

JAPAN | JUNE 20, 2024

aws SUMMIT



DOL-6

オブザーバビリティにおける AI/ML 活用の現状と展望

Mitsuaki Tsugo

Solutions Architect

Amazon Web Service Japan G.K.



自己紹介

津郷 光明

アマゾンウェブサービスジャパン
ソリューションアーキテクト

製造業のお客様を担当

好きな技術分野：

Observability / Infrastructure as Code



Observability helps you



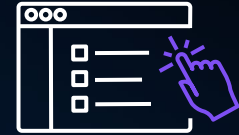
サービス状態
(正常 / 異常) の把握



パフォーマンスと
可用性の向上



運用コストの
削減



顧客体験の
向上

System

Business

Booking.com

ウェブサイトのリアルユーザー
パフォーマンスを収集・測定

 **CloudPassage**

潜在的なバグの発見と修正を高速化、
機能開発が加速

 **mapbox**

サードパーティシステムのセットアップ、
構成、学習の運用負荷を軽減

rego consulting

顧客からの通知前に問題を解決

Agenda

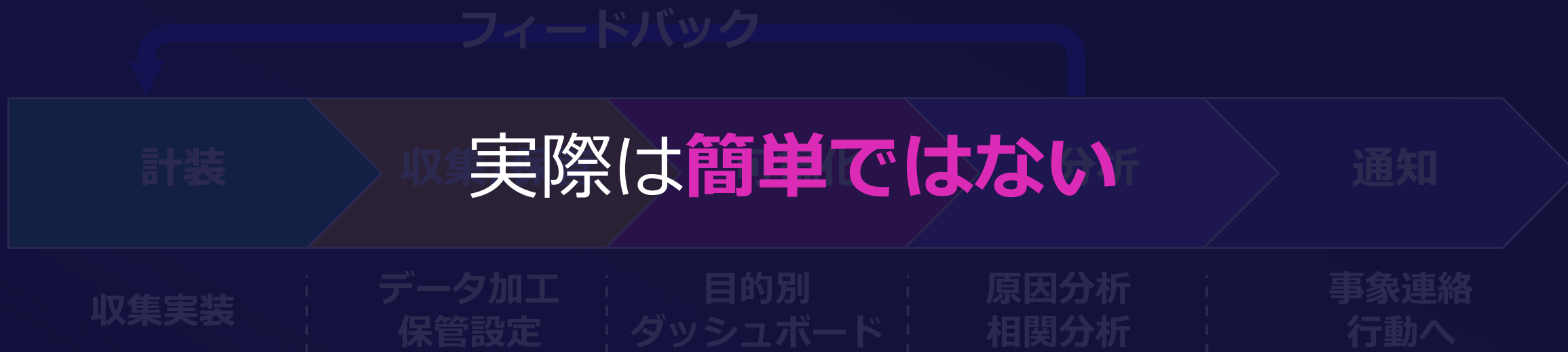
- オブザーバビリティのプロセスと課題
- オブザーバビリティにおける AI/ML の活用
- オブザーバビリティにおける AI/ML 活用の現状と展望
- まとめ

オブザーバビリティの プロセスと課題

オブザーバビリティのプロセス



オブザーバビリティのプロセス



多くのお客様が直面している課題

※一部抜粋



多くのお客様が直面している課題

※一部抜粋



技術の多様化

- コンテナ、Kubernetes、API連携など新しい技術の登場
- 学習コスト、実装コスト
- 管理 / 統制の複雑化

多くのお客様が直面している課題

※一部抜粋



ステークホルダ多様化
システム変化の高速化

- データが不十分
- 頻繁なシステム状態変化への対応
- 分析の多角化
- 非技術者の閲覧 / 見たいポイントのズレ

Observability

Open-source managed services



Amazon Managed Grafana

Application insights

Do it yourself (DIY) – AWS OSS Solutions



RUM

オブザーバビリティ ×



AWS X-Ray



Amazon OpenSearch Service



Amazon Managed Service for Prometheus



JAEGER



ZIPKIN

Jaeger and Zipkin Tracing



AWS Distro for OpenTelemetry

Collectors and SDKs

Instrumentation

医療 AI	Amazon HealthLake	Amazon Transcribe Medical	Amazon Comprehend Medical	AWS Panorama + Appliance	Amazon Monitron	Amazon for Eq
画像・動画	Amazon Rekognition	Amazon Polly	Amazon Transcribe	Amazon Transcribe Streaming	Amazon Translate	Amazon Textract
音声	Amazon Rekognition	Amazon Polly	Amazon Transcribe	Amazon Transcribe Streaming	Amazon Translate	Amazon Textract
テキスト	Amazon Rekognition	Amazon Polly	Amazon Transcribe	Amazon Transcribe Streaming	Amazon Translate	Amazon Textract
検索	Amazon Rekognition	Amazon Polly	Amazon Transcribe	Amazon Transcribe Streaming	Amazon Translate	Amazon Textract
機械学	Amazon SageMaker	ラベリング	準備	特徴量ストア	Auto ML	Spark/R
フレームワーク	TensorFlow	mxnet	PyTorch	GLUON	learn	Keras
						DeepGraphLibrary

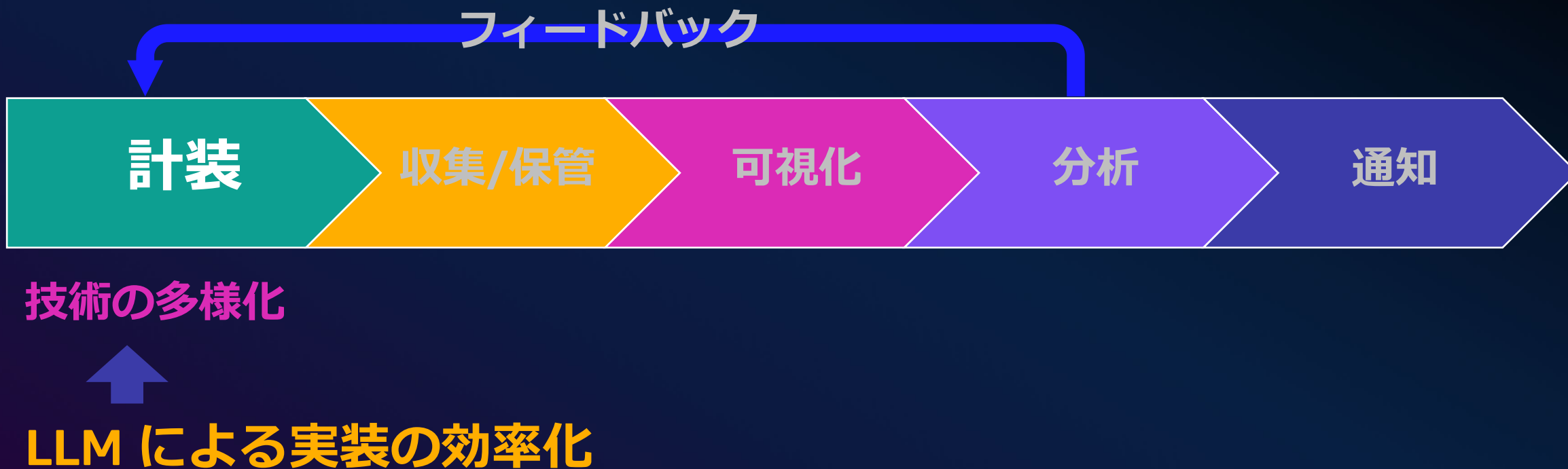
機械学習

AI / ML



オブザーバビリティにおける AI/MLの活用

AI/ML 活用による解決例



計装の課題

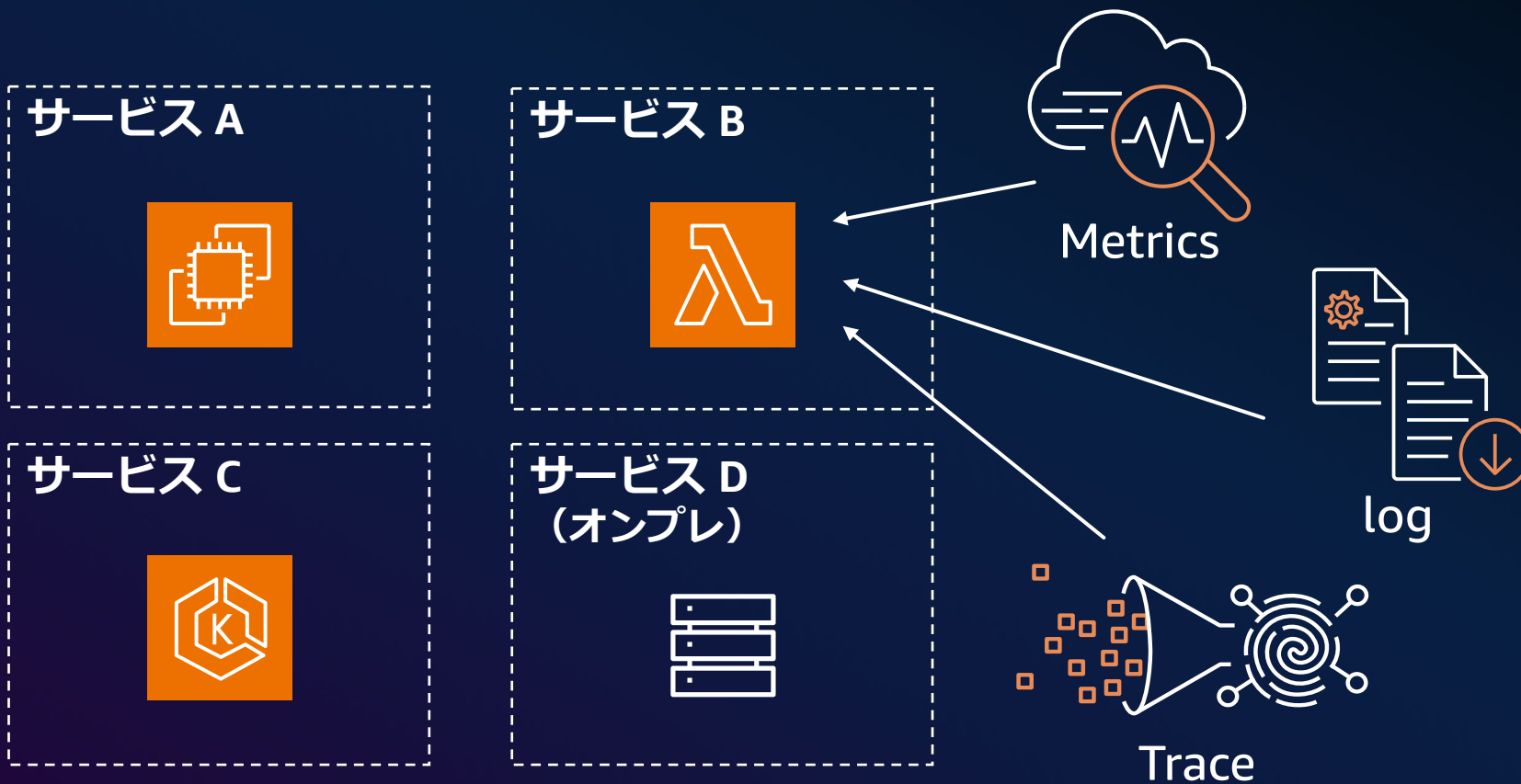
作業の多さ
設計観点の多さ

×

新しい技術
技術の混在

=

学習 / 実装 / 管理 / 運用
コストが高い



AI/ML を利用した計装の課題への対応

生成 AI (LLM)

×

Observability as Code



AWS Lambda の計装の例

ログ

Python

```
my_lambda = _lambda.Function(self, 'HelloHandler',
runtime = _lambda.Runtime.PYTHON_3_8,
code = _lambda.Code.from_asset('lambda'),
handler = 'hello.handler',
environment = {
'AWS_LAMBDA_EXEC_WRAPPER': '/opt/otel-instrument',
},
adot_instrumentation=_lambda.AdotInstrumentationConfig(
layer_version=AdotLayerVersion.from_python_sdk_layer_version(
AdotLambdaLayerPythonSdkVersion.LATEST),
exec_wrapper=AdotLambdaExecWrapper.INSTRUMENT_HANDLER
),
insights_version=_lambda.LambdaInsightsVersion.VERSION_1_0_9
8_0,
)
```

トレース

メトリクス

学習、実装、管理、運用コストを削減

参考：モダンアプリケーション開発におけるオブザーバビリティの実装 ~ Observability as Code ~

<https://resources.awscloud.com/aws-modern-applications-innovate-jp/implementing-observability-in-modern-application-development-aws-innovate-modern-apps-jp>



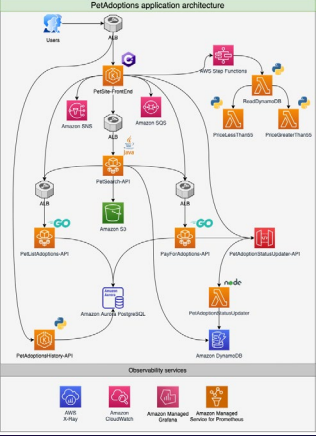
Amazon Bedrock を利用した Observability as Code の推進

Input data

LLM

Observability as Code

アーキテクチャ



各種要件および
パラメータ

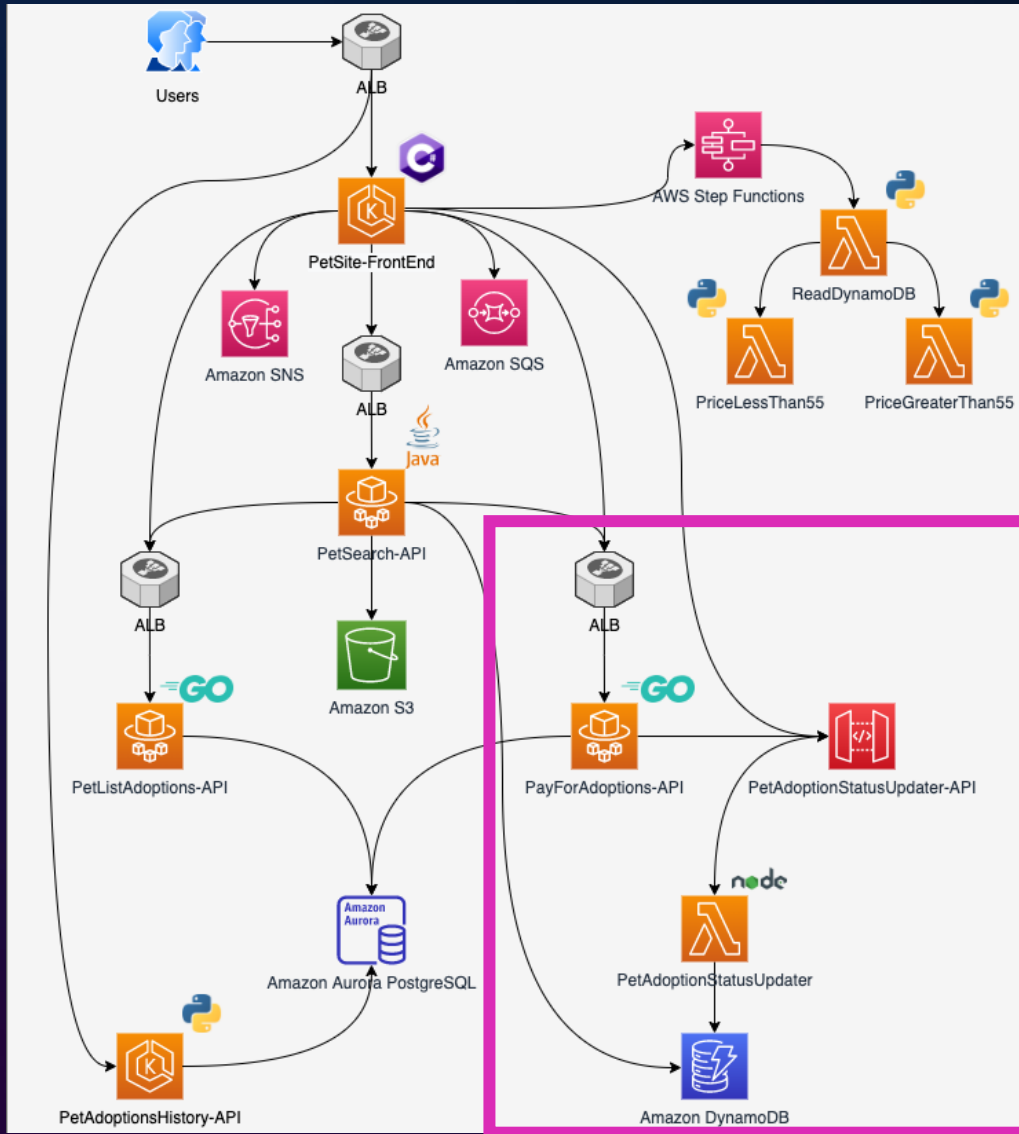
チャットアプリケーション

Amazon Bedrock
Claude3 sonnet



AWS Cloud Development
Kit (AWS CDK)

利用するアーキテクチャ



One Observability Workshop

<https://catalog.us-east-1.prod.workshops.aws/workshops/31676d37-bbe9-4992-9cd1-ceed13c5116c/ja-JP>

ここを生成

Amazon Bedrock を利用した Observability as Code の推進



生成されたコード (抜粋)



AWS CDK

TypeScript

```
// Lambda Function
const petAdoptionStatusUpdater = new lambda.Function(this, 'PetAdoptionStatusUpdater', {
  runtime: lambda.Runtime.GO_1_X,
  code: lambda.Code.fromAsset('path/to/lambda/code'),
  handler: 'main',
  environment: {DYNAMODB_TABLE_NAME: petAdoptionStatusTable.tableName,}, });
// API Gateway
const api = new apigw.RestApi(this, 'PetAdoptionStatusUpdaterAPI');
const petAdoptionStatusUpdaterIntegration = new apigw.LambdaIntegration(petAdoptionStatusUpdater);
api.root.addResource('update').addMethod('POST', petAdoptionStatusUpdaterIntegration);
// Log Groups
const logGroup = new lambda.LogGroup(this, 'PetAdoptionStatusUpdaterLogGroup', {
  logGroupName: `aws/lambda/${petAdoptionStatusUpdater.functionName}`,
  removalPolicy: cdk.RemovalPolicy.DESTROY,
});
// Metrics
petAdoptionStatusUpdater.metricErrors().with({
  label: 'LambdaErrors',
}).createAlarm(this, 'PetAdoptionStatusUpdaterErrorAlarm', {
  alarmDescription: 'Errors in PetAdoptionStatusUpdater Lambda',
  evaluationPeriods: 1,
  threshold: 1,
});
// X-Ray Tracing
const tracing = new lambda.Tracing(this, 'PetAdoptionStatusUpdaterTracing', {
  mode: lambda.TracingMode.ACTIVE,
});
```

実行時はエラー

- プロンプトの改善
- 詳細情報の追加
で解消が見込まれる

AI/ML 活用による解決例



ステークホルダ多様化
システム変化の高速化

- ダッシュボード自動生成
- アノテーション自動生成
- 可視化方法の推奨・自動生成
- 自然言語で可視化クエリ
- 予測モデルによる将来予測
- 根本原因・因果関係の自動分析

可視化・分析の課題

従来



シニア
インフラエンジニア



現在



シニア
インフラ
エンジニア

+



ビジネスメンバー



ジュニアエンジニア

多角的に可視化・分析できる状態が
求められ実装が複雑化

自然言語でデータ分析クエリ

CloudWatch Logs Insight と Metrics Insights では、生成 AI により 自然言語でクエリを生成

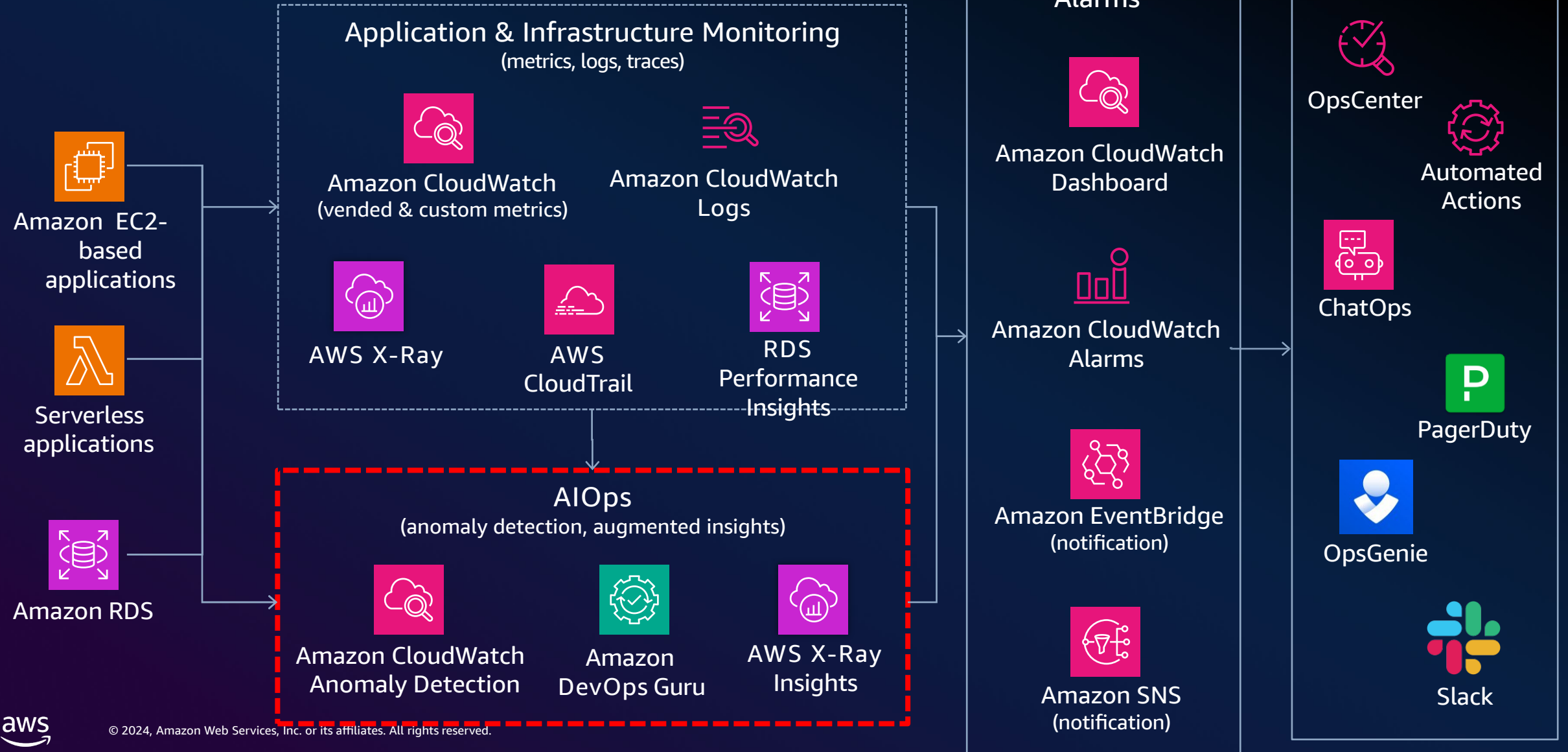
- 自然言語の質問からクエリ生成
- 生成したクエリの解説も付与
クエリ構文の学習を支援
- 自然言語を利用して **既存クエリを改良**
繰り返し実行するクエリの継続的な改善に有益
- 英語のみサポート

専門知識がなくても分析が可能に

The screenshot displays the AWS CloudWatch Logs Insights console. At the top, there's a navigation bar with 'CloudWatch > ログのインサイト'. Below this, the 'ログのインサイト' section is active, showing a 'テリングを開始' button. A time range selector is set to '3時間'. Below the time range, there's a dropdown for 'ロググループ' with a '参照' button. A search bar contains '/aws/lambda-insights' and a 'すべてクリア' button. A 'Query generator - new' button is visible. Below the search bar, there are buttons for 'クエリの実行', 'キャンセル', '保存', and '履歴'. A message states 'Logs Insights クエリは、最大 60 分まで実行できます。'. At the bottom, there's a 'Logs (-)' section with a '結果をエクスポート' button and a 'ダッシュボードに追加' button. The status shows '0 の 0 の一致したレコードの表示' and '1,204 レコード (1.3 MB) が 2.9s @ 412 records/s (455.6 KB/s) でスキャンされました'.

Monitoring with AIOps in AWS

アプリケーションの異常を自動で検知

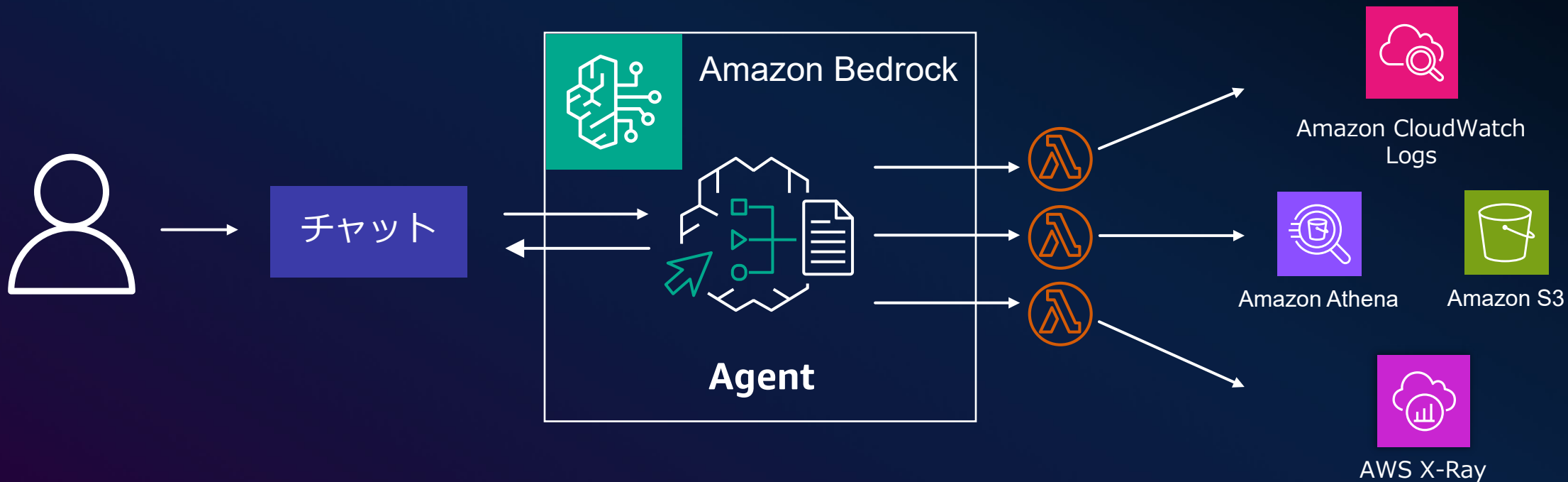


Agents for Amazon Bedrock を利用したチャットでの障害原因の特定

User

データ収集と要約

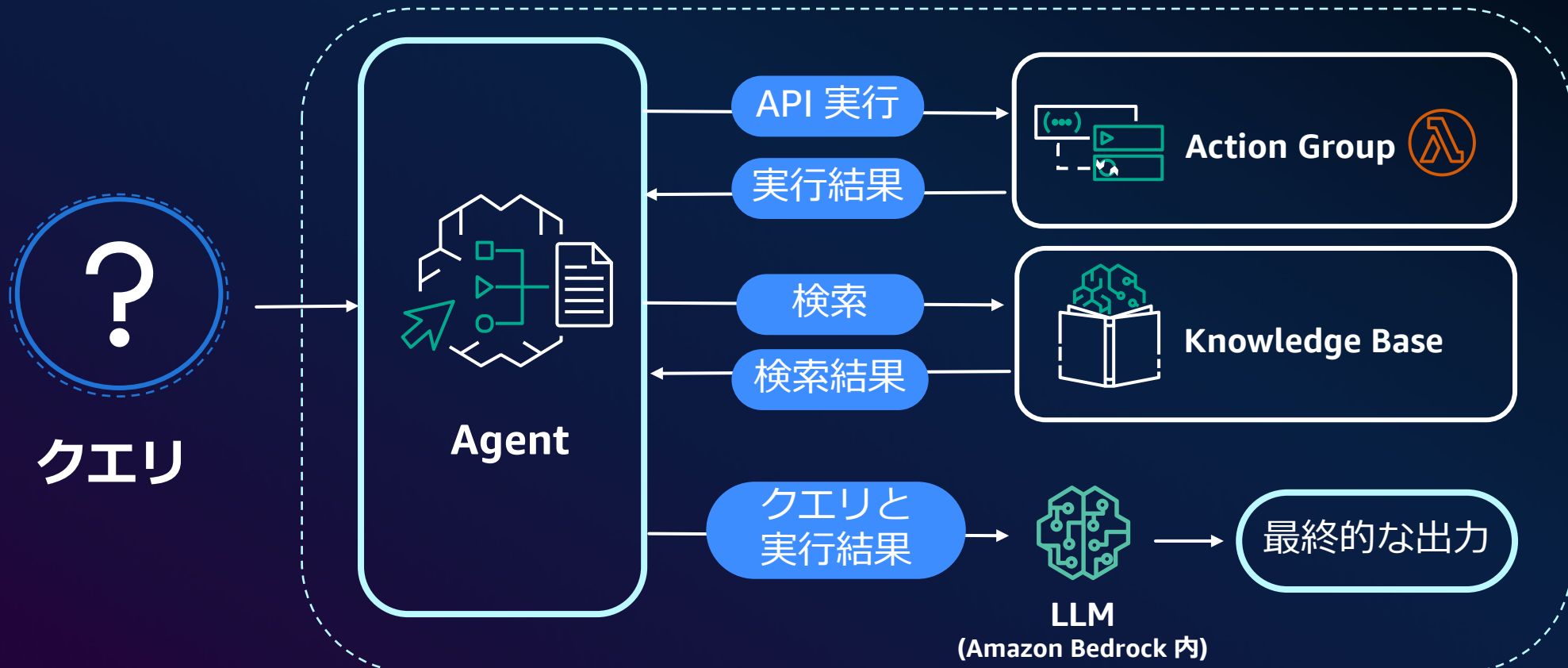
分散管理された
テレメトリデータ



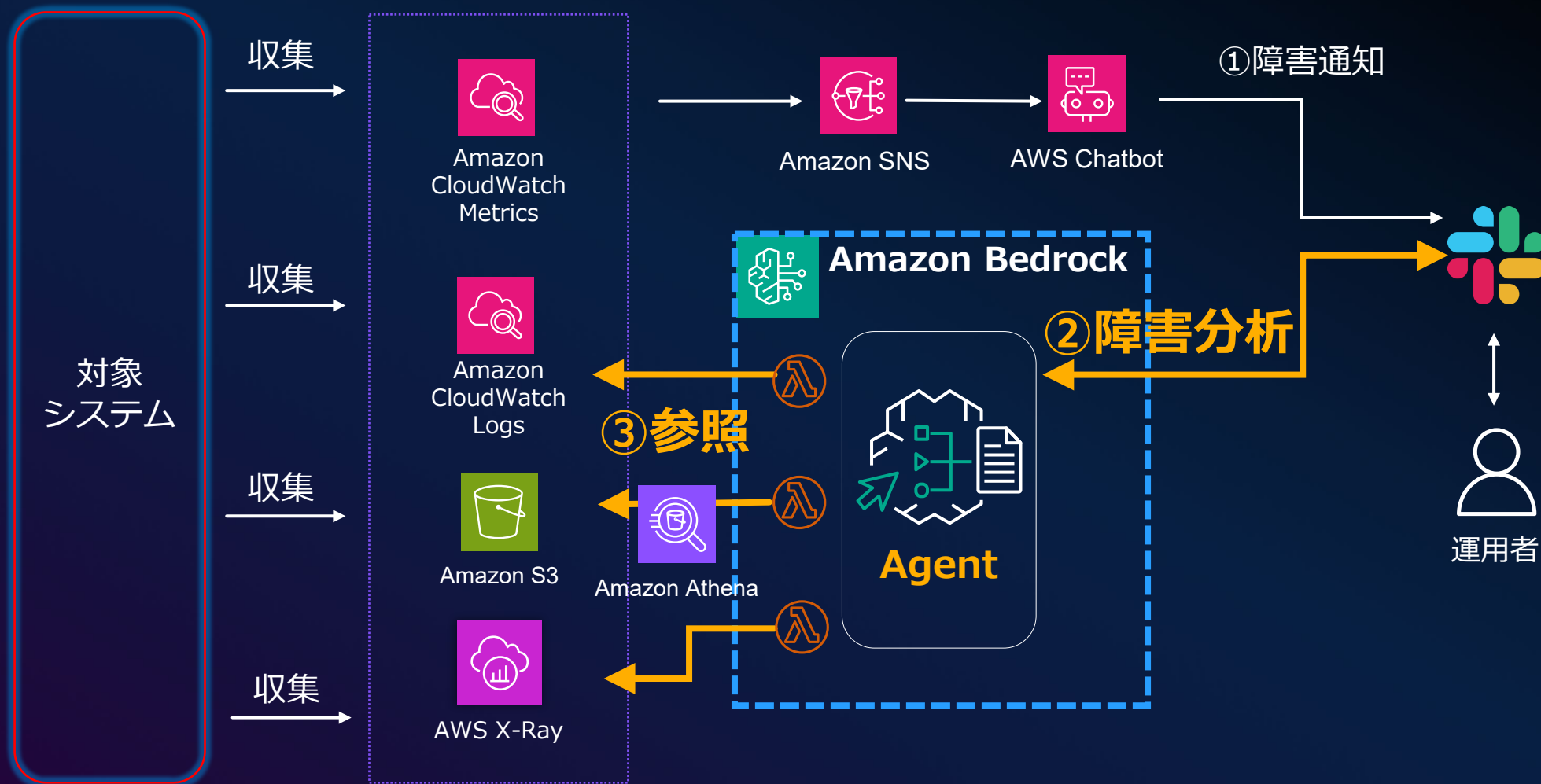
Agents for Amazon Bedrock

API を呼び出しタスクを実行する Agent 機能をフルマネージドで提供

基盤モデルを使ってユーザのクエリを理解し、登録された情報を Knowledge Base から検索したり、タスク完了に必要なアクションを実行



Agents for Amazon Bedrock を利用したチャットでの障害原因の特定





aws APP 12:54 PM

Today ▾

 **CloudWatch Alarm | Dev-TargetMonitoring-CanaryCanaryDurationAlarm9CB7D45A-yO8fuMu7WQTG | us-east-1 | Account: 654654344320**

Threshold Crossed: 2 out of the last 2 datapoints [10098.0 (18/06/24 03:53:00), 50118.0 (18/06/24 03:52:00)] were greater than or equal to the threshold (400.0) (minimum 2 datapoints for OK -> ALARM transition).

List dashboards

Query logs



Namespace

CloudWatchSynthetics

Metric

Duration

Timestamp

Tue, 18 Jun 2024 03:54:50 UTC

CanaryName

devtargetmonito76b313

[See more](#)



RCA Bot APP 12:54 PM

アラームが発生したようです。エラーの原因分析をしたい場合は、ログ検索を行う時刻の範囲を以下のフォームから入力してください。アラームのDatapointを参考に入力いただくと、比較的良い結果が得られやすいです。

エラーの通知（アラーム）の内容

例：外形監視のアラームで、エラー回数が規定以上になっています。

ログ取得の開始日

 Today ▾

ログ取得の開始時刻

 12:54 PM ▾

ログ取得の終了日

 Today ▾

提供：
Prototyping SA 鈴木 陽三

生成 AI, AWS for Data & AWS for Every Application Zone 展示中



Prototyping SA
鈴木 陽三



オブザーバビリティにおける AI/ML 活用の現状と展望

オブザーバビリティにおける AI/ML 活用との付き合い方

将来機械学習によってオブザーバビリティが飛躍的に高度化される

一方で課題も残る

- 機械学習全般での誤検知の可能性
- 生成AIでのハルシネーションのリスク など



- ✓ 人の判断をサポートする位置付けで利用していくことが妥当
- ✓ 可視化や分析の高度化に向けてはテレメトリデータの取得から

まとめ

まとめ

- 現時点では人をサポートする位置付けでの利用が現実的
- AI/MLによりオブザーバビリティを効率化、高度化されることが予想される
- 可視化や分析の高度化に向けてテレメトリデータを収集しておく

Thank you!

Mitsuaki Tsugo

tsugom@amazon.co.jp

