AWS Black Belt Online Seminar

Amazon EventBridge Pipes

櫻谷 広人

Partner Solutions Architect
2024/3

AWS Black Belt Online Seminar とは

- 「サービス別」「ソリューション別」「業種別」などのテーマに分け、アマゾン ウェブ サービス ジャパン合同会社が提供するオンラインセミナーシリーズです
- AWS の技術担当者が、 AWS の各サービスやソリューションについてテーマごとに動画を公開します
- 以下の URL より、過去のセミナー含めた資料などをダウンロードすることができます
  - https://aws.amazon.com/jp/aws-jp-introduction/aws-jp-webinar-service-cut/
  - https://www.youtube.com/playlist?list=PLzWGOASvSx6FIwIC2X1nObr1KcMCBBlqY

ご感想は X (Twitter) へ！ハッシュタグは以下をご利用ください
#awsblackbelt

内容についての注意点

- 本資料では 2024 年 3 月時点のサービス内容および価格についてご説明しています。AWS のサービスは常にアップデートを続けているため、最新の情報は AWS 公式ウェブサイト (https://aws.amazon.com/) にてご確認ください
- 資料作成には十分注意しておりますが、資料内の価格と AWS 公式ウェブサイト記載の価格に相違があった場合、AWS 公式ウェブサイトの価格を優先とさせていただきます
- 価格は税抜表記となっています。日本居住者のお客様には別途消費税をご請求させていただきます
- 技術的な内容に関しましては、有料の AWS サポート窓口へお問い合わせください
- 料金面でのお問い合わせに関しましては、カスタマーサポート窓口へお問い合わせください (マネジメントコンソールへのログインが必要です)

自己紹介

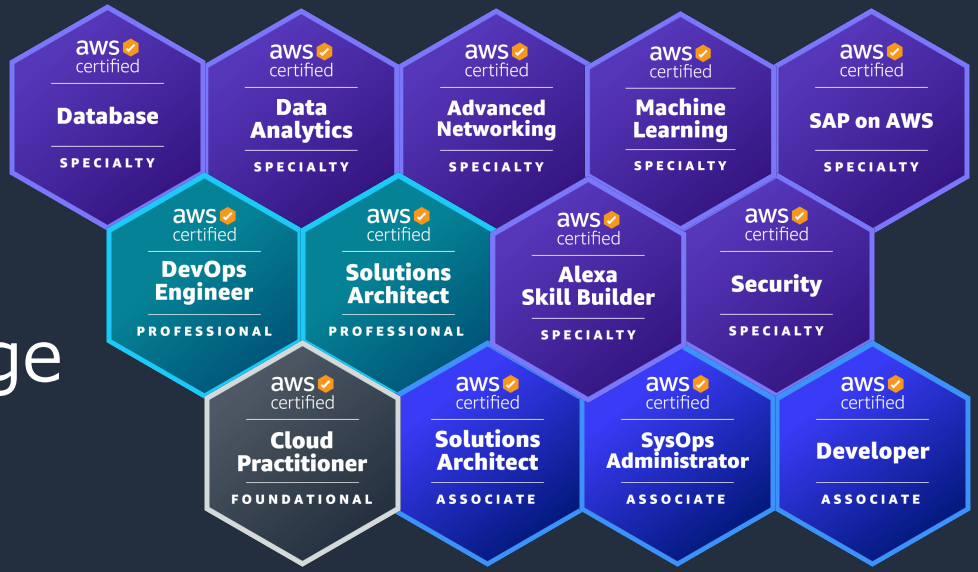
名前：櫻谷 広人 (Hiroto Sakuraya)

所属：AWS Technology Partnerships

SaaS, Partner Solutions Architect

経歴：主にバックエンドエンジニアとして Web サービスやネイティブアプリの開発に従事。前職では CtoC のスタートアップで執行役員 CTO を務める。

好きな AWS サービス：Amazon EventBridge



本セミナーの対象者

- Amazon EventBridge について深く学びたい方
- イベント駆動型アプリケーションの開発に興味をお持ちの方
- ノーコード/ローコードでのシステム間連携を実現したい方

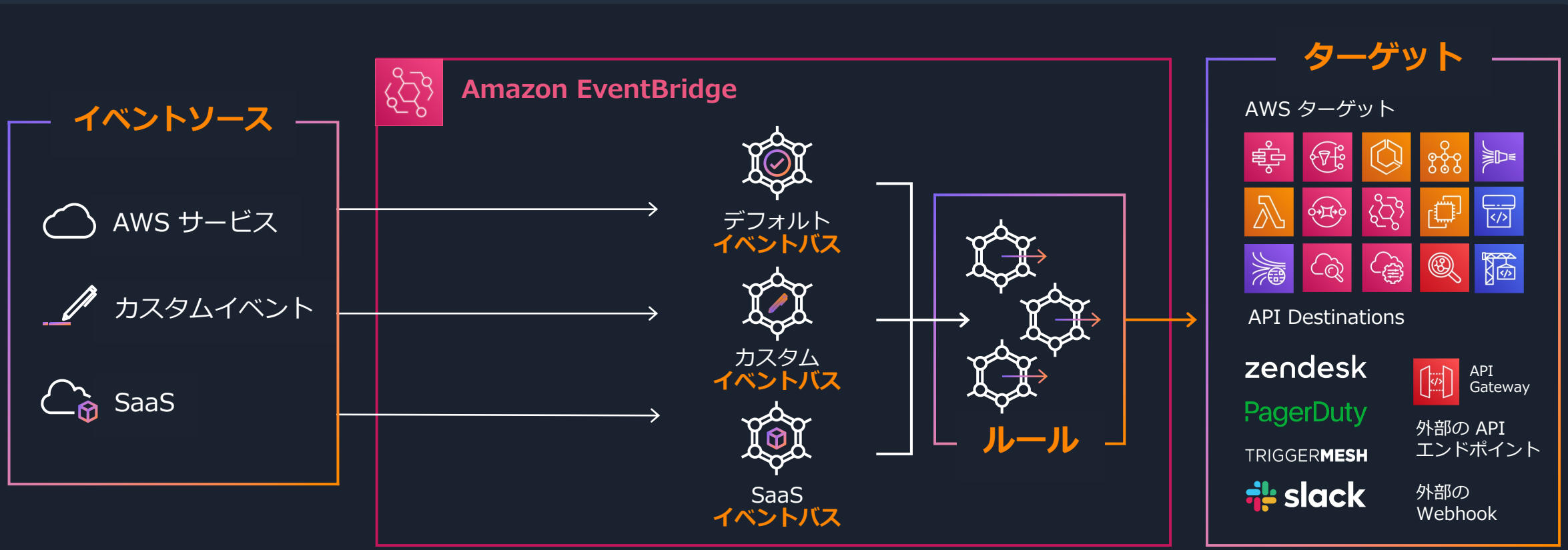
本セミナーの取り扱う範囲

- Amazon EventBridge Pipes について

* Amazon EventBridge の他の機能については別のセミナー動画をご覧ください

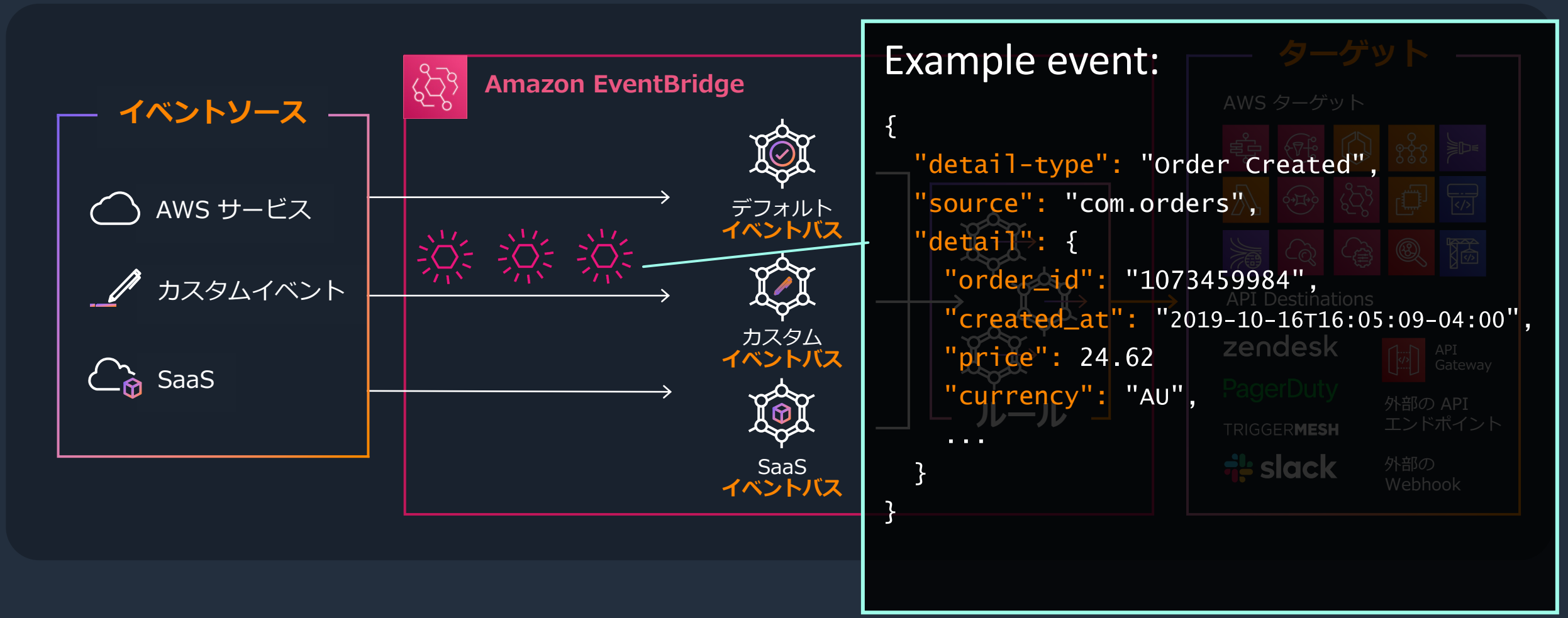
  - **Amazon EventBridge グローバルエンドポイント【AWS Black Belt】**
  - **Amazon EventBridge Scheduler 【AWS Black Belt】**

Amazon EventBridge とは

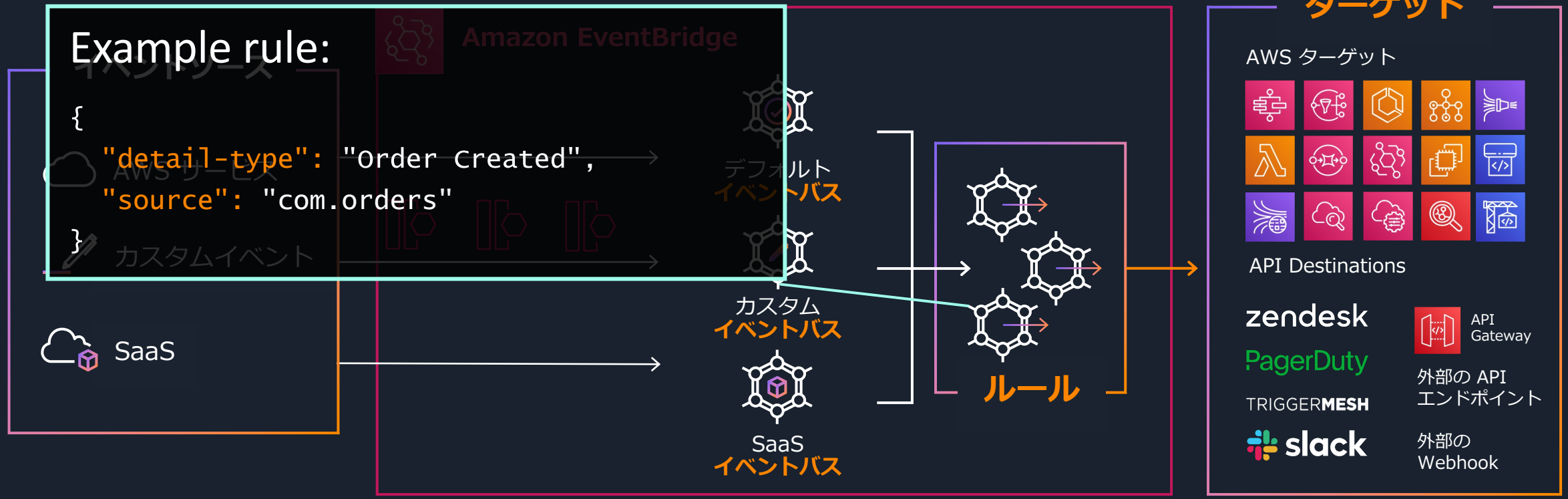


アプリケーション、統合された SaaS アプリケーション、および AWS のサービスから生成されたイベントを受信、フィルタリング、変換、ルーティング、および配信することができるサーバーレスイベントバス。大規模なイベント駆動型アプリケーションの開発を可能に。

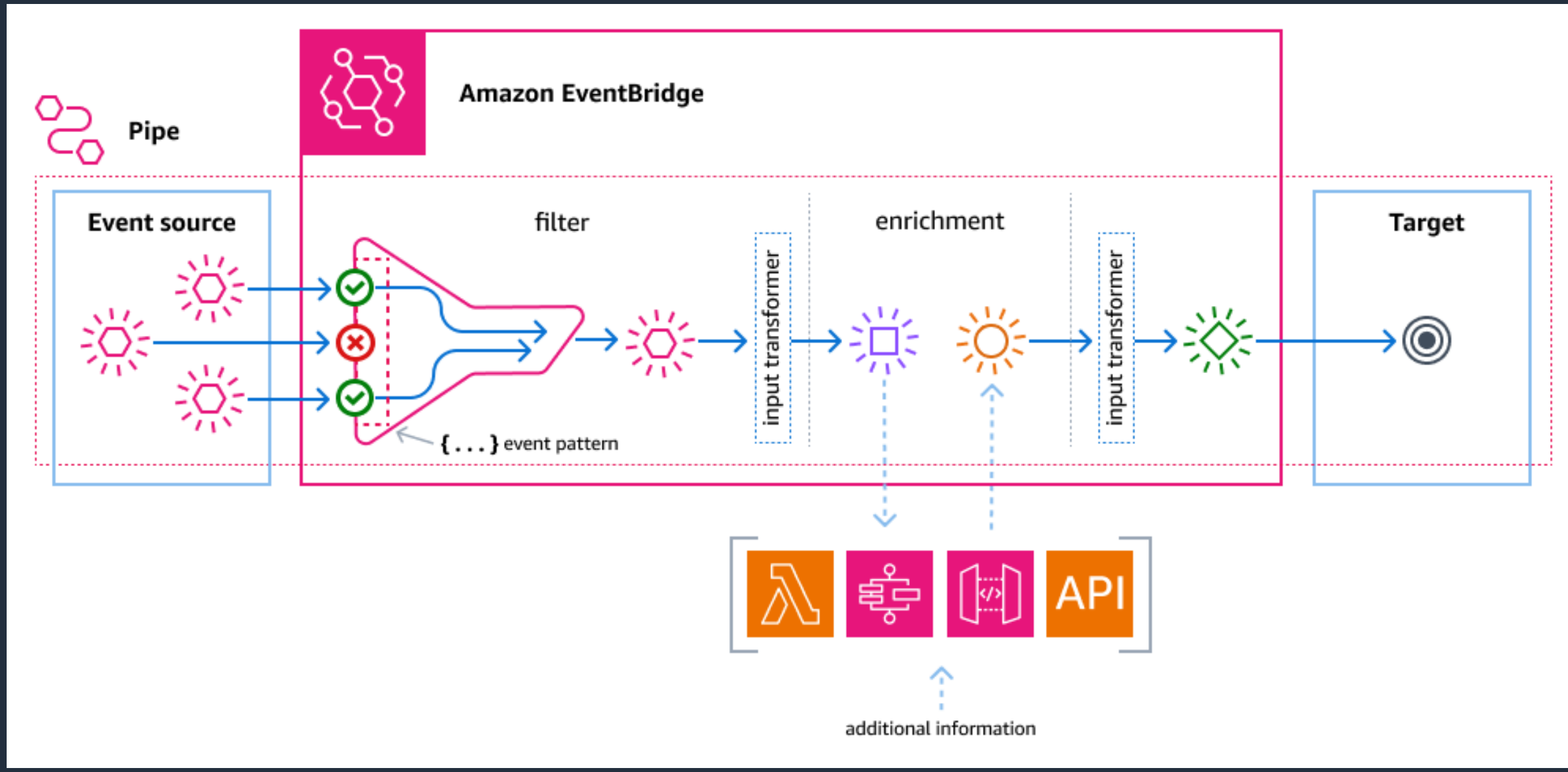
イベントバスを使用した場合



イベントバスを使用した場合

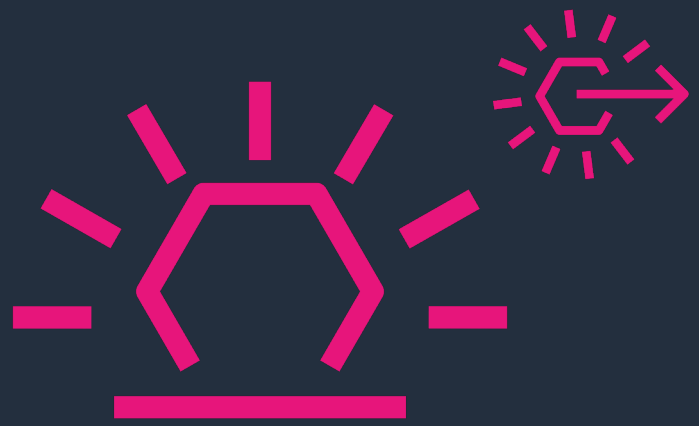


EventBridge Pipes とは



イベントバスとパイプの違い

イベントバス



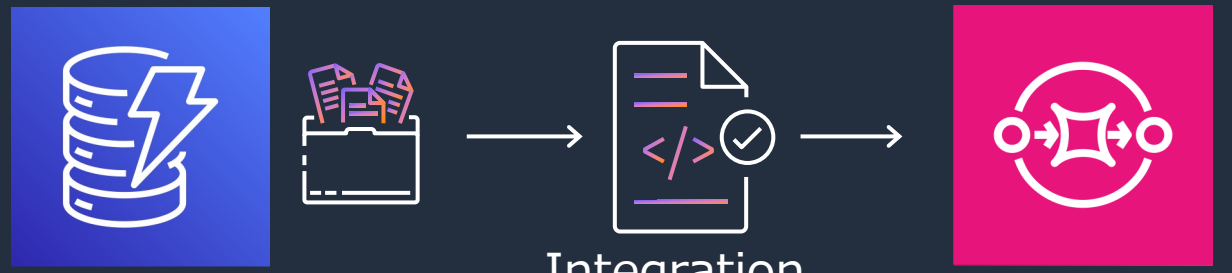
Many publishers
to many consumers

パイプ

Single publisher
to single consumer

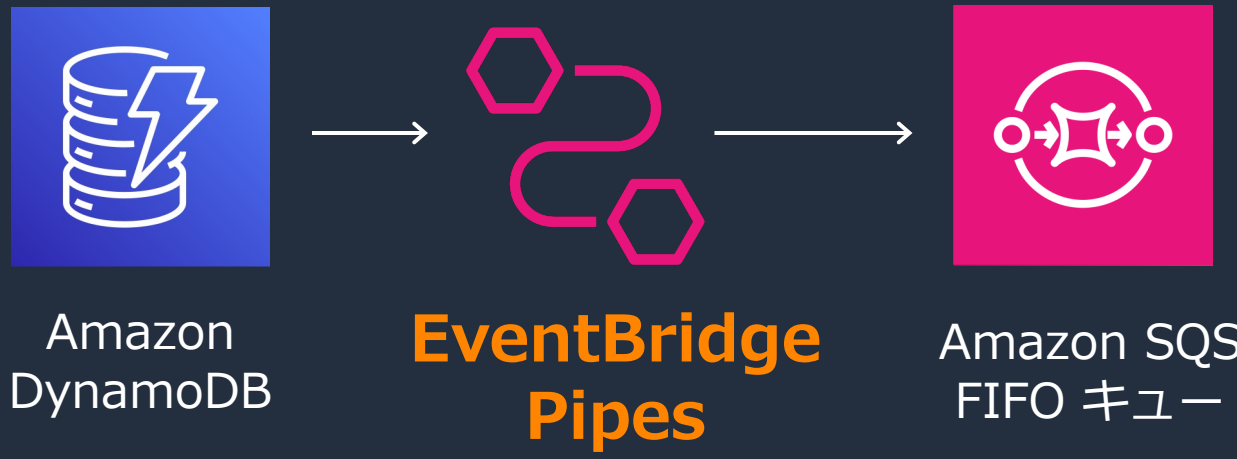
サービス間連携の課題

- サービスごとに異なる仕様や実装
- エラーハンドリング
- 配信順序の維持
  (*サポートしているサービスに限る)
- 認証、認可、流量制御
- パフォーマンステスト
- デプロイ
- 運用監視

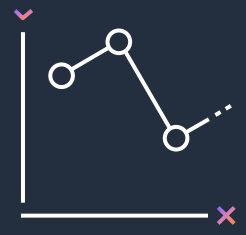


サービス間連携の課題

- サービスごとに異なる仕様や実装
- エラーハンドリング
- 配信順序の維持
  (*サポートしているサービスに限る)
- 認証、認可、流量制御
- パフォーマンステスト
- デプロイ
- 運用監視



EventBridge Pipes を利用することで TCO を削減



開発速度の
低下

バグ発生の
リスク

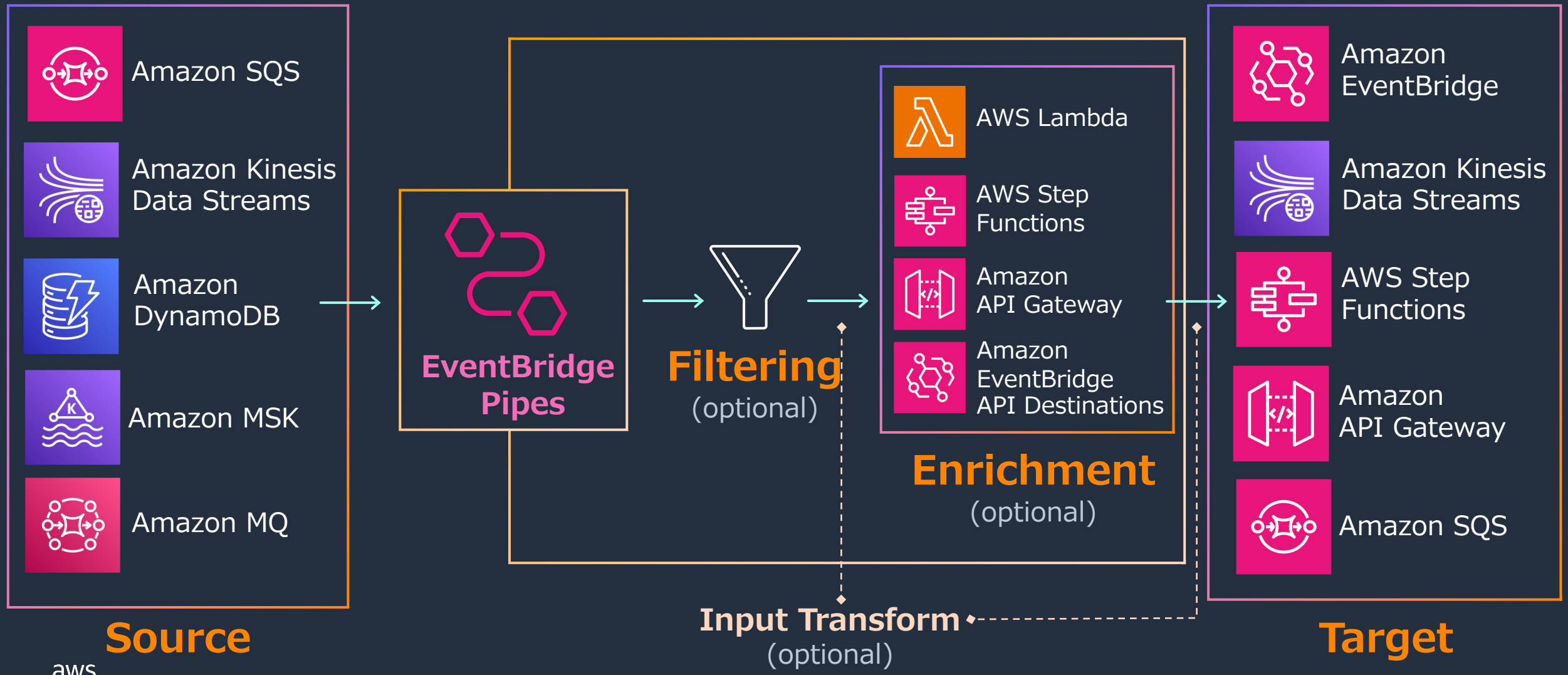
コードの
複雑化

運用負荷の
増加

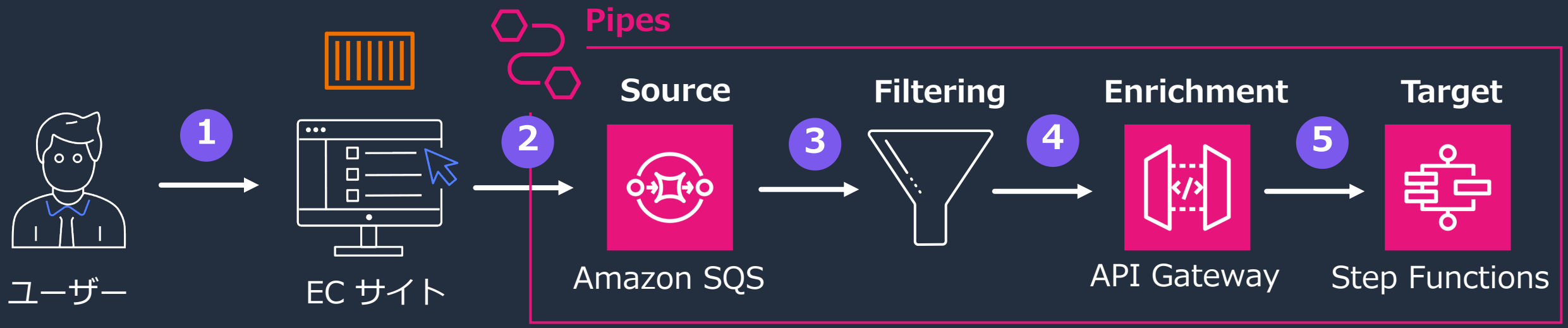
非効率な
ポーリング

High total cost of ownership

EventBridge Pipes を構成する要素



EventBridge Pipes の利用イメージ



1. ユーザーが Web から注文を行う
2. バックエンドのアプリケーションが注文内容を保存し、注文 ID と合計金額を SQS に送信
3. 一定の金額以上の注文にクーポンを付与するために、合計金額でフィルタリングする
4. ユーザーサービスを呼び出してユーザーの詳細情報を追加で取得
5. クーポンサービスのワークフローをトリガーしてクーポンの付与処理を開始する

EventBridge Pipes の主な機能

イベントのフィルタリング
パターンマッチングによるフィルターを追加して、ターゲットに渡す必要のないイベントを除外できる。
除外されたイベントは課金対象外。フィルターでは JSON パスが利用可能。

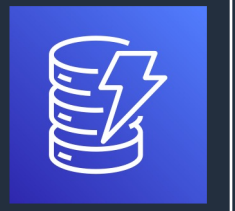
バッチ処理
イベントソースから受信したイベントを、ターゲット/エンリッチメントに対してバッチで送信できる。
バッチの利用可否や最大バッチサイズは後続のサービス仕様に依存する。

順序保証
イベントソースから受信したイベントの順序を維持したまま同期でターゲットを呼び出す。
順次保証がないソースからのイベントの場合は、非同期でターゲットを呼び出す。

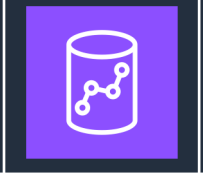
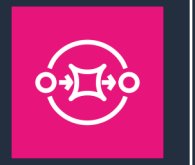
同時実行による高いスループット
バックログに応じて自動でスケールアップ/ダウンを行う。
サービスごとに最大同時実行数は異なる (例: Amazon SQS – 1,250)。

イベントデータの拡張
AWS Lambda, AWS Step Functions, API Gateway, API Destinations を呼び出して追加のデータを取得できる。
各サービスは同期的に呼び出し、最大レスポンスサイズは 6 MB に制限される。

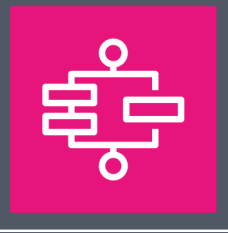
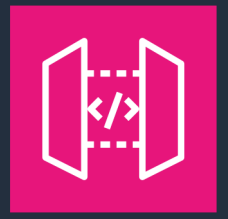
イベントソースに指定できるサービス

	Amazon SQS キュー		Amazon MSK トピック
	Amazon Kinesis Data Streams		Amazon MQ メッセージブローカー
	Amazon DynamoDB Streams		セルフマネージド Apache Kafka ストリーム

ターゲットに指定できるサービス

	Amazon EventBridge • イベントバス • API Destinations		AWS Batch ジョブキュー		Amazon Redshift Data API
	Amazon Simple Notification Service (SNS) * 標準トピックのみ		Amazon Elastic Container Service (ECS) タスク		Amazon SageMaker Pipelines
	Amazon Simple Queue Service (SQS)		AWS Lambda 関数 (同期 or 非同期)		Amazon CloudWatch ロググループ
	AWS Step Functions • 標準: 非同期 • Express: 同期 or 非同期		Amazon Kinesis Data Streams		Amazon Inspector 評価テンプレート
	Amazon API Gateway		Amazon Kinesis Data Firehose		

エンリッチメントに指定できるサービス

	Amazon EventBridge API Destinations
	AWS Step Functions * Express ワークフローのみ
	Amazon API Gateway
	AWS Lambda 関数

- 同期呼び出しのみをサポート
- レスポンスのサイズは最大 6 MB

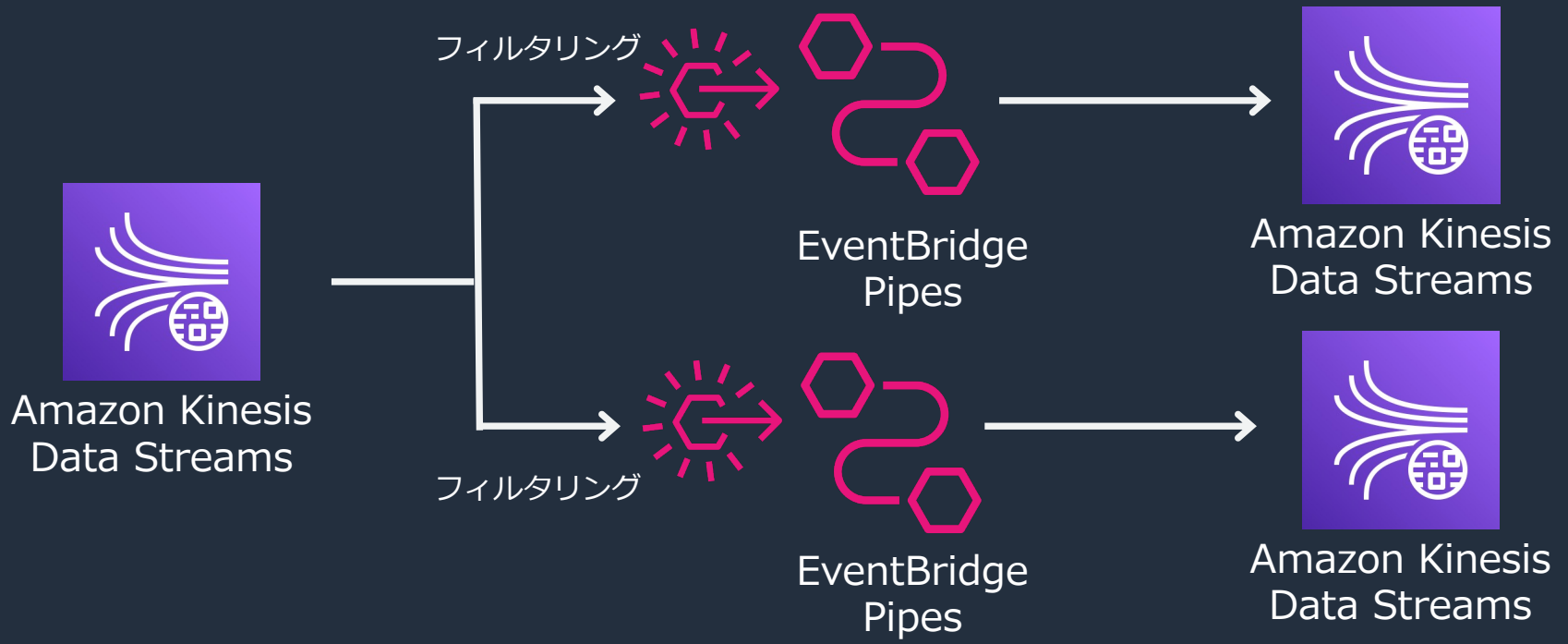
ユースケース 1: SQS → Step Functions

キュー内のメッセージに対して、複雑な一連の処理 (ワークフロー) を実行

Amazon SQS　　EventBridge Pipes　　AWS Step Functions

ユースケース 2: Kinesis Data Streams の分割

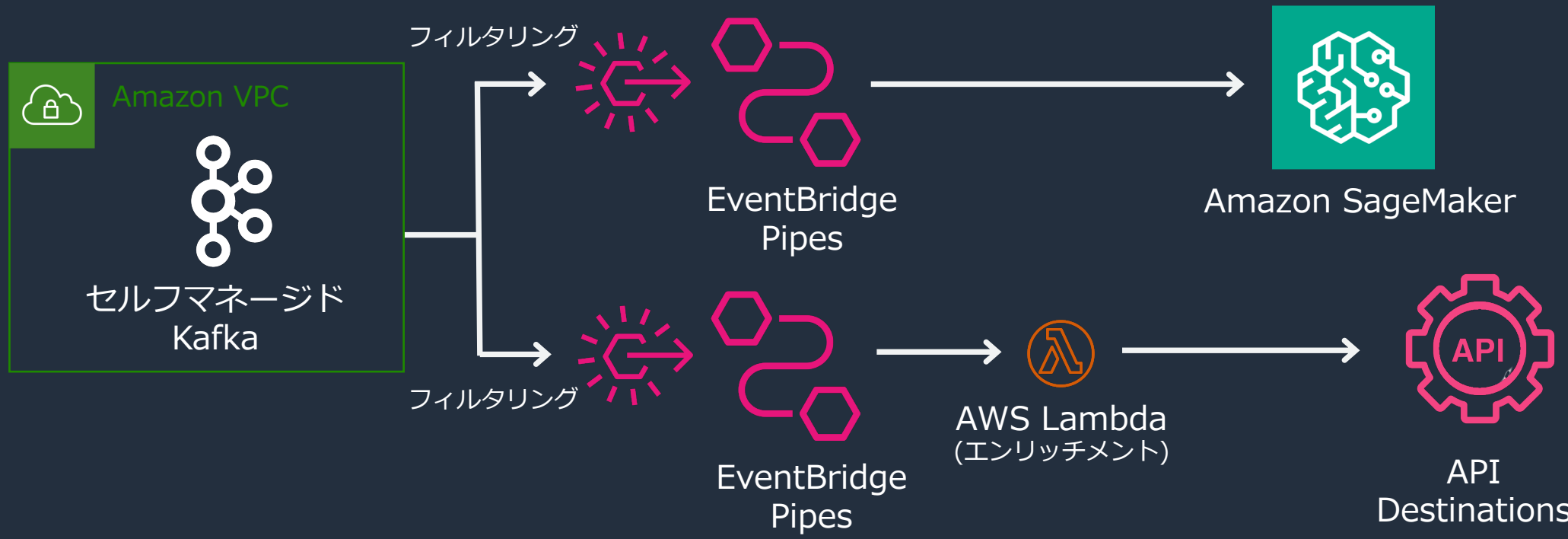
多重化されたデータストリームをドメインに応じて複数のデータストリームに分割



ユースケース 3: Kafka → EventBridge API Destinations

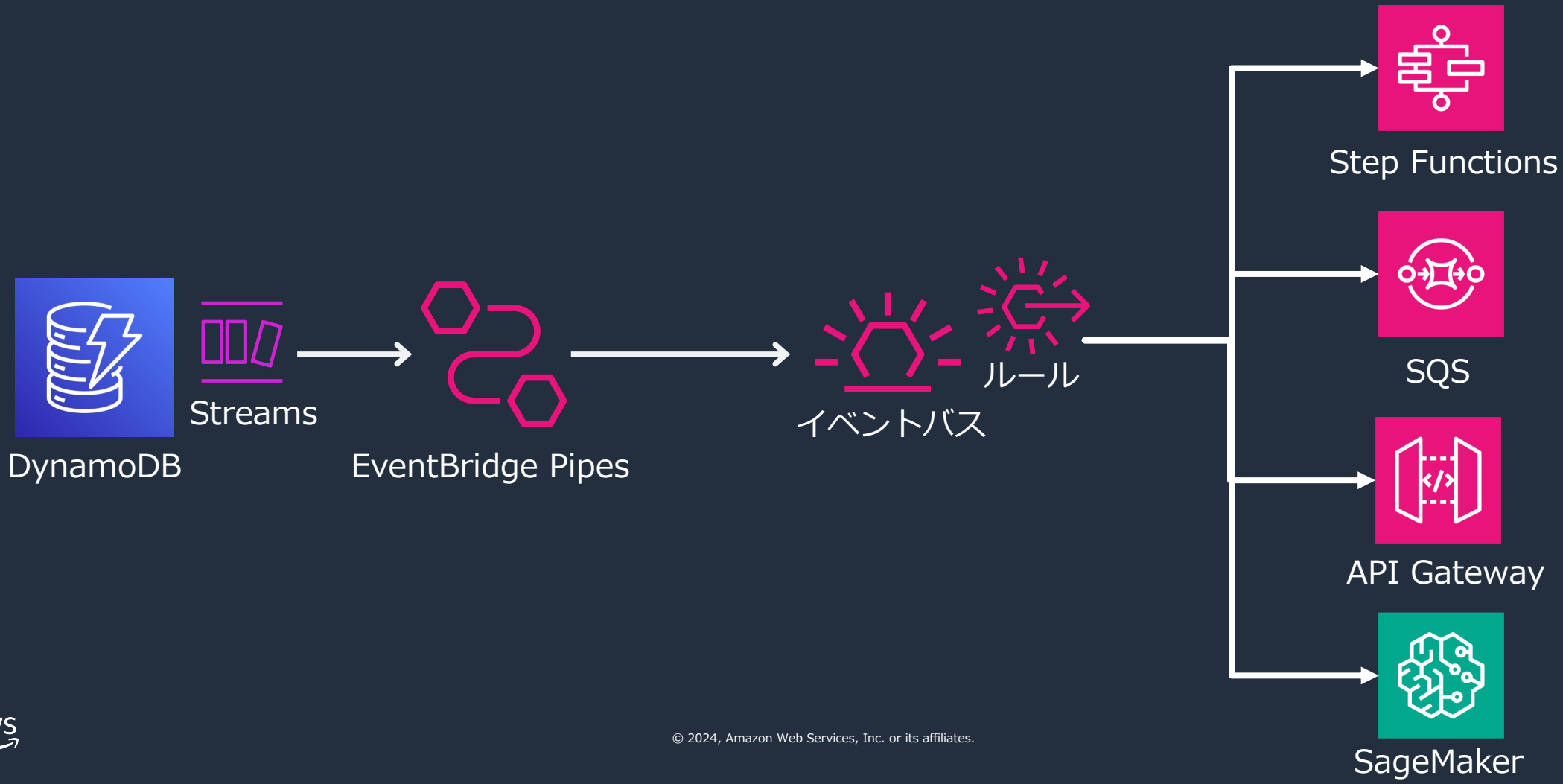
セルフマネージド Kafka クラスターと AWS サービスを接続

コードを書かずに HTTP(s) API を実行



ユースケース 4: DynamoDB → EventBridge

DynamoDB レコードの変更をノーコードで他 AWS サービスに連携



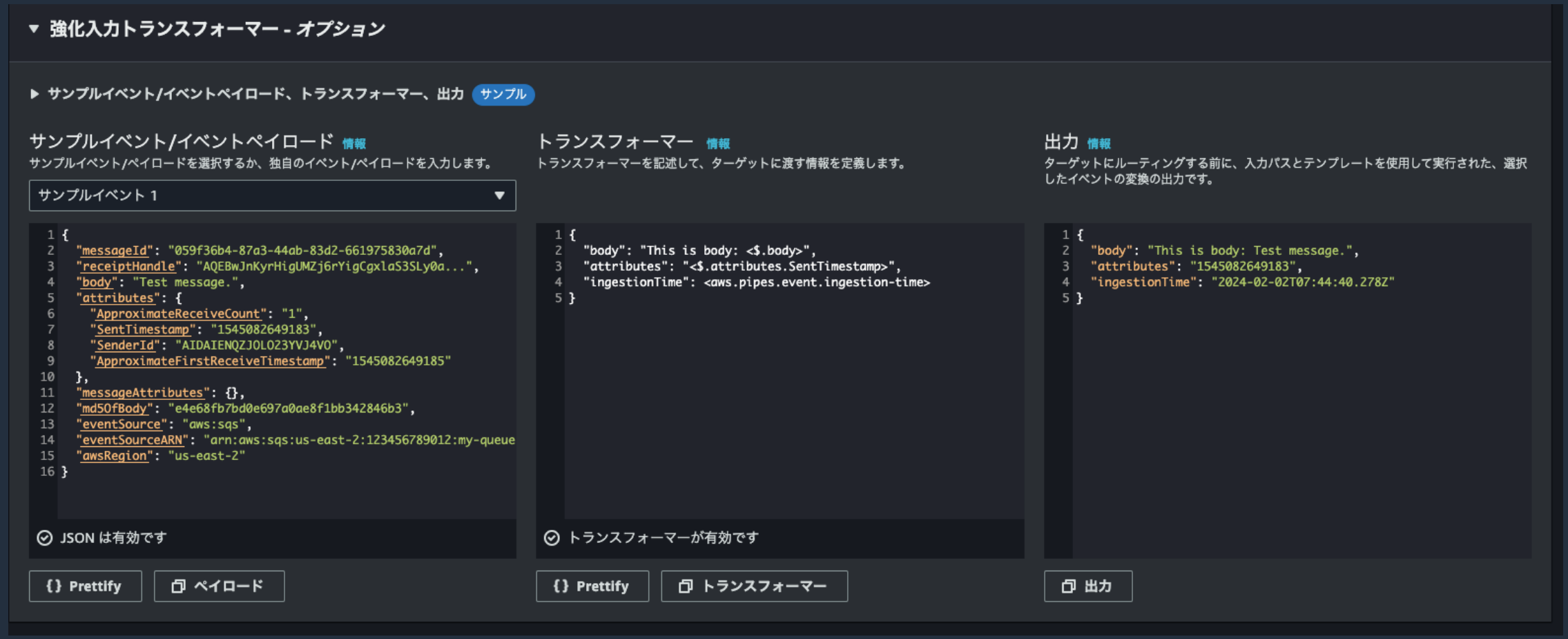
ソースの設定

フィルタリングの設定

参考: [Amazon EventBridge イベントパターンでのコンテンツのフィルタリング]

エンリッチメントの設定

エンリッチメントの設定 – Input Transformer



参考: [Amazon EventBridge Pipes の入力変換]

ターゲットの設定

バッチの設定について (1/2)

- ソースからの取得およびターゲットの呼び出しでバッチをサポート
- 複数イベントが JSON レコードの配列として渡される
- バッチの利用可否や最大バッチサイズは利用する各サービスに依存
- エンリッチメントについては、Lambda および Step Functions でサポート
- エンリッチメントおよびターゲットの呼び出しは、各サービスのバッチ API にマッピングされる (※Lambda や Step Functions の場合は 1 リクエストの入力に JSON 配列全体が渡される)
- エンリッチメントはレスポンスとして JSON の配列を返すようにする

ターゲットごとの最大バッチサイズ

CloudWatch Logs	10,000
EventBridge イベントバス	10
Kinesis Data Firehose ストリーム	500
Kinesis ストリーム	500
Lambda 関数	任意 *
Step Functions ステートマシン	任意 *
Amazon SNS トピック	10
Amazon SQS キュー	10

Lambda および Step Functions については、バッチに含まれるペイロードの合計サイズが制限を超過しないか注意する必要がある

- Lambda: 6 MB
- Step Functions: 262144 Bytes

バッチの設定について (2/2)

- ターゲットがサポートするより大きいバッチサイズを指定することはできない
  - 例：Kinesis Data Streams をソースに、SQS をターゲットにする場合
    Kinesis は最大 10,000 レコードのバッチが設定可能
    SQS は最大 10 メッセージ
    この場合、パイプで設定できる最大バッチサイズは 10
- バッチ内の一部のレコードの処理に失敗した場合の挙動
  - SQS や Kinesis および DynamoDB のようなストリームがソースの場合、失敗したレコードに対して自動的にリトライを行う
  - エンリッチメントでの失敗の場合、部分的なリトライは行われない
  - ターゲットが Lambda または Step Functions の場合、*batchItemFailures* にリトライを行うイベントの ID を含めてレスポンスを返すことで、部分的なリトライの実行が可能

スループットと同時実行について

- STARTED 状態のパイプは、ソースからのイベントを継続的にポーリングし、利用可能なバックログとバッチの設定に応じて自動的にスケーリングを行う
- 単一のパイプは、デフォルトで以下の最大同時実行数までスケールする
  - DynamoDB: ストリームのシャード数 × 設定した *ParallelizationFactor* の数
  - Kinesis: ストリームのシャード数 × 設定した *ParallelizationFactor* の数
  - Apache Kafka: トピックのパーティション数 (最大 1,000)
  - Amazon MQ: 5
  - Amazon SQS: 1250
- これらの制限はベストエフォート型
- パイプの実行時間は最大 5 分 まで (※エンリッチメントおよびターゲットを含む)

実行ロールの設定

- パイプごとに実行 IAM ロールを指定
  Principal: { "Service": "pipes.amazonaws.com" }
- ソースへのアクセス許可を付与
  (例: DynamoDB の場合)
  - dynamodb:DescribeStream
  - dynamodb:GetRecords
  - dynamodb:GetShardIterator
  - dynamodb:ListStreams
- 必要に応じてエンリッチメントおよびターゲットの呼び出し許可を付与

ログの設定 (1/2)

- イベントバッチが処理される際の各実行ステップでログレコードを生成
- ログレベルを選択可能
  - OFF: ログを記録しない (デフォルト)
  - ERROR: パイプ実行中に発生したエラーに関連するレコードを記録
  - INFO: エラー以外の他の実行されたステップも記録
  - TRACE: すべてのステップのすべてのレコードを記録
- ログレベルと記録される実行ステップの対応表はドキュメントを参照

実行ステップ

- *Enrichment Transformation Started*
- *Enrichment Transformation Succeeded*
- *Enrichment Transformation Failed*

- *Target Invocation Started*
- *Target Invocation Succeeded*
- *Target Invocation Failed*
- *Target Invocation Partially Failed*
- *Target Invocation Skipped*

Source

Filtering

Input Transformer

Enrichment

Input Transformer

Target

- *Execution Started*
- *Execution Succeeded*
- *Execution Failed*
- *Execution Partially Failed*
- *Execution Throttled*
- *Execution Timeout*

- *Enrichment Stage Entered*
- *Enrichment Stage Succeeded*
- *Enrichment Stage Failed*

- *Enrichment Invocation Started*
- *Enrichment Invocation Succeeded*
- *Enrichment Invocation Failed*
- *Enrichment Invocation Skipped*

- *Target Transformation Started*
- *Target Transformation Succeeded*
- *Target Transformation Failed*

- *Target Stage Entered*
- *Target Stage Succeeded*
- *Target Stage Failed*
- *Target Stage Partially Failed*
- *Target Stage Skipped*

詳細は[ドキュメント]を参照

ログの設定 (2/2)

- ログの配信先として、Amazon CloudWatch, Amazon Kinesis Data Firehose, Amazon S3 をサポート (複数選択可)
- CloudWatch が選択されている場合、ログレコードの最大サイズは 256 KB に制限され、フィールドごとに自動的に切り捨てられる
- 実行データをログに含めるか選択可能
  - 実行データには、イベントバッチペイロード、エンリッチメントおよびターゲットのリクエスト/レスポンスが含まれる
  - 機密データがログに含まれる可能性がある点に注意

料金

- パイプが処理するリクエスト数に応じて課金
- 100 万リクエストごとに $0.50 (*2024 年 3 月時点/東京リージョン)
- フィルタリングで除外されたイベントは課金対象外
- ペイロードサイズ 64 KB を上限として 1 リクエストとして扱う

例：バッチで合計 256 KB のイベントを処理する場合

256 KB / 64 KB = 4 リクエスト となる

クォータ

- アカウント全体で作成できるパイプの上限: 1,000
- アカウント全体でのパイプの最大同時実行数:
  - バージニア、オレゴン、アイルランドリージョン: 3,000
  - 東京/大阪を含むその他のリージョン: 1,000
- いずれも上限緩和申請が可能

まとめ

- EventBridge Pipes は、イベントプロデューサーとイベントコンシューマーを直接 point-to-point で連携する機能
- フィルタリングやバッチ、順序保証、HTTP API の呼び出しによるペイロードの変換、リトライなど豊富な機能をサポート
- 連携のためのコード (glue code) やインフラストラクチャの管理のための時間を削減し、よりビジネス的な価値に集中することができる
- 既存のイベントバスは引き続き利用可能。Pipes が提供するリッチな機能が必要ない場合は無理に移行する必要はない

Thank you!