に使用
      - Jenkins で自動テストや時間指定した自動構築
    - SoftLayer では API を叩いて一括サーバー発注など
  - サーバーを設置する DC/ゾーンを分けたり
    - Availability Zone など

想定される事態(2)

RICOH
imagine. change.

- 大規模な障害
  - Region（地域）単位
    - 本資料における「系統」が複数含まれる
      - JP制御ドメインにおけるJP/AU向け、US制御ドメインにおけるUS/EU向け
    - Priority による接続先の切り替えなど（後述）
  - クラウド事業者単位
    - 急激な値上げ！？
    - 事業者が倒産！？

システム監視について

これまでの監視

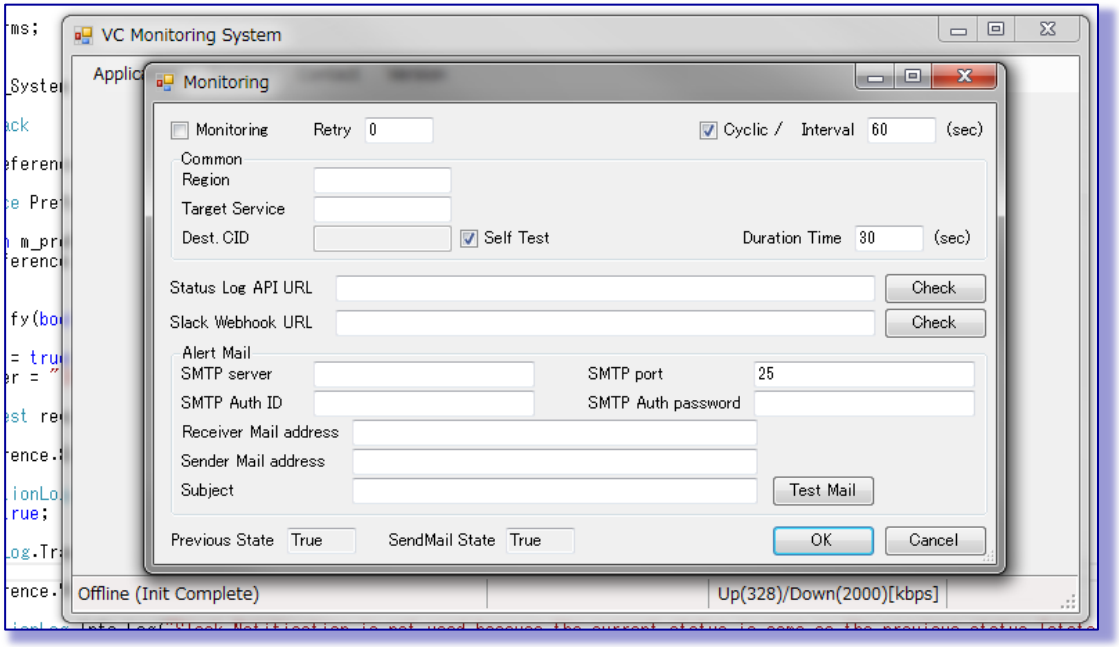
RICOH
imagine. change.

- リソース監視が主体
  - Zabbix を利用
  - 日々の状態を記録しておく
  - 大規模障害の際は、アラートメールが大量に送信される問題も
- 異なる DC を相互監視
  - 特に重要な機能についてのみ
  - Zabbix の Web 監視など

ビデオ会議確認クライアントを開発

RICOH
imagine. change.

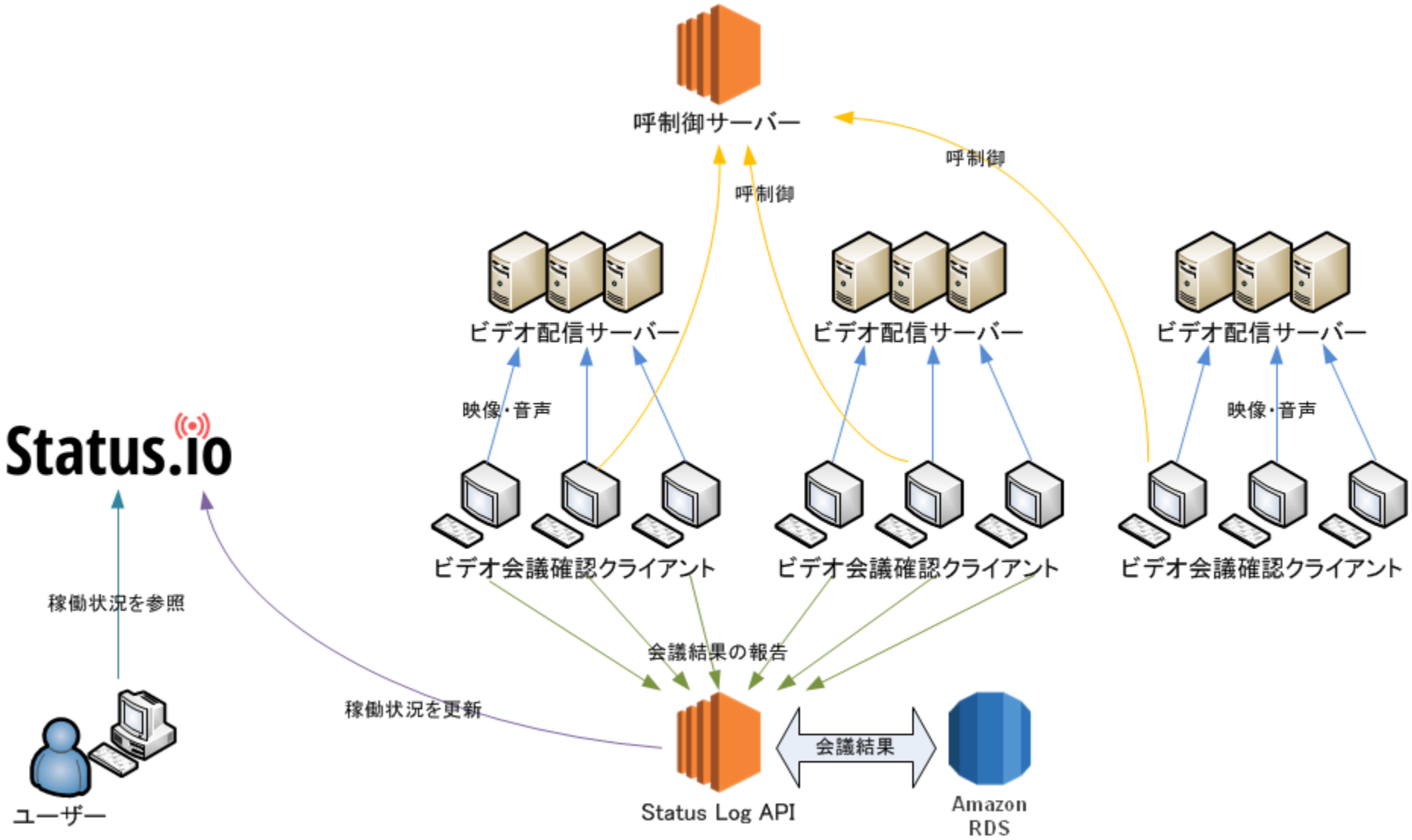
- ビデオ会議レベルでの監視をしたいという要望
- SDK を用いて
  - .NET Framework (C#)
  - 各地に用意した Windows Server VM上で周期的に動作
  - アラートメール送信機能
  - Slack 通知機能
  - CID で接続先を限定



Web によるステータス表示

- お客様に、RICOH UCS の現状をお知らせしたいという要望
- 表示には status.io というサービスを利用
  - http://status.ja.ucs.ricoh.com/ (日本語)
  - http://status.en.ucs.ricoh.com/ (英語)
- 各地に設置したビデオ会議確認クライアントからの情報を集計するソフトウェアを開発
  - 名前は「Status Log API」
  - Ruby/Sinatra
  - 入出力両方の API を提供

全体図


呼制御サーバー
呼制御
呼制御
ビデオ配信サーバー
ビデオ配信サーバー
ビデオ配信サーバー
映像・音声
映像・音声
Status.io
ビデオ会議確認クライアント
ビデオ会議確認クライアント
ビデオ会議確認クライアント
稼働状況を参照
会議結果の報告
稼働状況を更新
会議結果
ユーザー
Status Log API
Amazon
RDS

障害発生時の自動切り替えについて

呼制御に頼っていた自動切り替え

RICOH
imagine. change.

- CID には「Priority」という属性がある
  - 呼制御の際に使用される
    - どのビデオ配信サーバーの「系統」に優先的に接続させるか
    - 例）US制御ドメイン内の EU「系統」向け CID の場合
      - Priority 100→EU, Priority 90→US
      - 通常時は EU へ接続させるといった制御
- クライアントは、指示されたビデオ配信サーバーへ接続しに行く

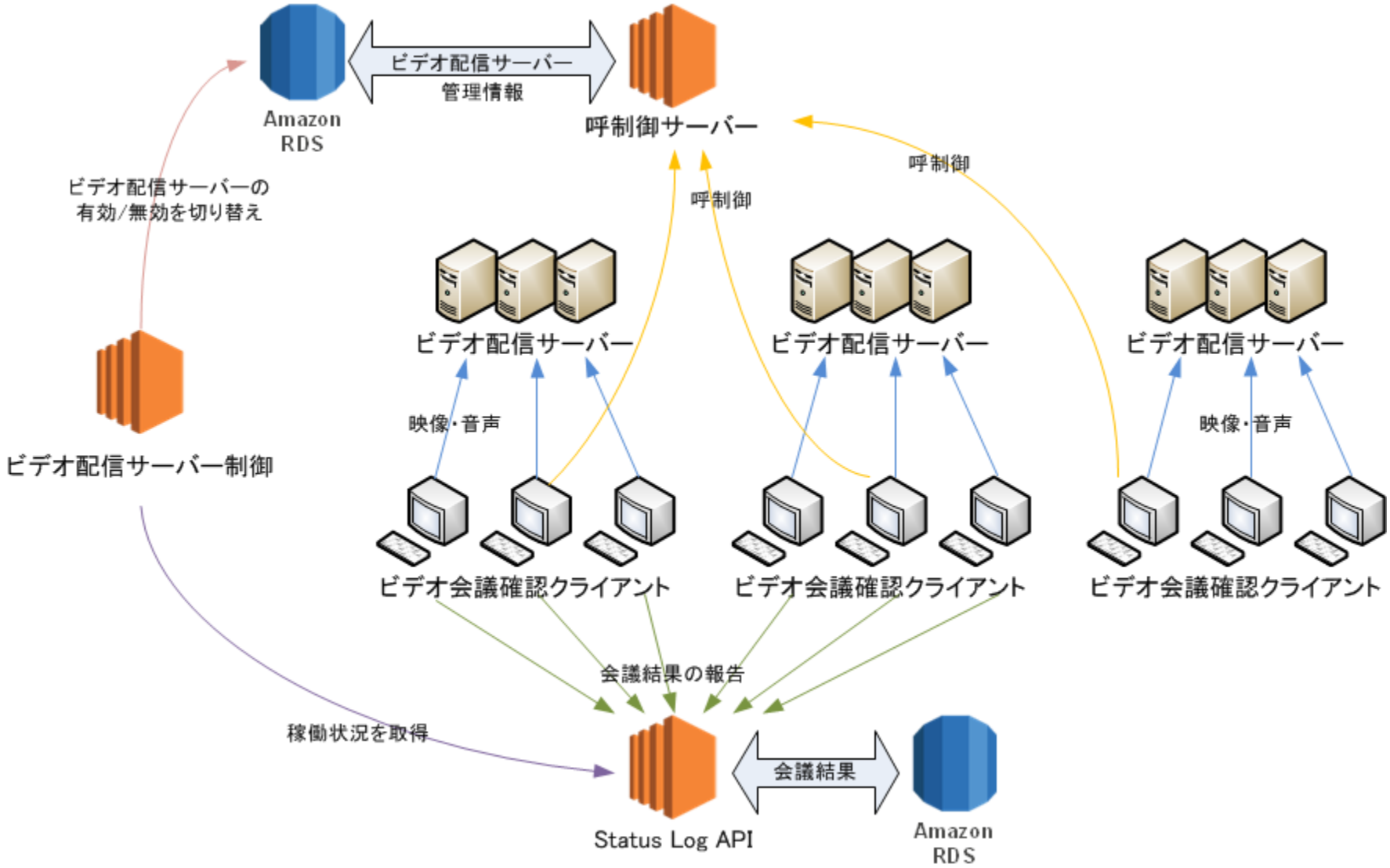
- しかし
  - クライアントは、そのビデオ配信サーバーが「正常稼働中かどうか」を、ビデオ会議前に知る術はない
  - 呼制御サーバーも、ビデオ配信サーバーの稼働状況を知らない

状況に応じた切り離し及び再投入

RICOH
imagine. change.

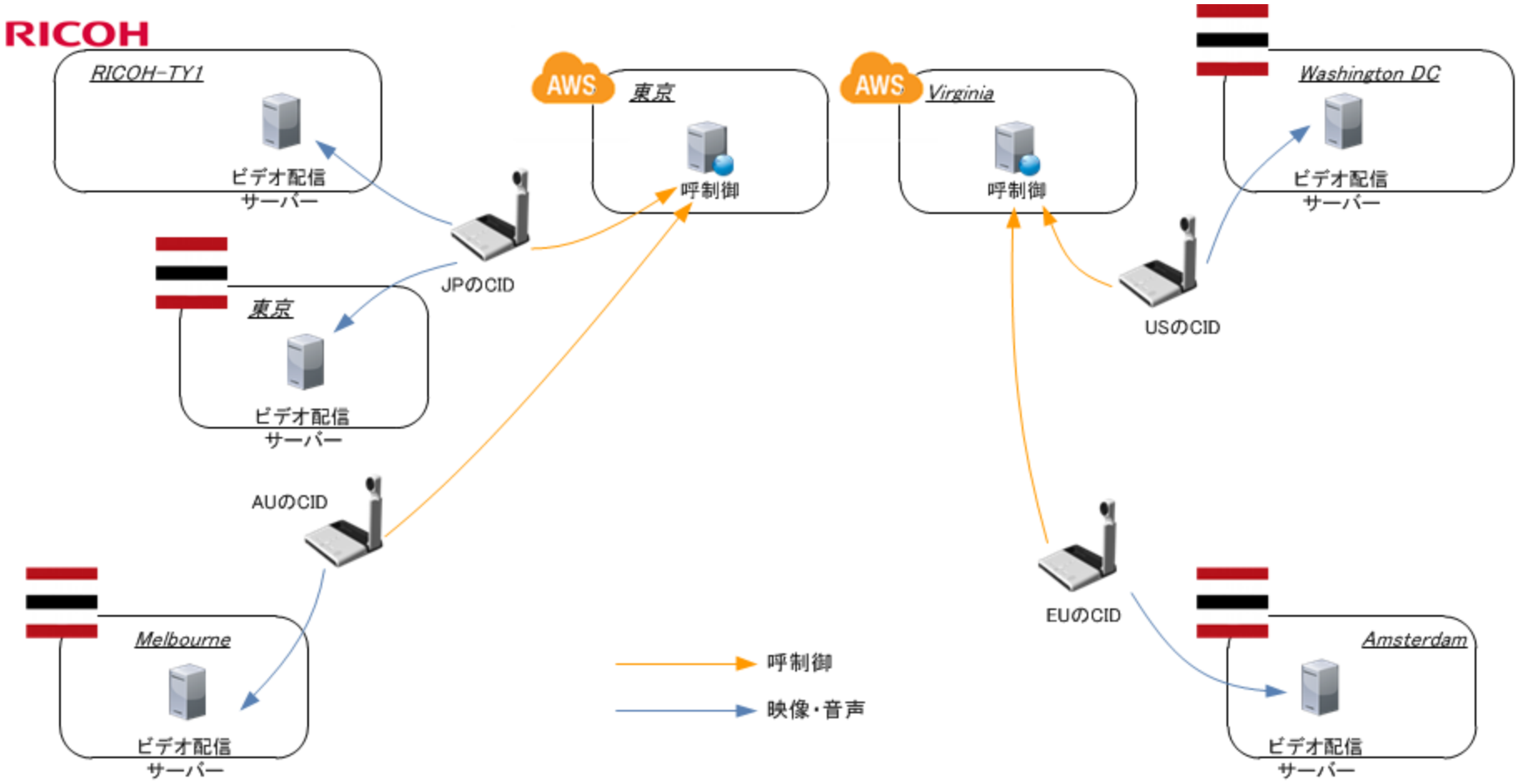
- クライアントにリトライさせたくない、待たせたくないという要望
- ビデオ会議監視システムを利用して稼働状況を取得
  - システム監視用に開発したものを流用
  - 稼働状況は Status Log API から取得
- ある「系統」のビデオ配信サーバー（群）が停止した場合に
  - 呼制御の際に、その「系統」をクライアントに教えないようにする
- 停止から復旧した場合に
  - 呼制御の際に、その「系統」もクライアントに教えるようにする
    - ビデオ配信サーバー単位での切り離し/投入も可能

全体図


ビデオ配信サーバー
管理情報
Amazon
RDS
呼制御サーバー
ビデオ配信サーバーの
有効/無効を切り替え
呼制御
呼制御
ビデオ配信サーバー
ビデオ配信サーバー
ビデオ配信サーバー
映像・音声
映像・音声
ビデオ配信サーバー制御
ビデオ会議確認クライアント
ビデオ会議確認クライアント
ビデオ会議確認クライアント
会議結果の報告
稼働状況を取得
会議結果
Status Log API
Amazon
RDS

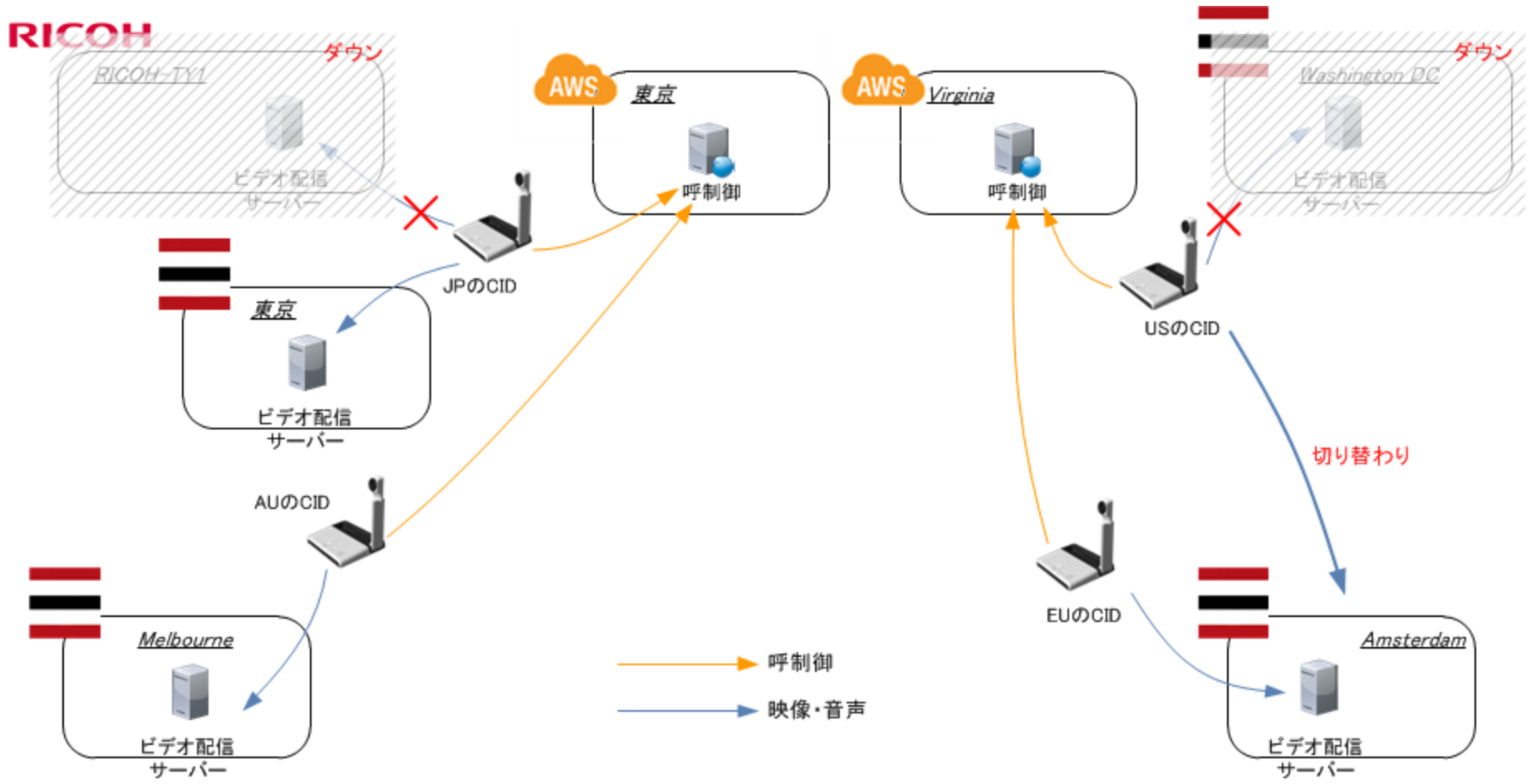
通常時

RICOH
imagine. change.



障害時

RICOH
imagine. change.



メールによるお客様通知の自動化

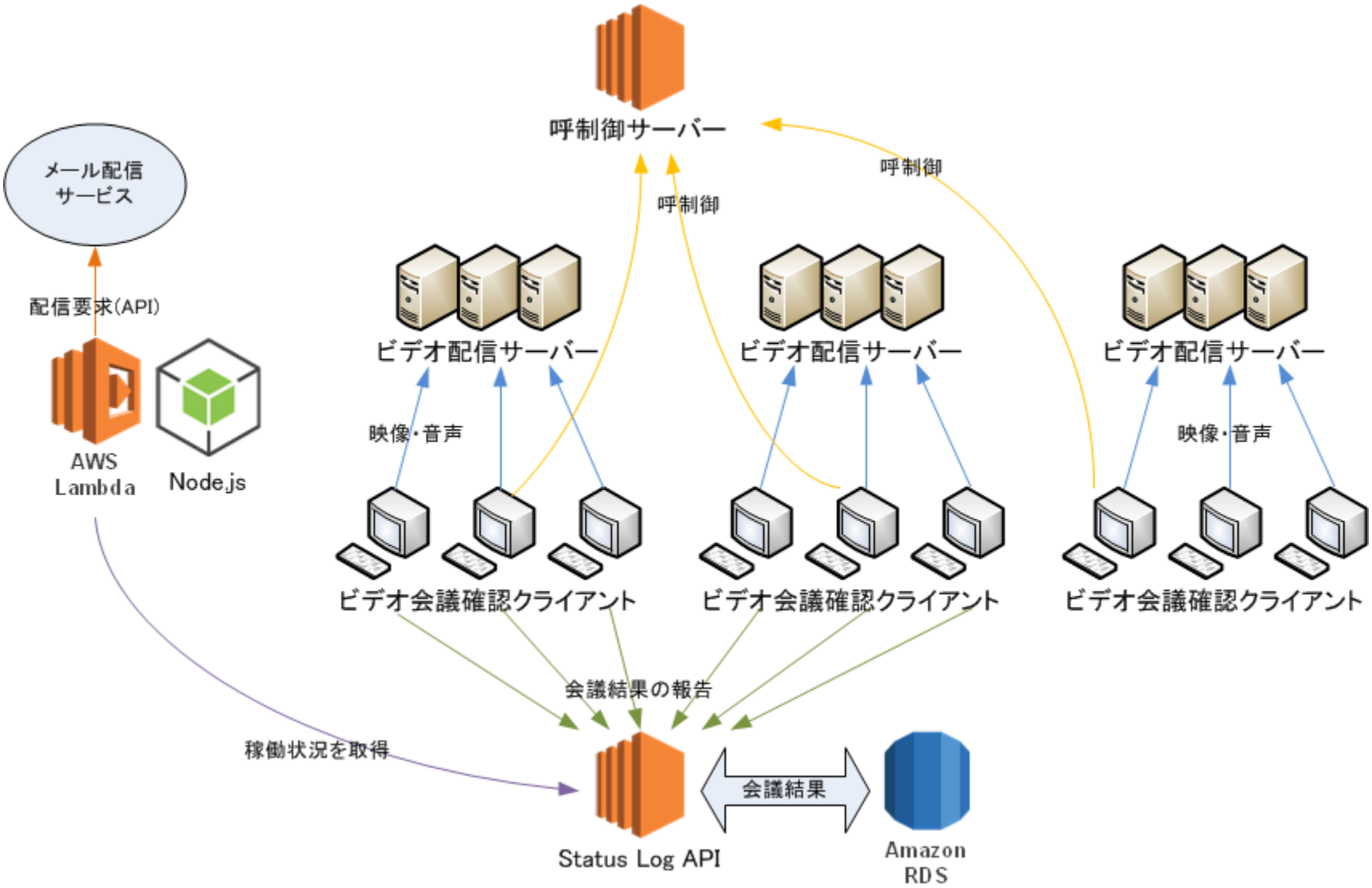
RICOH
imagine. change.

- お客様へ、障害発生時に早くお知らせしたいという要望

- ビデオ会議監視システムを利用
  - 状態は Status Log API から取得
    - JavaScript (TypeScript)
    - AWS Lambda上で動作
- しきい値を超えた場合にメールでお知らせ
  - 地域ごとに
    - どの地域で使えるか、使えないか
  - 外部のメール配信サービスを利用
    - 提供されている API を操作してメールを送信

全体図

RICOH
imagine. change.



自動化できない部分も

RICOH
imagine. change.

- 障害復旧
  - 運用チームが担当
- お客様対応
  - コールセンターと技術支援チームが担当

まとめ

RICOH
imagine. change.

- うちはこんな感じでやっています
  - マルチクラウド化
    - 障害発生時は地域単位で除去することも
      - EUは全部切り離し、など
  - 独自実装も
    - ビデオ会議に特化した結果
  - 完全に復旧させるのではなくて、とりあえずサービスを継続させる方向
  - お客様通知も自動的に
- 結果、各階層で障害対策が
  - ISP さん、DC 事業者さん、クラウド事業者さんも対策されていますが
    - 自分たちで制御できるかどうかが重要だったり
  - 障害発生時に「待つ」よりは「対策を進める」

議論したいこと

RICOH
imagine. change.

- みなさん、どうしています？
  - どこまで自動化？どこまで頑張る？頑張れる？
    - 事業規模・サービス内容にも依存しますが
  - 切り替えの 270 秒ってどう思います？
    - 60秒未満の「瞬断っぽい」のもビデオ会議には影響が大きかったり
  - クラウドロックインについてどう思います？
    - 「長いものには巻かれろ」的メリットもあるし…
    - 「おんぶに抱っこ」もそれはそれでちょっと…

RICOH
imagine. change.

ありがとうございました。


RICOH
imagine. change.