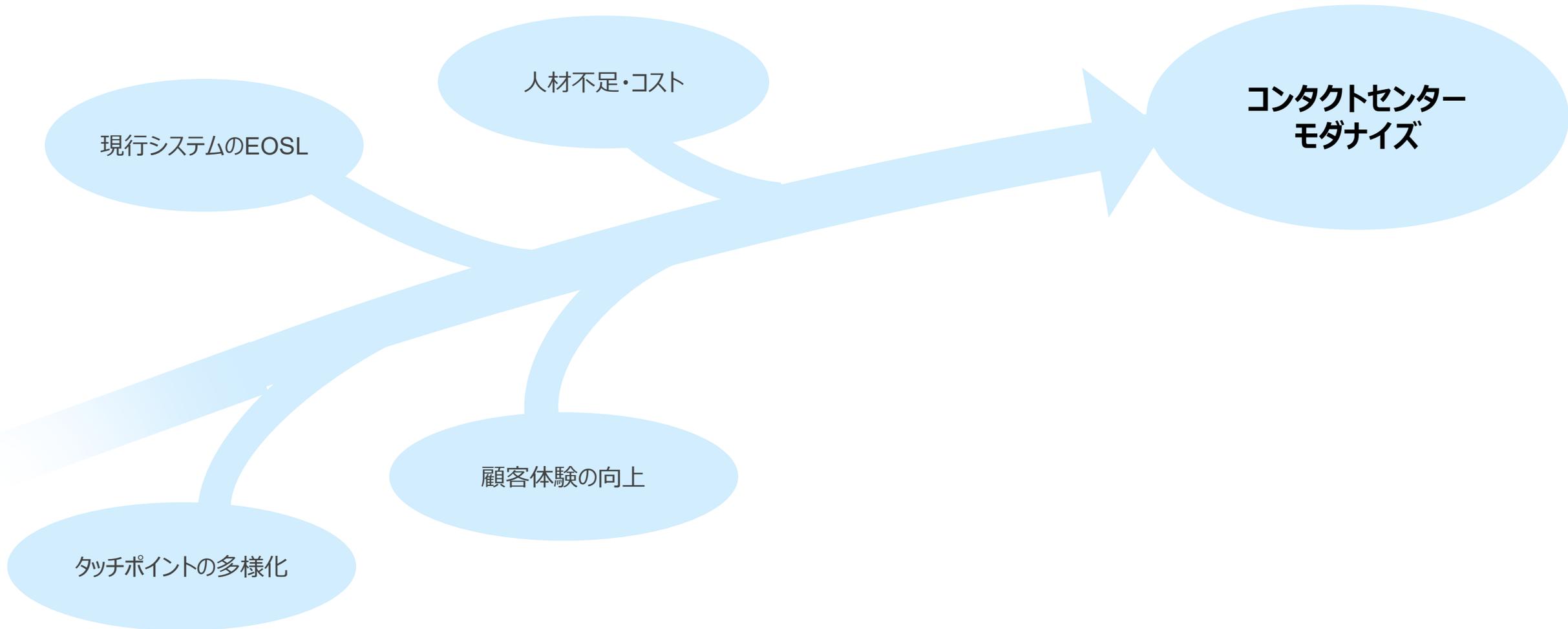


NTT DATA

# Amazon Connect コンタクトセンター導入のポイント

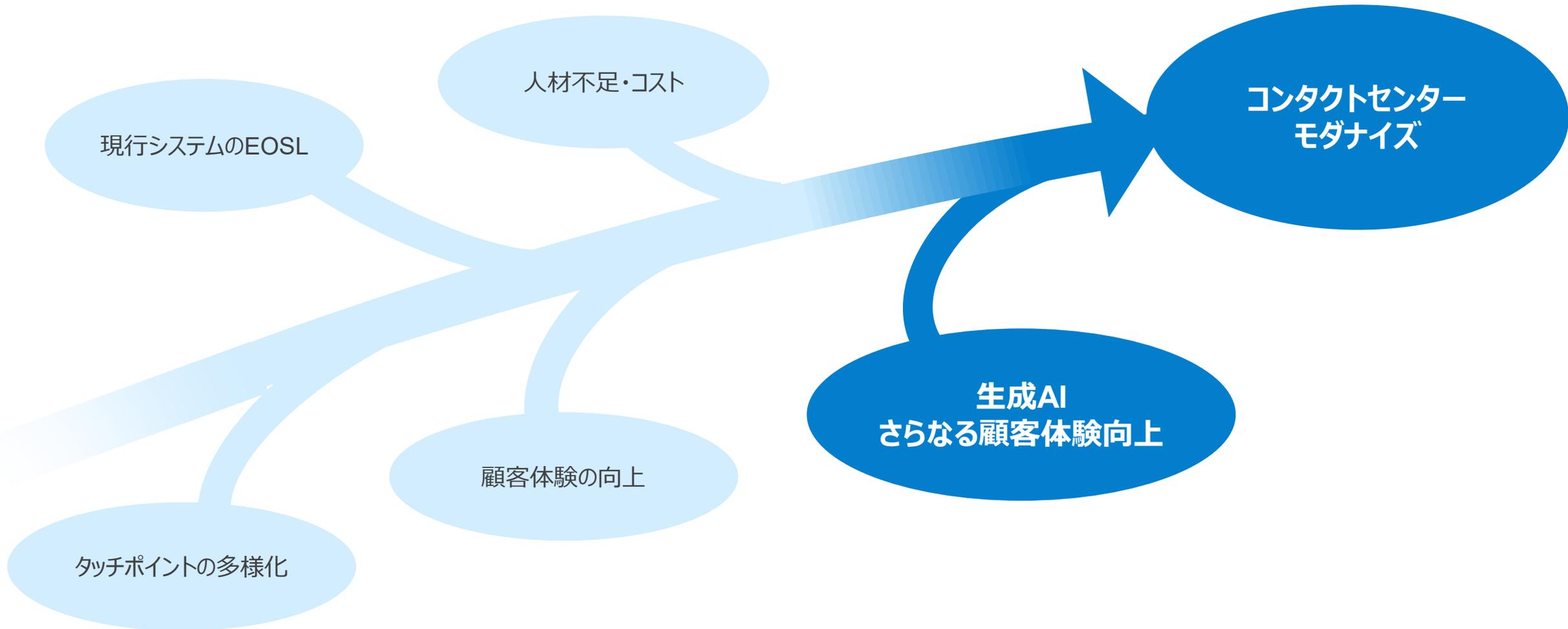
# ■ コンタクトセンターのモダナイズを取り囲む環境

コンタクトセンターモダナイズに取り組むモチベーションは様々



# コンタクトセンターのモダナイズを取り囲む環境

さらなるモダナイズモチベーション『生成AI』が到来



# Amazon Connectによるコンタクトセンターモダナイズ

Amazon Connectはモダンなコンタクトセンターをスモールスタート可能

ワークフローデザイナー  
ノーコード・ローコード  
IVR

オムニチャネル

生成AIとの  
連携・統合

エージェント  
ワークスペースによる  
オンボーディング

Amazon Connect  
Contact Lensによる  
会話分析

# Amazon Connectによるコンタクトセンターモダナイズ

Amazon Connectであっても本格導入・運用・活用には課題に直面することも

ワークフローデザイナー  
ノーコード・ローコード  
IVR

オムニチャネル

生成AIとの  
連携・統合

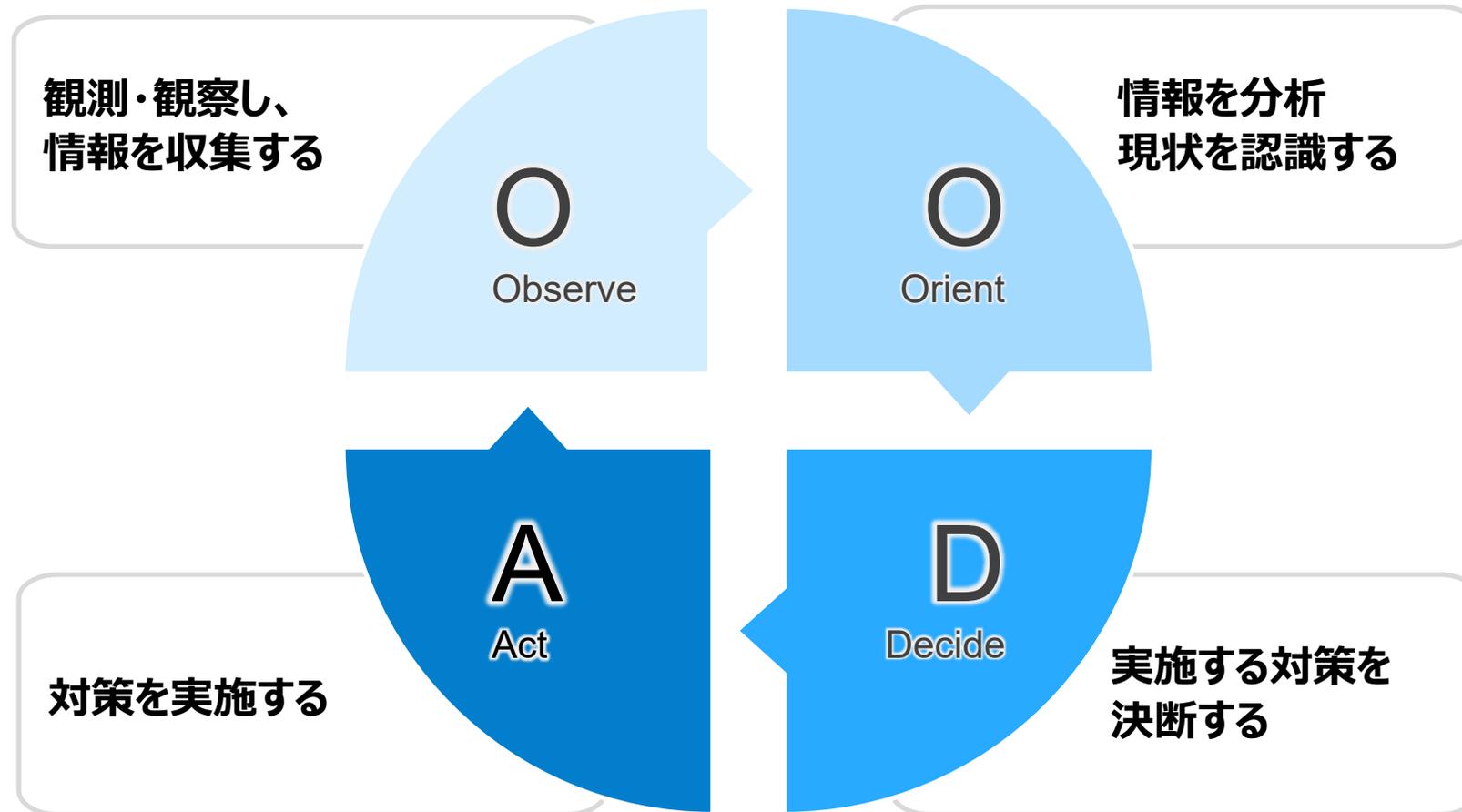
エージェント  
ワークスペースによる  
オンボーディング

Amazon Connect  
Contact Lensによる  
会話分析

**本格導入  
さらなる活用**

# 課題を乗り越えるためのアプローチ OODA

複雑・変化の速い課題を乗り越えるフレームワークの1つOODA



# 課題を乗り越えるためのアプローチ OODA

ObserveとActを実行しやすいシステムであることが重要

観測・観察し、  
情報を収集する

O  
Observe

情報を分析  
現状を認識する

O  
Orient

情報収集・対策の速度と精度は  
課題を乗り越える上で重要

対策を実施する

A  
Act

OrientとDecideも重要だが、  
経営及び業務要素の影響が大きい

実施する対策を  
決断する

D  
Decide

## 本日のアジェンダ | ObserveとActを実行しやすくするためのポイントを少しご紹介

01

メトリクス活用を見越したキューの設計

Observe

02

ノーコード・ローコードにおけるFeature Toggle

Act

# 本日のアジェンダ

01

メトリクス活用を見越したキューの設計

Observe

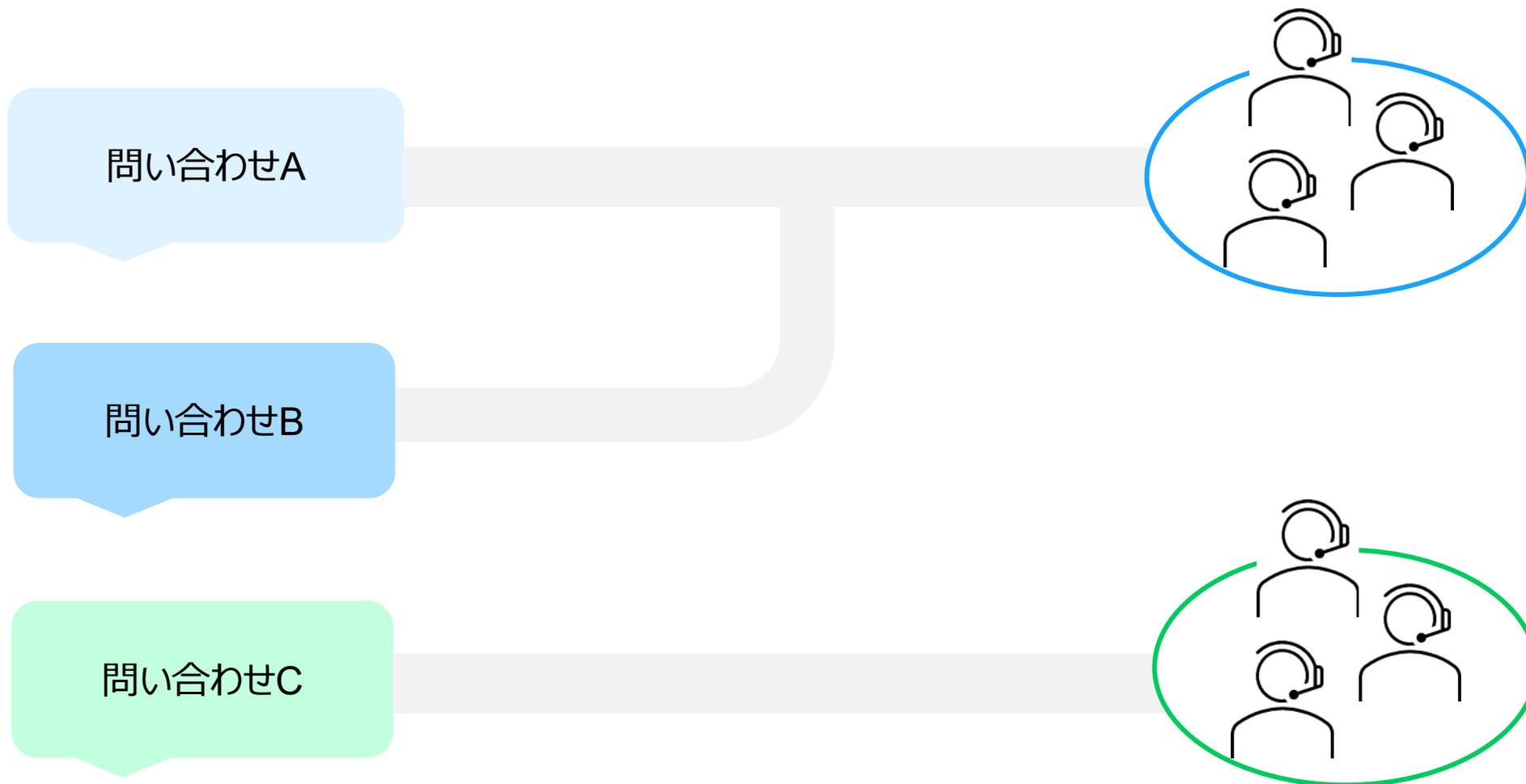
02

ノーコード・ローコードにおけるFeature Toggle

Act

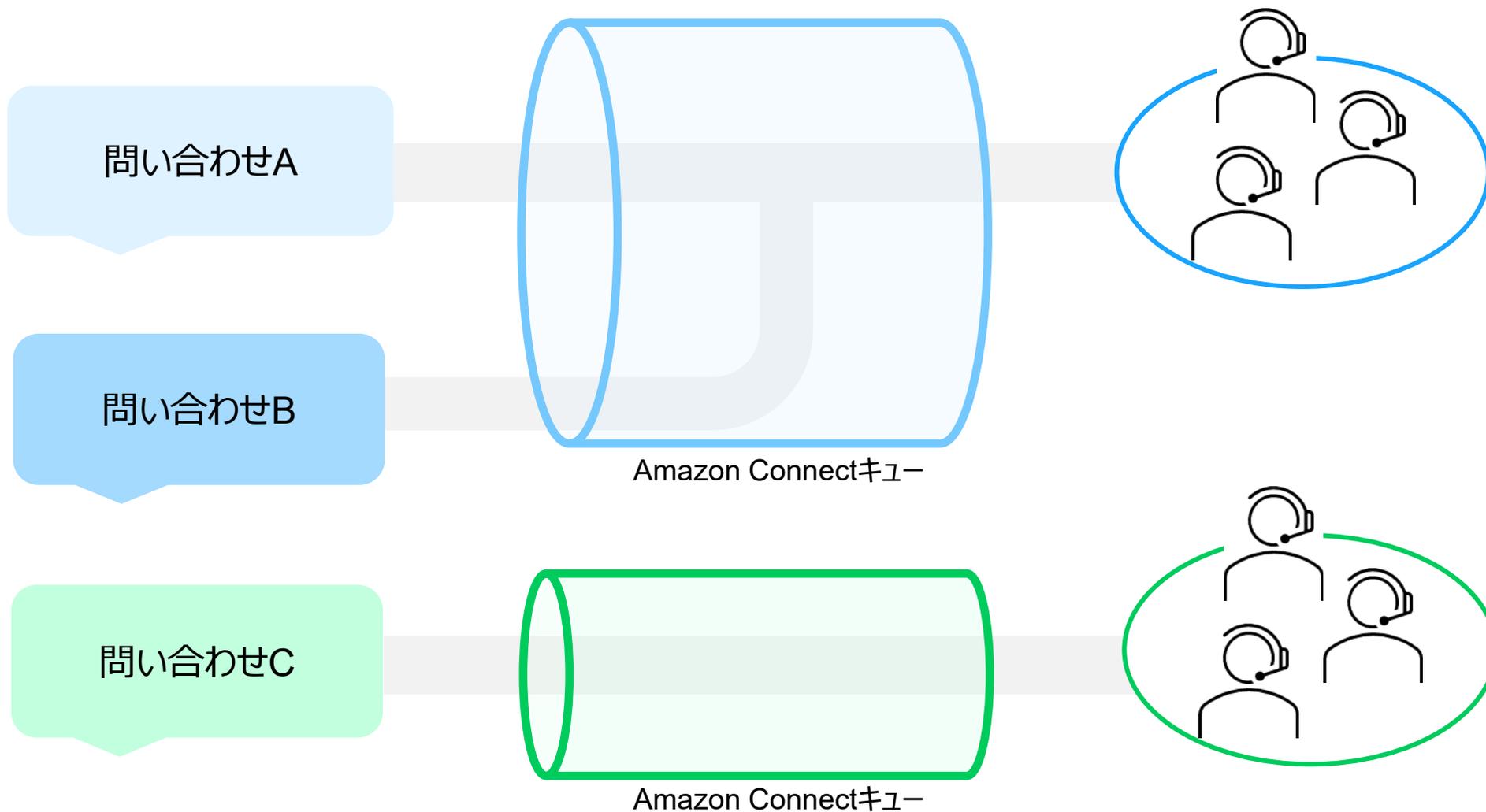
# メトリクス活用を見越したキューの設計

あるエージェントグループが複数種類の問い合わせに対応する場合



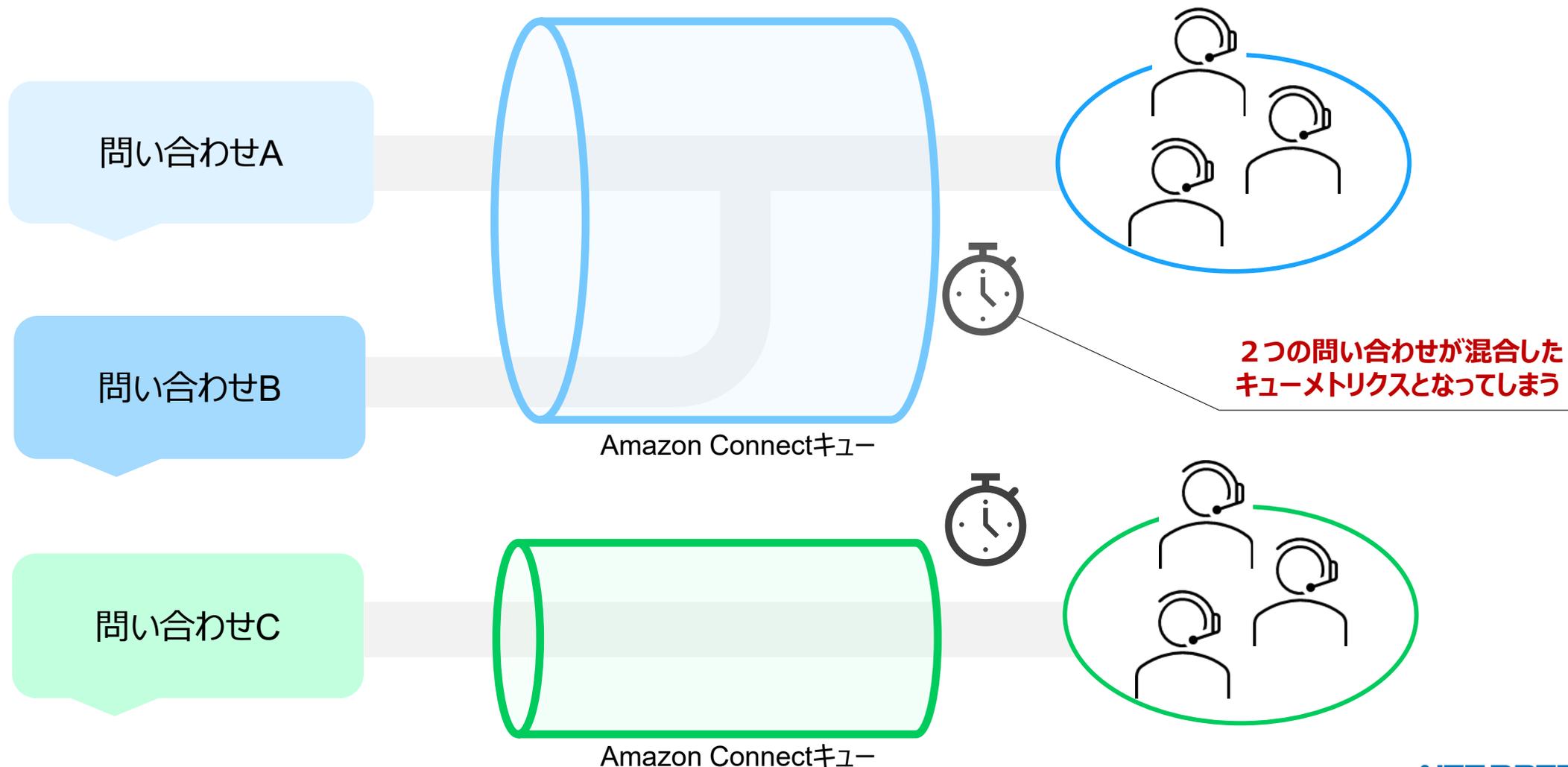
# メトリクス活用を見越したキューの設計

実装上は1つのキューで複数種の間い合わせを受け付けることも可能だが



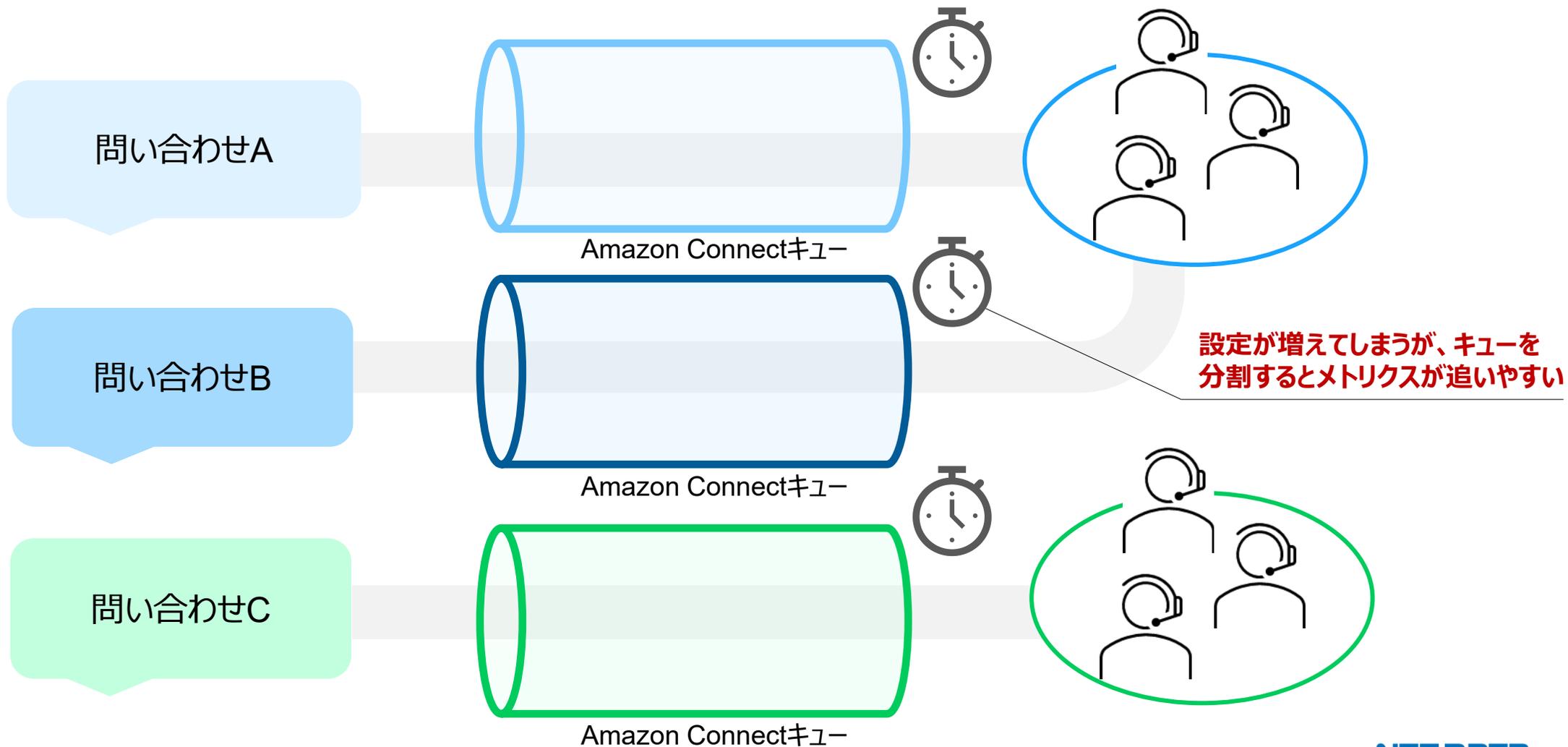
# メトリクス活用を見越したキューの設計

複数の問い合わせが混ざるため、キューメトリクスの観測性が低下する



# メトリクス活用を見越したキューの設計

各問い合わせ種別ごとのメトリクスをクイックに見たい場合はあえてキューを分けるのも選択肢



# 本日のアジェンダ

01

メトリクス活用を見越したキューの設計

Observe

ルーティングだけでなく、計測と改善を見越したキュー設計が望ましい。

02

ノーコード・ローコードにおけるFeature Toggle

Act

# 本日のアジェンダ

01

メトリクス活用を見越したキューの設計

Observe

ルーティングだけでなく、計測と改善を見越したキュー設計が望ましい。

02

ノーコード・ローコードにおけるFeature Toggle

Act

# ノーコード・ローコードにおけるFeature Toggle

相対的に低くなっているが、ノーコード・ローコードでも開発にはハードルがある



## 道具がある

ワークフローデザイナーで  
ノーコード/ローコード  
フロー設計ができる



## 描ける

実際にフローを変更して、  
改善をすることができる。



# ノーコード・ローコードにおけるFeature Toggle

そのハードルを乗り越えるための支援を用意することでActが迅速・正確化



## 道具がある

ワークフローデザイナーで  
ノーコード/ローコード  
フロー設計ができる



## 枠組みがある

フローを変更するに  
あたっての枠組みがあり、  
完全自由変更ではない



## 描ける

実際にフローを変更して、  
改善をすることができる。



# ノーコード・ローコードにおけるFeature Toggle

技術的側面での対応方法の1つはFeature Toggle

ひとつの枠組み（考え方）

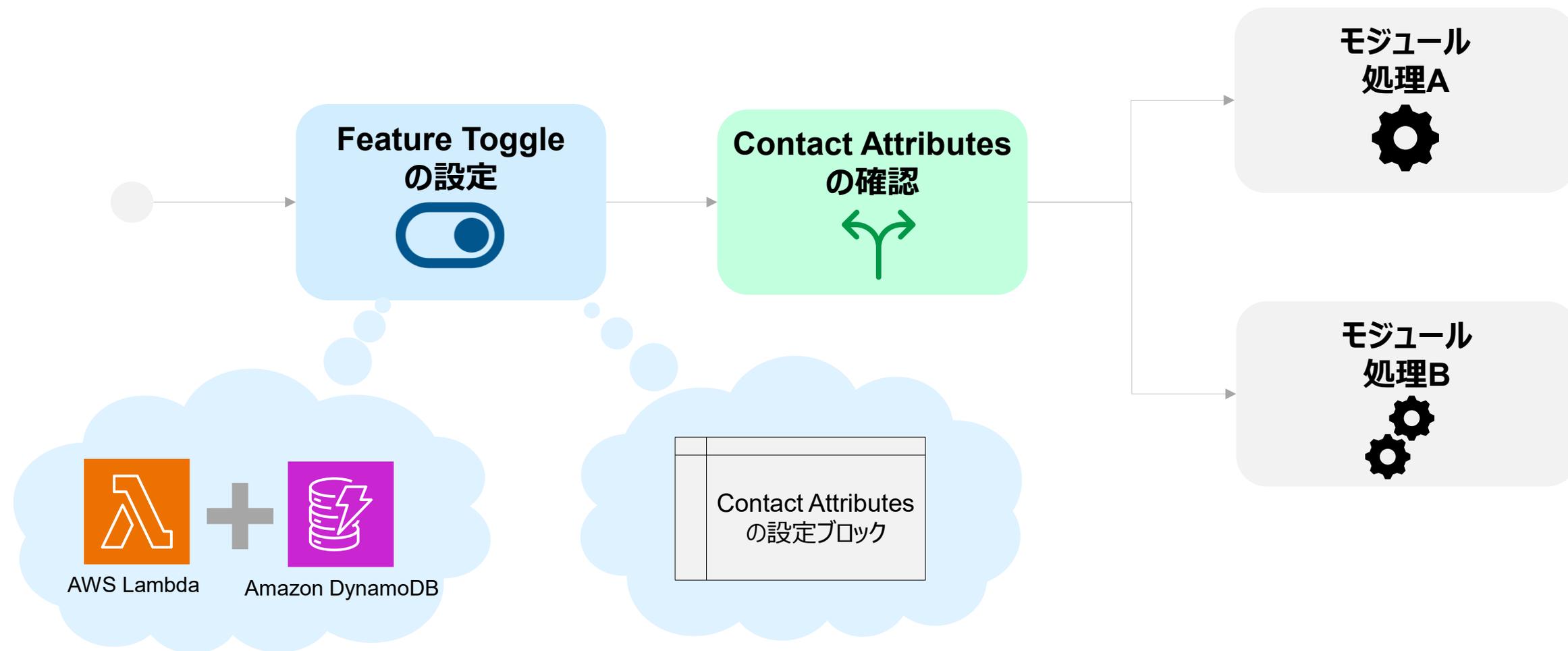


## Feature Toggle

コードの変更なしに、機能のON/OFFなどを切り替えるための変数

# Amazon ConnectにおけるFeature Toggle

DynamoDBや、Contact Attributesを利用することでToggleを作成・制御



# Amazon ConnectにおけるFeature Toggle

例えば以下のようなトグルによって、より機動的にコンタクトフローの修正が可能



## DRトグル

災害時などに主要なコンタクトパス以外を臨時に閉鎖するためのトグル



## ルーティング条件設定トグル

特定の問い合わせに要求されるスキルレベルの最小値などを変更するトグル



## 流量調整トグル

キューごとの流量割合を変更するためのトグル

## まとめ

01

メトリクス活用を見越したキューの設計

Observe

ルーティングだけでなく、計測と改善を見越したキュー設計が望ましい。

02

ノーコード・ローコードにおけるFeature Toggle

Act

計測結果をスピーディに反映することは重要。1つの方針はFeature Toggle

A low-angle photograph of a modern city skyline with several tall skyscrapers. The sky is a clear, deep blue. In the foreground, there are some trees and a street with a few vehicles. The text 'NTT Data' is overlaid in the center in a white, bold, sans-serif font.

**NTT Data**