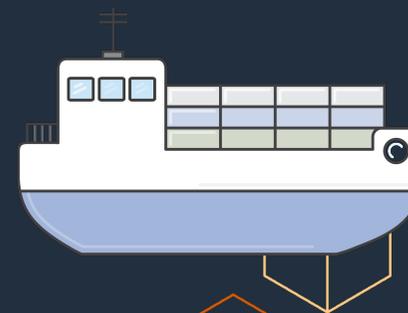




AWS Blackbelt Container

サービスメッシュ入門

2021-11



このセッションで扱うこと

- サービスメッシュとは何か
- サービスメッシュが求められるようになった経緯
- サービスメッシュ実装

サービスメッシュとは？



サービスメッシュとは

- アプリケーションレベルの通信を、アプリケーション自身が制御するのではなくインフラストラクチャーで制御できるようにする技術

アプリケーションが行う通信制御

- HTTP 通信のリトライやタイムアウト
- 通信のトレーシングやログ、メトリクスの取得
- TLS を使用した暗号化通信



サービスメッシュとは

- アプリケーションレベルの通信制御を、サービスメッシュの基盤で行うので、アプリケーションに組み込む必要がなくなる

- HTTP 通信のリトライやタイムアウト
- 通信のトレーシングやログ、メトリクスの取得
- TLS を使用した暗号化通信

サービスメッシュ基盤



サービスメッシュが 求められるようになった背景



サービスメッシュが求められるようになった背景

分散システムにおける様々なユースケース解決のため

- 動的サービス構成法の普及（A/Bテスト、カナリアリリース、サービスディスカバリ）
- アーキテクチャ、言語の多様化
- さらにクラウドによる動的サービス作成頻度向上
- 可観測性のために、標準化されたモニタリング要求
- サービス間通信の信頼性を積極的に確保する実装の徹底実施
- セキュリティ

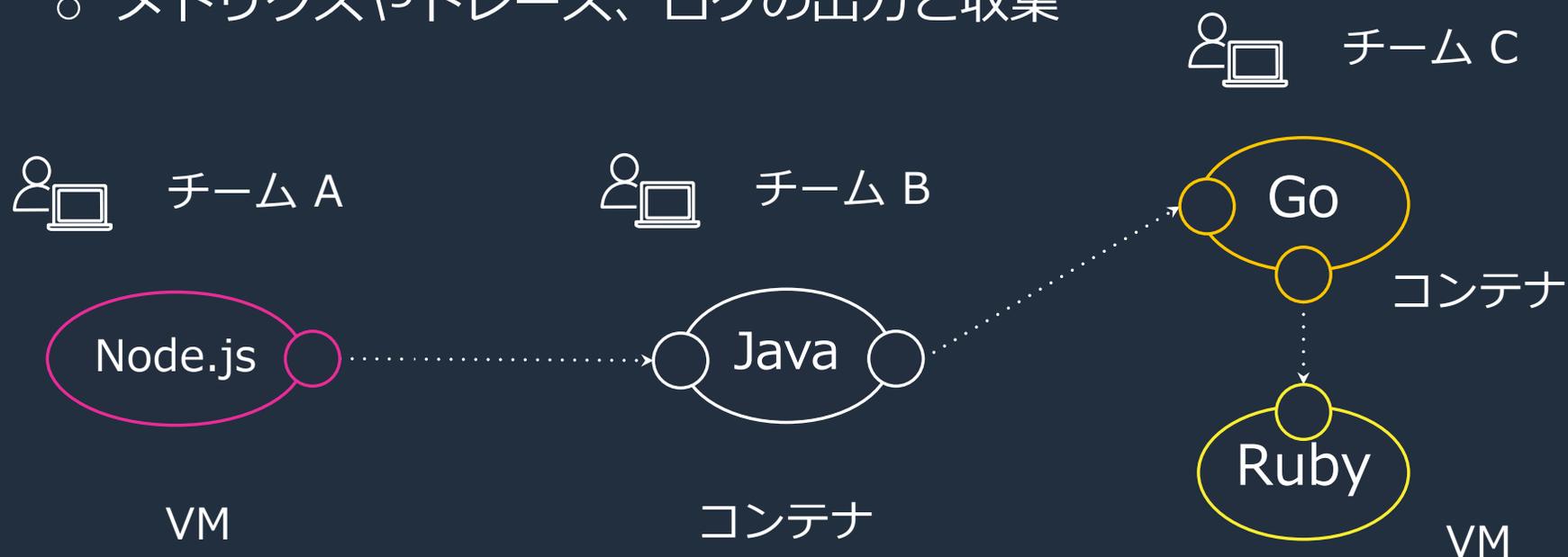
例：多様なアプリケーションが通信を行うシステム

- チームごとに最適な技術を選択してアプリケーションを動かす
- プログラミング言語だけでなく、VM やコンテナなど様々なコンピューティングプラットフォームを選択できる

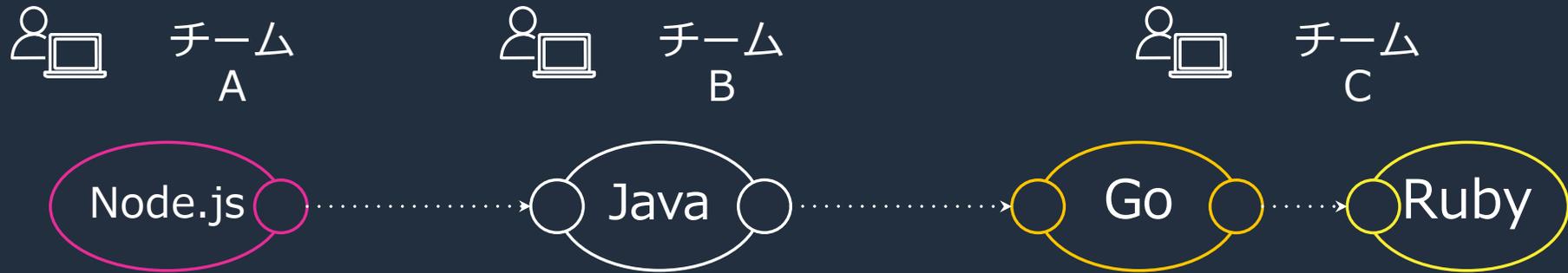


全てのアプリケーションに 同じような通信の仕組みが必要

- 信頼性の確保
 - リトライやタイムアウト、暗号化通信の実装
- 可観測性の確保
 - メトリクスやトレース、ログの出力と収集



ライブラリによる通信制御の共通化



通信の仕組みを共通ライブラリとして実装、配布

共通基盤チーム

共通ライブラリ

Node.js

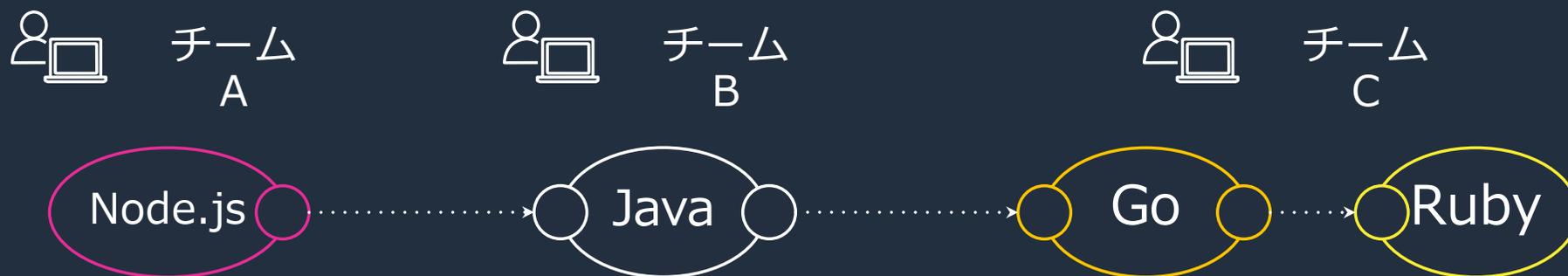
Java

Go

Ruby

- リトライやタイムアウト、暗号化通信の実装
- メトリクスやトレース、ログの出力と収集

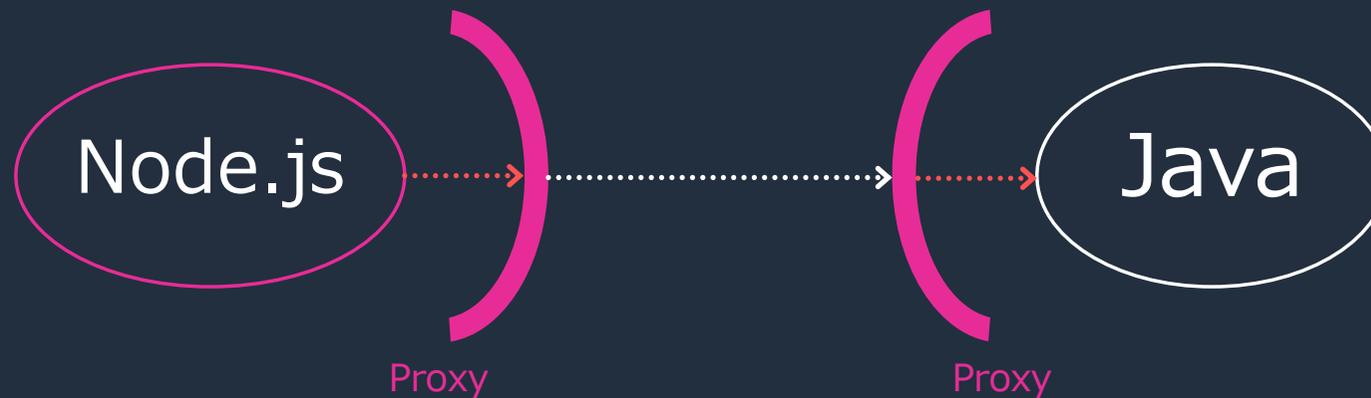
共通ライブラリの導入における課題



- 共通ライブラリを入れると依存関係が衝突する
- 言語ごとにライブラリを実装する必要があり負担が大きい

アプリケーション開発者・共通基盤チーム共に運用負荷が増大する

解決法：アプリケーションから通信処理を分離



サービスマッシュが注力するのはアプリケーション間通信。通信制御、ログ・メトリクスの収集を代わって行う

APIゲートウェイや、EdgeサービスではIngressや、North-Southトラフィックを対象にするところが根本的な違い

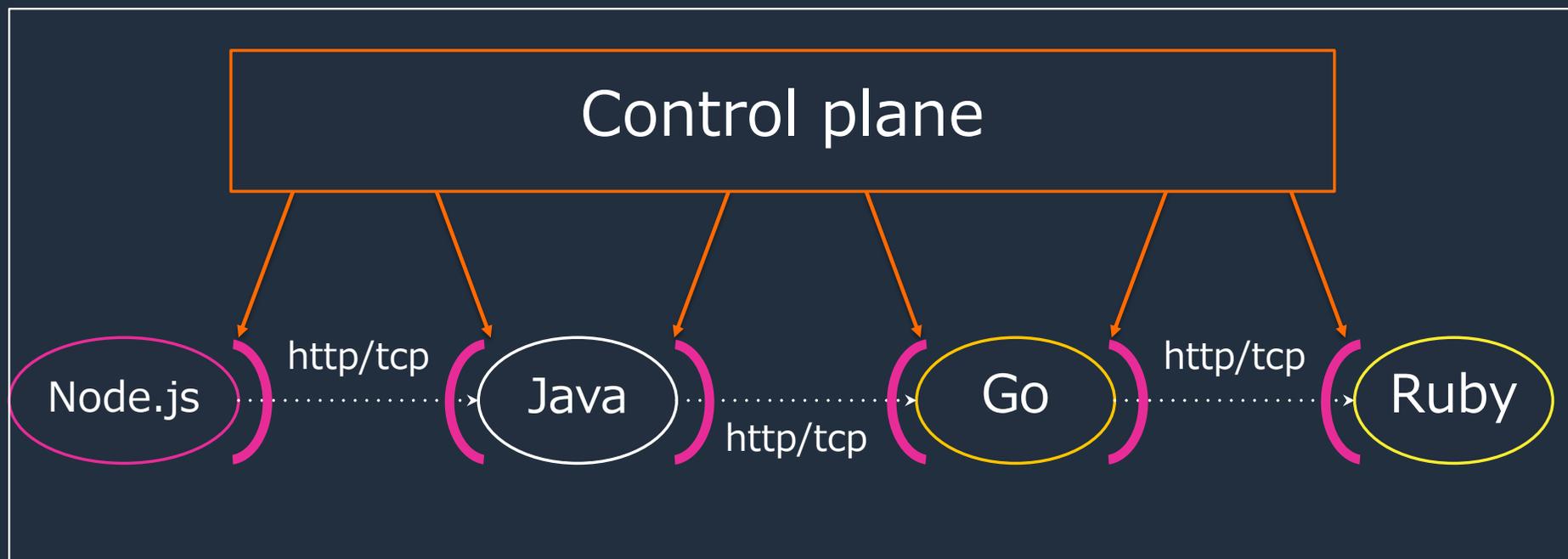
サービスメッシュ実装の紹介

サービスメッシュの仕組み

プロキシをコントロールプレーンで管理

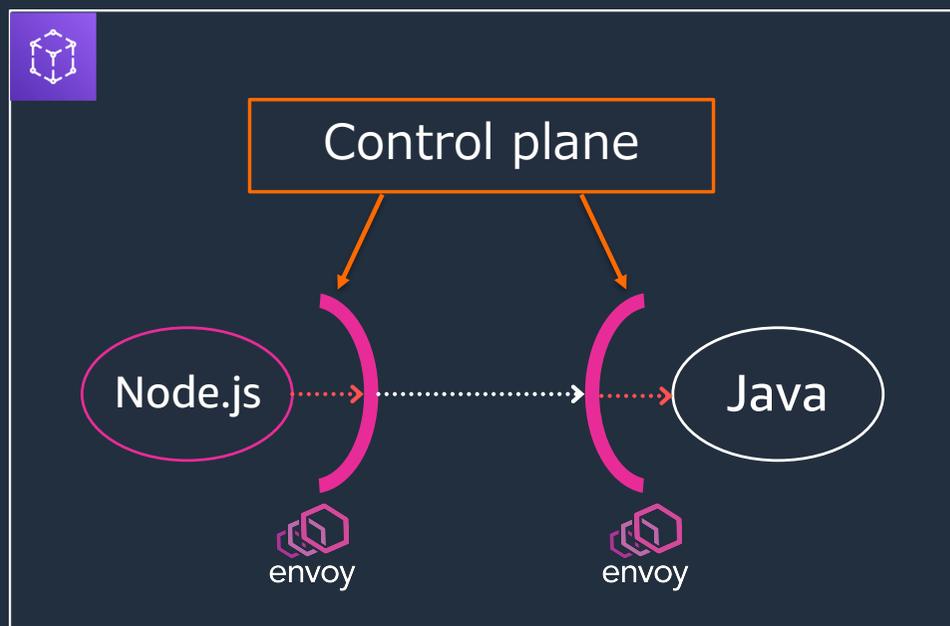
=> アプリケーションレベルの通信制御をインフラストラクチャ側で管理可能に

サービスメッシュ基盤



AWS App Mesh

サービスメッシュを管理するコントロールプレーンを提供



アプリケーションレベルの通信をネットワークモデルとして定義

ネットワークモデルを Envoy の設定に変換して配布

ログやメトリクスは外部サービスと連携収集(CloudWatch等)

2018年reinvent発表&2019年3月GA

サービスメッシュの代表的・歴史的OSS実装

- 2013年 SmartStack
- 2014年 Prana
- 2017年 Istio
- 2018年 Linkerd 2.0, Consul Connect, SuperGloo
- 2019年 Maesh, Kuma

CNCFにおいてKubernetes でのサービスメッシュインターフェースについての標準として、Service Mesh Interface あり。最新は2021年1月のversion 0.6.0

まとめ サービスメッシュ入門



このセッションで扱ったこと

- サービスメッシュとは何か
 - アプリケーションレベルの通信を、アプリケーション自身が制御するのではなくインフラストラクチャーで制御できるようにする技術
- サービスメッシュが求められるようになった経緯
 - 分散システムにおける様々なユースケース解決のため
- サービスメッシュ実装
 - AWS App Mesh はEnvoy Proxyをデータプレーンとして、コントロールプレーンと可観測性のために CloudWatch や X-Ray との連携を提供

AWSの監視サービス群 全体像

Observability

Amazon CloudWatch ServiceLens

Container Insights

Lambda Insights

Synthetics

Contributor Insights



CloudWatch Logs



CloudWatch metrics



AWS X-Ray



Amazon Managed Service for Grafana

Do it Yourself (DIY)



Amazon Elasticsearch Service - Logs



Amazon Managed Service for Prometheus



AWS X-Ray



CloudWatch Logs



CloudWatch metrics

Insights



CloudWatch agent



X-Ray Daemon

Instrumentation



AWS Distro for OpenTelemetry

関連アイテム

- AWS Black Belt Container 2021
 - CON246 Observability – ログ入門
 - CON247 Observability – メトリクス入門
 - CON248 Observability – トレース入門