



【NGINX】 NGINXにたどり着くまでの軌跡 API Gatewayの夢と本当に必要だったもの

湯川 勇太

DXエンジニアリング1
チーフエキスパート
NRIデジタル株式会社



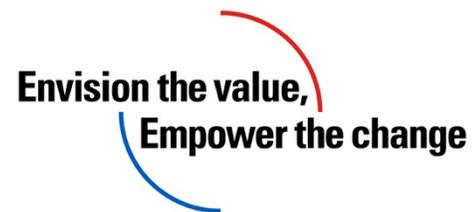
F5 AppWorld 2024

NGINXにたどり着くまでの軌跡 API Gatewayの夢と本当に必要だったもの

チーフエキスパート 湯川 勇太

NRIデジタル株式会社
DXエンジニアリング1

2024年09月04日



01

自己紹介

02

前置き：APIを保護する手段としてのNGINX Plus

03

本日のテーマ：API Gatewayの夢と現実

04

まとめ

自己紹介

■ 湯川 勇太

■ 略歴

- 前職では家電量販店のECサイトやECモールを8年間担当
- 2018年NRI中途入社
- 2022年4月NRIデジタル出向
- 現在は大手物流企業様で方式設計や技術検証、障害調査などを担当

NRIデジタルから見たJava on Azureのメリットと使い所
Javaのクラウド活用を学ぶ「Java on Azure Day 2022」が今年も開催

2022年04月22日 10時00分更新 文 ● 大谷イビサ 編集 ● ASCII

BI | シェア 一覧 ツイートする 本文印刷

2022年4月21日、マイクロソフトはAzure上のJava活用にフォーカスした「Java on Azure Day 2022」を開催した。基調講演に登壇したMicrosoft Corporationの寺田 佳央氏は、この1年のJava on Azureのアップデートを紹介。また、ヤマト運輸のシステムでJava on Azureを実践しているNRIデジタルの湯川 勇太氏をゲストに招いて、クラウド活用のポイントについて聞いた。



TECH Powered by NRI

企業IT テクノロジー ホワイトペーパー セミナー

連載 ゼロからはじめる実践 Kubernetes セキュリティ

コンテナ技術の基礎 (1) 「カーネルの分離 namespace」

掲載日 2022/02/14 11:09 更新日 2022/03/09 09:43

f t g

コンテナ kubernetes Microsoft Azure

- 目次
- 1 .namespaceとは
- 2 .PID プロセス群
- .Mount マウントポイント

目次を戻す

転職ノウハウ

- 平当にあったエンジニアの転職体験談
- 職場で「好かれる」「嫌われる」行動パターン5つ！ 共通点は〇だった
- AI時代に「必要とされる人材」とは?? 4月1日より...
- 71%の人が仕事を辞めたいと思った経験あり。その理由と対処法は??
- あなたが「本職発掘できる職種」もサクッと診断!
- エンジニアあるある!? システム開発現場・業務IT用語辞典

「クエリ診断 - 大規模システムデータベースの健康診断をどのように実現するか」 Qiita Conference 2023 Autumn イベントレポート

Tweet Share BI Hatena Pocket

RSS

Qiita Conference 2023 Autumn

クエリ診断

エキスパート 湯川 勇太

NRIデジタル株式会社

2023年10月25日

NRI NRIデジタル NRI Digital まだ見ぬ世界を切り拓こう。



NRIデジタルの紹介

- NRIデジタルは、野村総合研究所（NRI）グループのデジタルビジネス専門会社です。デジタル化戦略の構想から、先端ITソリューションの選定・構築、事業の実行支援、プロジェクト全体の検証・改善に至るまで、顧客企業のDXをトータルに支援しています。
- NRIデジタルの公式ページではAzureやAWSなどパブリッククラウドについて技術情報発信を行っています。

NRI Digital
Azure Synapse Link for SQLの非機能検証
湯川 秀太
2022.12.15 クラウド 技術検証 Azure 分析用DB

こんにちは、NRIデジタルの湯川です。

業務システムで発生したばかりのデータをほぼリアルタイムで分析したいという場合、AzureではEventHubsやStream Analyticsを使ったストリーム処理を作り込む方式がリファレンスアーキテクチャとして提示されています。

Microsoft Azure

こうしたアプローチは業務システムか分析システムのどちらかで、もともとストリーム処理を扱っているのであれば、特に問題なく採用できるのだと思いますが、リアルタイム化のために新たにストリーム処理を設計しなければならない場合には比較的重いデータ連携手法になると思います。

もっと手軽に、業務システムの中にあるデータをなるべくリアルタイムに分析したいという要望に応えるために生まれたのがAzure Synapse Link for SQLです。

<https://www.nri-digital.jp/>

NRI Digital
大規模プロジェクトでのAKS活用に向けた取り組みの紹介
荒井 良祐
2022.12.22 クラウド 技術検証 kubernetes Azure

1. はじめに

NRIデジタルでは多くの大規模プロジェクトを推進していますが、新規性が高い技術を導入する場合はいつも多くの苦労があります。

本稿では、そのような新技術導入のプロジェクト例として、Azure Kubernetes Service（以下、AKS）活用に向けた取り組みについてご紹介します。

直近のプロジェクトの中での活動になりますが、大所帯のプロジェクトで関係者が多く、また発足当時は、Kubernetesも普及していたとはいえ業務系の中では比較的新しい仕組みでしたので、多くの関係者に、いかに広めるか、作ってもらるか、使ってもらるか、といった観点で取り組みを行っています。

以下では、その内容を紹介します。

導入フェーズの取り組み

はじめに、導入フェーズの取り組みとして関係者にkubernetesに対する理解を促進する取り組みについて記載します。

Kubernetesをプロジェクト関係者に広く進めようとするために、プロジェクト内で実施した勉強会やモチベーション維持の取り組みについてご紹介します。

NRI Digital
空調機シミュレーションにおける Amazon Timestream 活用【前半】
島 良本
2022.10.18 クラウド AWS 技術検証 時系列データベース ストリーク可能化

こんにちは、NRIデジタルの島です。

先日、AWSのフルマネージド型サーバーレス時系列データベースである「Amazon Timestream(以下Timestream)」が東京リージョンでGAされました。

[Amazon Timestream is now available in the Asia Pacific \(Sydney\) and Asia Pacific \(Tokyo\) AWS regions](#)

実際のユースケースを元にシミュレーションデータを作成し、そのデータ処理および可視化を東京リージョンで試してみたので、その内容を共有したいと思います。今回は【前半】として、Timestreamの概要、Timestream検証構成・構築、データの書き込みについて紹介します。

Timestreamとは

AWSには様々なマネージド型データベースがありますが、その中でTimestreamはフルマネージド型サーバーレス時系列データベースとして時系列データを専門に扱うデータベースになっています。

AWS BlackBeltより引用



01

自己紹介

02

前置き：APIを保護する手段としてのNGINX Plus

03

本日のテーマ：API Gatewayの夢と現実

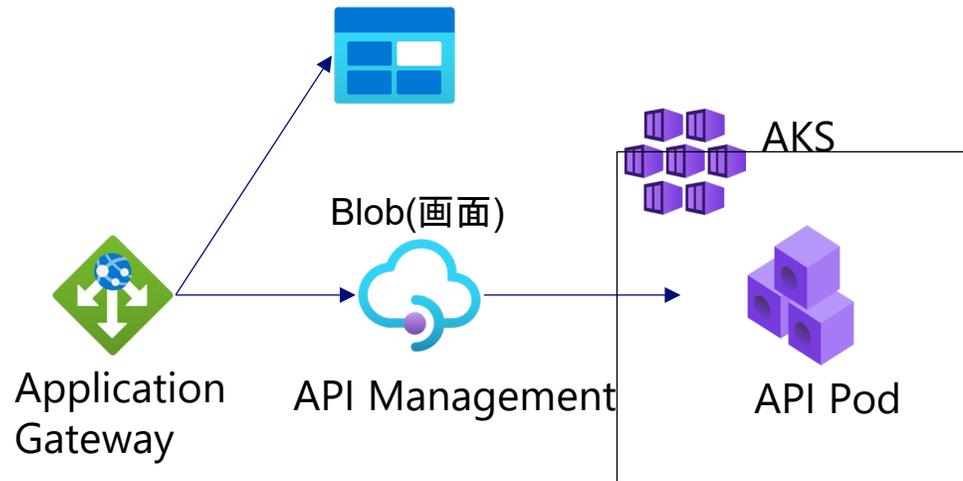
04

まとめ

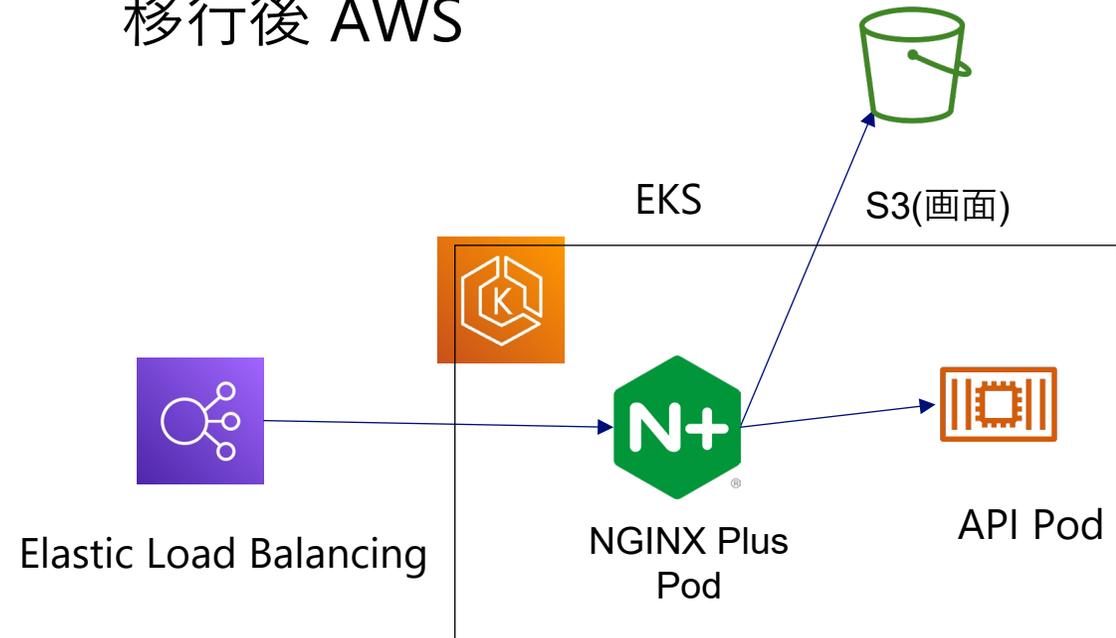
APIを保護する手段としてNGINX Plusを活用している

- 担当している大手物流企業様ではAPIを保護する手段としてNGINX Plusを活用している
- APIを保護する手段としてももとはAzureが提供するAPI Managementを利用していた
- AzureからAWSへの基盤移行をきっかけにAPI ManagementをNGINX Plusに置き換えた

移行前 Azure



移行後 AWS



前置き

NGINX Plus採用の理由

- APIを保護する手段としてNGINX Plusを採用した理由については、2024年2月16日に開催されたNGINXユーザ会2024でご紹介させていただいた

NGINXユーザ会2024

クラウド間の架け橋:
AzureからAWS/プライベートクラウドへ、
NGINX Plusを活用した移行アプローチ

エキスパート 湯川 勇太

NRIデジタル株式会社
DXエンジニアリング1

アソシエイト 長瀬 知輝

株式会社野村総合研究所
産業基盤サービス部

2024年02月16日

NRI



NGINX Plus採用の背景

NGINX Plus採用の理由

なぜNGINXなのか

シンプルで扱いやすい

なぜNGINX Plusなのか

機能要求を満たし、安心のサポートを提供

なぜNGINX Ingress Controllerではないのか

NGINXのノウハウを活かしたい

01

自己紹介

02

前置き：APIを保護する手段としてのNGINX Plus

03

本日のテーマ：API Gatewayの夢と現実

04

まとめ

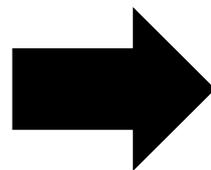
API Gatewayの夢と本当に必要だったもの

- 本日のテーマはNGINXユーザ会2024の前日譚。API Gatewayの検討初期からNGINX Plusまでの検討経緯
- 当初、API Gatewayには多くの役割を期待していたが、現状では以下に整理されている

検討初期に期待していた役割

リトライ
オンラインの冗長化
ルーティング
通信制限
開発者ポータル
API利用申請ワークフロー
複数APIの統合
開発用サーバ
API認証

バリデーション
API認可
レガシーシステムへの総合窓口
ヘッダ書き換え
流量制限
キャッシュ
CORS
問い合わせ窓口
利用状況の可視化
API権限管理



実際に担っている役割

リトライ
オンラインの冗長化
ルーティング
通信制限
開発者ポータル
API利用申請ワークフロー
複数APIの統合
開発用サーバ
API認証

バリデーション
API認可
レガシーシステムへの総合窓口
ヘッダ書き換え
流量制限
キャッシュ
CORS
問い合わせ窓口
利用状況の可視化
API権限管理

なぜ夢をみたのか？

■ APIを純化したい

- API本体をビジネスロジックに集中させ、それ以外のものをAPI Gatewayに担わせたい
- バリデーション、複数APIの統合、キャッシュ、リトライ、CORS、ルーティング、ヘッダ書き換え、レガシーシステムへの窓口

■ APIの利用を促進したい

- 有識者の確保や関係者間の調整を減らし、スムーズにAPIを利用できるようにしたい
- API利用申請ワークフロー、問い合わせ窓口、開発サーバ

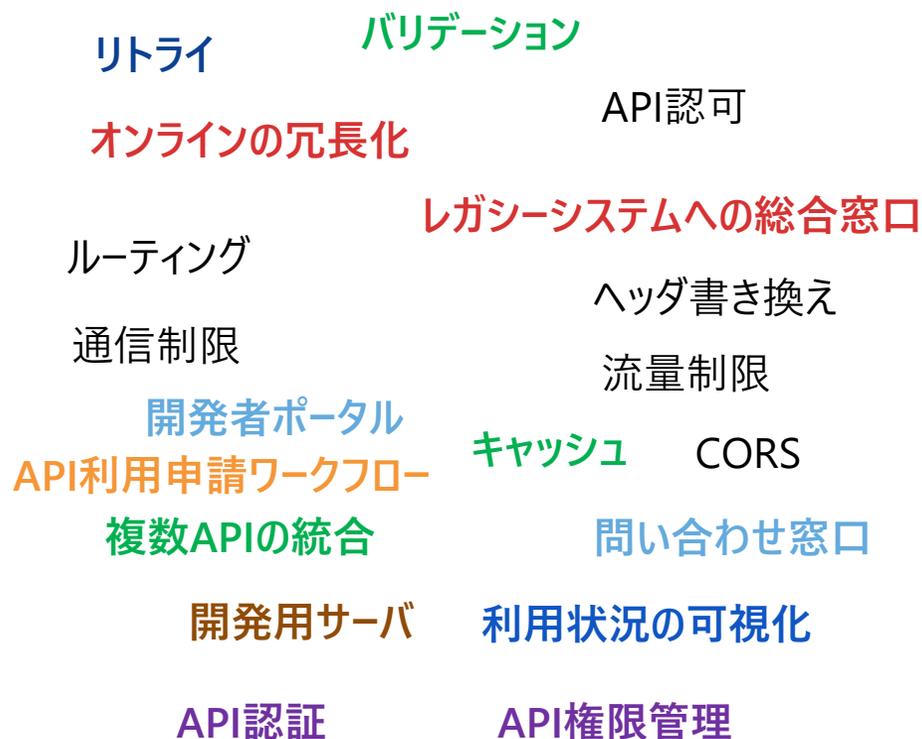
■ APIを保護したい

- APIの利用を制限し、想定外の利用による問題を防止したい
- 通信制限、流量制限、API認証、API認可、API権限管理、オンラインの冗長化、開発ポータル、利用状況の可視化

API Gatewayは全部機能として提供してる！ 使いこなせばすべての課題が解決するかも！

夢と現実 夢のあと

- 要件がなくなったわけではない
- ほとんどの要件はAPI Gateway以外のものを使って実現している

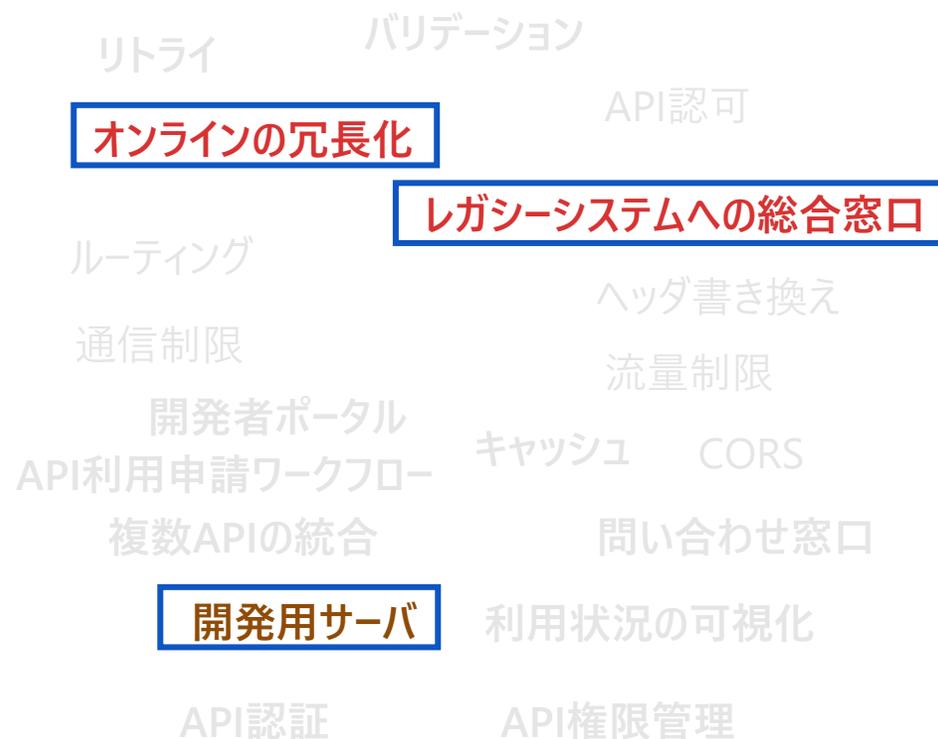


機能実装先

- NGINX Plus (API Gateway)
- API提供システム内部
- APIの呼び出し元
- 廃止、断念
- API Gatewayとは別に検討中
- 利用者種別毎の申請システムなど
- 認証・権限管理サービス
- APM
- アプリ実装、SoapUIなどのツール利用

夢と現実 夢のあと

■今回は夢と現実のギャップが大きかった青枠の3項目について取り上げる



機能実装先

- NGINX Plus (API Gateway)
- API提供システム内部
- APIの呼び出し元
- **廃止、断念**
- API Gatewayとは別に検討中
- 利用者種別毎の申請システムなど
- 認証・権限管理サービス
- APM
- **アプリ実装、SoapUIなどのツール利用**

開発用サーバ

■ どんない夢をみたのか

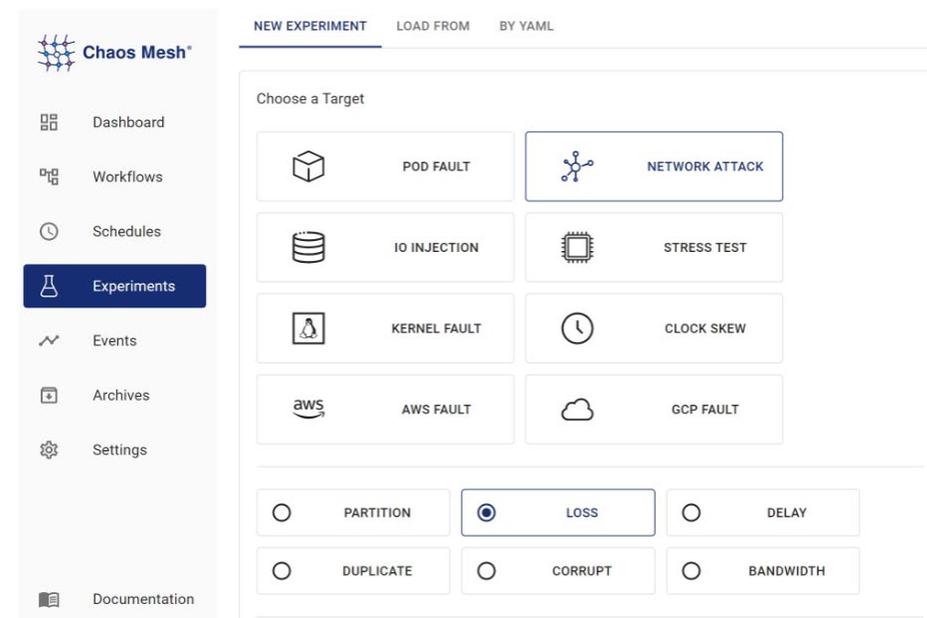
- オンプレを中心に、本番同等の開発環境がなく、性能テストでは利用できないAPIが多数存在している
- API Gatewayでさくっと開発で使うスタブサーバを立てたい

■ 直面した現実

- さくっと開発で使うサーバはAPI Gatewayを使わなくても、SoapUIなど様々な方法で立てられる
- 応答が早すぎて困る

遅いスタブサーバを利用することは、性能テストでは非常に重要
利用しているスタブサーバ用のツールにsleepを仕込むことが困難
な場合は、tc qdiscやChaos Meshのネットワークカオスを使って
遅延を発生させる

データセンターをまたがるケースなどでは、パケットロスをあえて発生
させた状態でのテストもしておきたい



<https://chaos-mesh.org/docs/simulate-network-chaos-on-kubernetes/>

レガシーシステムへの総合窓口

■ どんな夢をみたのか

- SOAPなどの古いプロトコルで実装されたレガシーなシステムをREST化することで呼び出し側の仕組みを単純化したい
- APMなどが導入されていない古いシステムをどのシステムがどのくらい利用しているのかを把握できるようにしたい
→レガシーシステムへの総合窓口になるAPI Gatewayを立てることで、解決できるのではないか？

■ 直面した現実

- 総合窓口化された**API Gatewayは誰が運用するのか**
- API Gatewayで障害が発生した場合、バックエンドのシステムが稼働していても、サービスとして利用できなくなる
→オンライン連携を仲介するようなプロダクト横断の共通基盤は構築せず

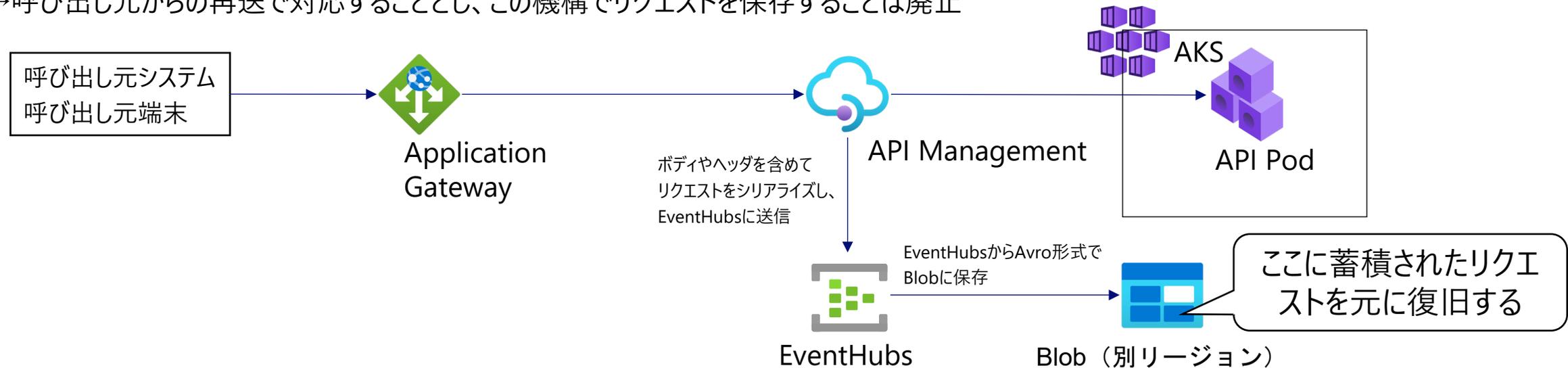
オンラインの冗長化

■ どんな夢をみたのか

- APIで障害が発生した際に、なるべく呼び出し元からの再送に頼らずに復旧できるようにしたい

■ 直面した現実

- 復旧作業が人の判断にならざるをえず、大量のトランザクションデータから復旧させるのは大変
- 一律で再送させることもできるが、前提となる状態の変化で失敗する可能性もある
- 上位障害すべてに対応できるわけではなく、呼び出し元からの再送に頼るしかないケースは残る
→呼び出し元からの再送で対応することとし、この機構でリクエストを保存することは廃止



本日触れなかった項目について

- 本日触れなかった項目について、一言解説を記載した
- 機会があれば別の場でより詳細について触れさせていただきたい

要件	実装先	一言解説
リトライ	APIの呼び出し元	リトライを多段で仕込むと輻輳などの問題が起きる 管理可能な一番外側のリトライに集約した方が設計・管理が簡単になる
バリデーション、キャッシュ、複数APIの統合	API提供システム内部	API Gatewayにロジックを記述することはできるが、アプリよりテストがしにくい 現在はアプリで対応している
開発者ポータル、問い合わせ窓口	API Gatewayとは別に検討中	APIだけで必要なわけではなく、ファイル連携やメッセージ連携でも必要である データカタログ等と合わせて検討中
API利用申請ワークフロー	利用者種別毎の申請システムなど	利用者が社外の場合は契約等が必要でAPI Gatewayだけでまかなえない 社内の場合は担当者間のやりとりをしているのが現状
利用状況の可視化	APM	現状はAPIの利用状況に応じて制限と課金を行うような要求がない システム運用として利用状況を可視化するという意味ではAPMを利用している
API認証、API権限管理	認証・権限管理サービス	認証にはIDaaSを利用している 権限管理は認証時の返却項目と権限管理用のシステムを組み合わせている

01

自己紹介

02

前置き：APIを保護する手段としてのNGINX Plus

03

本日のテーマ：API Gatewayの夢と現実

04

まとめ

まとめ：夢から覚めて学んだこと

- 一つの仕組みにあまりにも多くのことを要求しすぎると結果としてコスト高になる
- 多機能なものを採用してすべての課題解決を目指そうとしてもなかなか進まない
- 夢を見すぎかなと思ったら、一度立ち止まって考えてみよう



**Envision the value,
Empower the change**

