SECURITY FOR APP BUILDERS #2

AI エージェントのアイデンティティ制御

AI エージェントを安全に提供するために

Ryuhei Shibata

Amazon Web Services Japan G.K.

Sr. Solution Architect

自己紹介

柴田　龍平

アマゾンウェブサービスジャパン
シニアソリューションアーキテクト

SaaS / ソフトウェアベンダーのお客様や
セキュリティに課題をお持ちのお客様を中心に
技術的なご支援しています。

アジェンダ

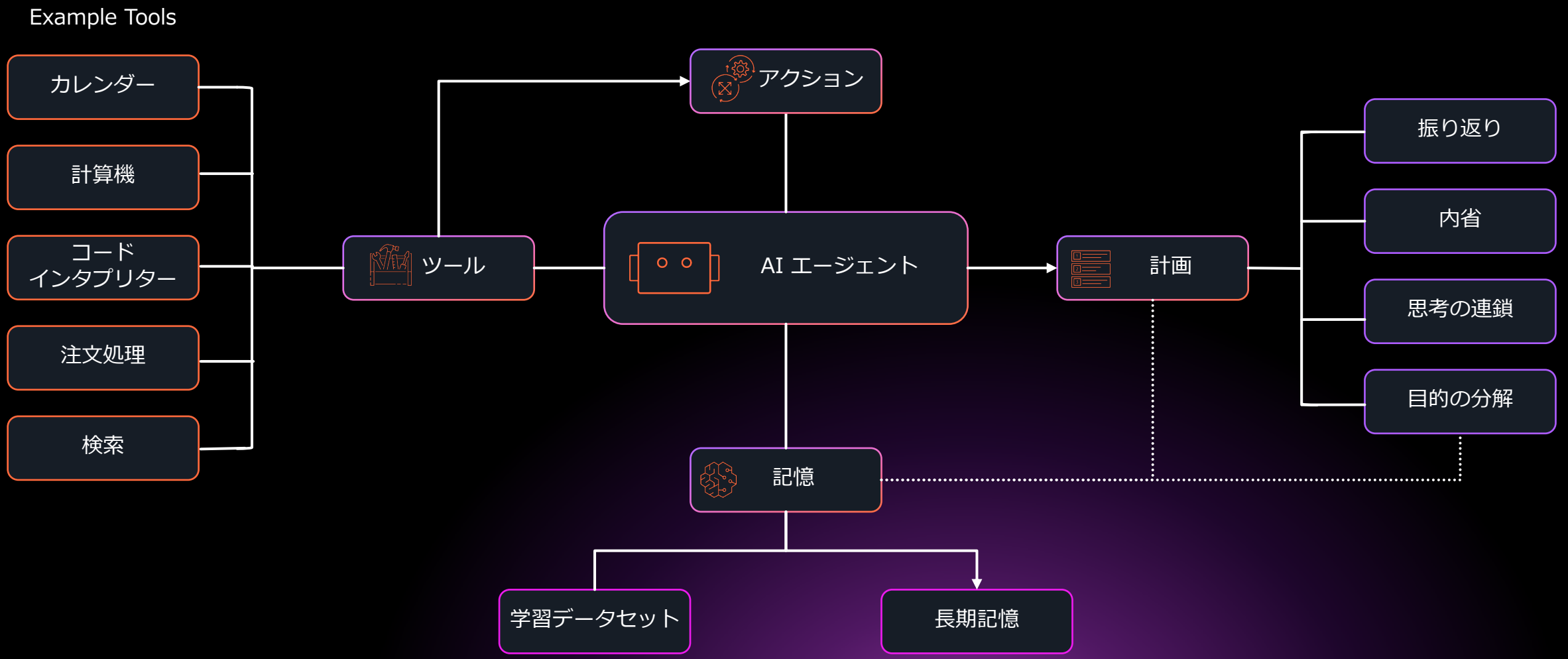
エージェント型 AI のセキュリティの基本

エージェントに求められる認証・認可・監査可能性

エージェントの認証・認可・監査可能性を支える AWS サービス

まとめ

エージェント型 AI とは？



エンタープライズの AI エージェントへの投資が拡大

33%

2028 年までにエンタープライズアプリケーションの 33 % が Agentic AI を組み込む (2024 年は 1% 未満)

Gartner, “Top strategic Technology Trends for 2025,” October 2024

15%

2028 年までに日々の業務における 15% の決断が Agentic AI により自動的に行われる

Gartner, “Top Strategic Technology Trends: Agentic AI—the Evolution of Experience” February 2025

エージェント型 AI の特性

エージェント型 AI は従来のソフトウェアとも生成 AI の特徴を併せ持つ

	従来のソフトウェア	生成 AI（チャット型）	Agentic AI
動作	決められた処理を実行	テキストを生成し人間に返す	**自律的にアクションを実行**
実行速度	高速 （自動的に実行）	低速 （人間のレビューで律速）	**中～高速 （人間の介在なく連続実行）**
外部影響	あり （ロジックに基づき直接作用）	限定的 （出力を人間が判断して実行）	**あり （予測不能な判断で直接作用）**
予測可能性	高 （決定論的動作）	中 （出力は確率的だが、人間が最終的な判断を実施）	**低 （非決定論的な動作かつ副作用あり）**

エージェント型 AI の特性

エージェント型 AI は従来のソフトウェアとも生成 AI の特徴を併せ持つ

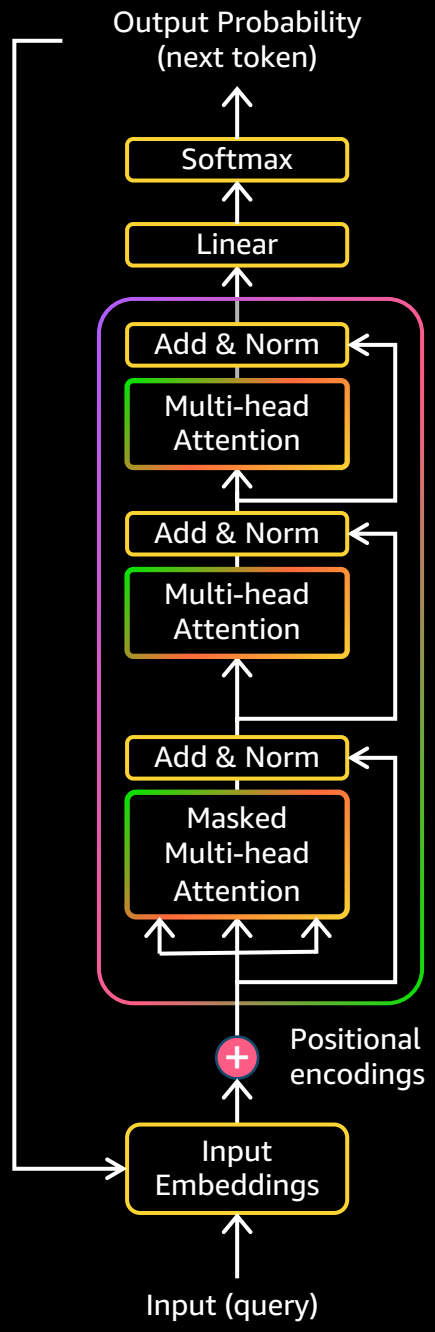
	従来のソフトウェア	生成 AI（チャット型）	Agentic AI
動作	決められた処理を実行		**自律的にアクションを実行**
実行速度	高速 （自動的に実行）	（人間のレビューで律速）	**中～高速** **（人間の介在なく連続実行）**
外部影響	あり （ロジックに基づき直		**あり** **（予測不能な判断で直接作用）**
予測可能性	高 （決定論的動作）	（出力は確率的だが、人間が最終的な判断を実施）	**低** **（非決定論的な動作かつ副作用あり）**

人間であれば止まるところを
何十倍、何百倍の速度で実行

与えるツールの権限が過剰だと
意図しない操作が発生しうる

LLM による推論

Transformer は入力（プロンプト）と追加コンテキストを、大量の学習データから獲得したパターンを用いて Attention や Feed Forward Network などからなる複数のニューラルネットワーク層で処理し、出力トークンを生成する。



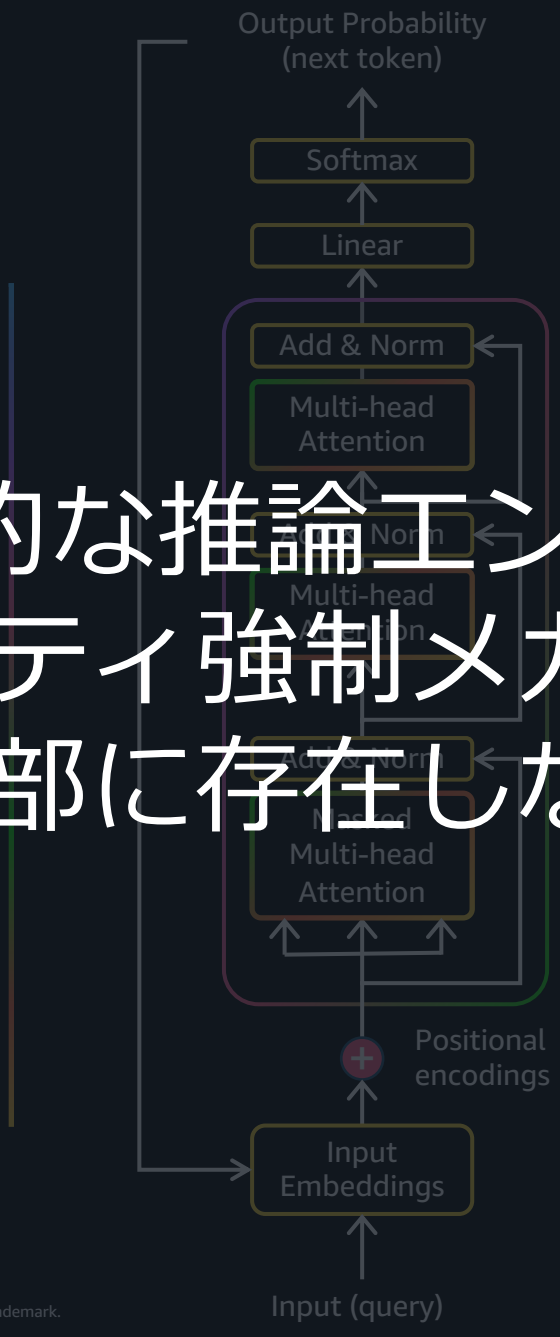
すべてのアテンションヘッドを通過すると、LLM は次のトークンの予測候補リストを出力する

LLM はクエリ・コンテキスト・学習データ（>数十億パラメータ）に基づき多様な関係性とパターンを捉える

クエリが エンベディングと位置エンベディングに変換される

LLM による推論

Transformer は入力（プロンプト）と追加コンテキストを、大量の学習データから獲得したパターンを用いて複数のニューラルネットワーク層で処理し、出力トークンを生成する。



すべてのアテンションヘッドを通過すると、LLM は次のトークンの予測候補リストを出力する

LLM はクエリ・コンテキスト・学習データ（>数十億パラメータ）に基づき多様な関係性とパターンを捉える

クエリが エンベディングと位置エンベディングに変換される

LLM は確率的な推論エンジンであり、セキュリティ強制メカニズムは内部に存在しない

Four Security Principles for agentic AI

https://aws.amazon.com/jp/blogs/security/four-security-principles-for-agentic-ai-systems/

Principle 1. セキュアな開発ライフサイクルのプラクティスはシステムの全構成要素に適用される

Principle 2. 従来のセキュリティコントロールは引き続き完全に適用される

Principle 3. 決定論的な外部コントロールがエージェントセキュリティの出発点である

Principle 4. より大きな自律性は継続的な評価を通じて獲得されるべきである

Defense in Depth Security (多層防御)

脅威検知とインシデント対応

データ保護

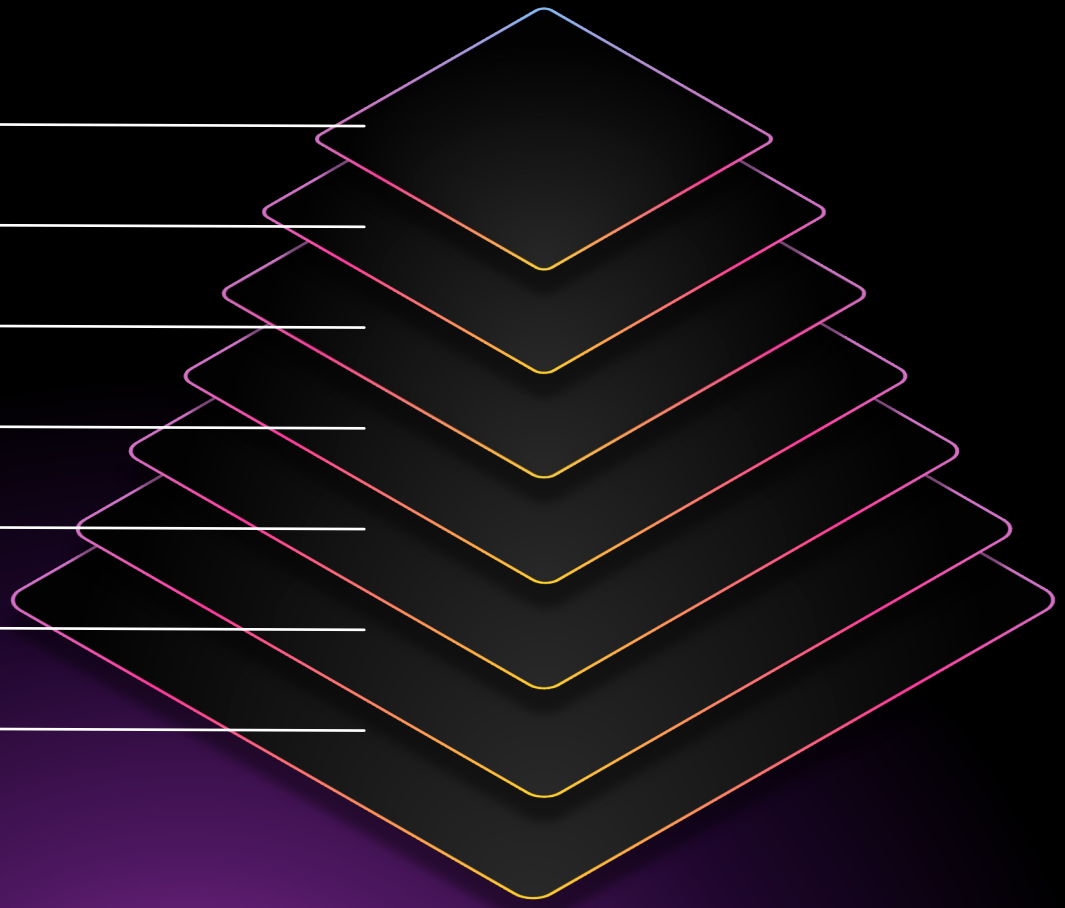
アプリケーション保護

ネットワーク＆エッジ保護

インフラストラクチャ保護

アイデンティティとアクセス管理

セキュリティに関する規定・手順・教育啓発



エージェントに決定論的制御を与えるアイデンティティ

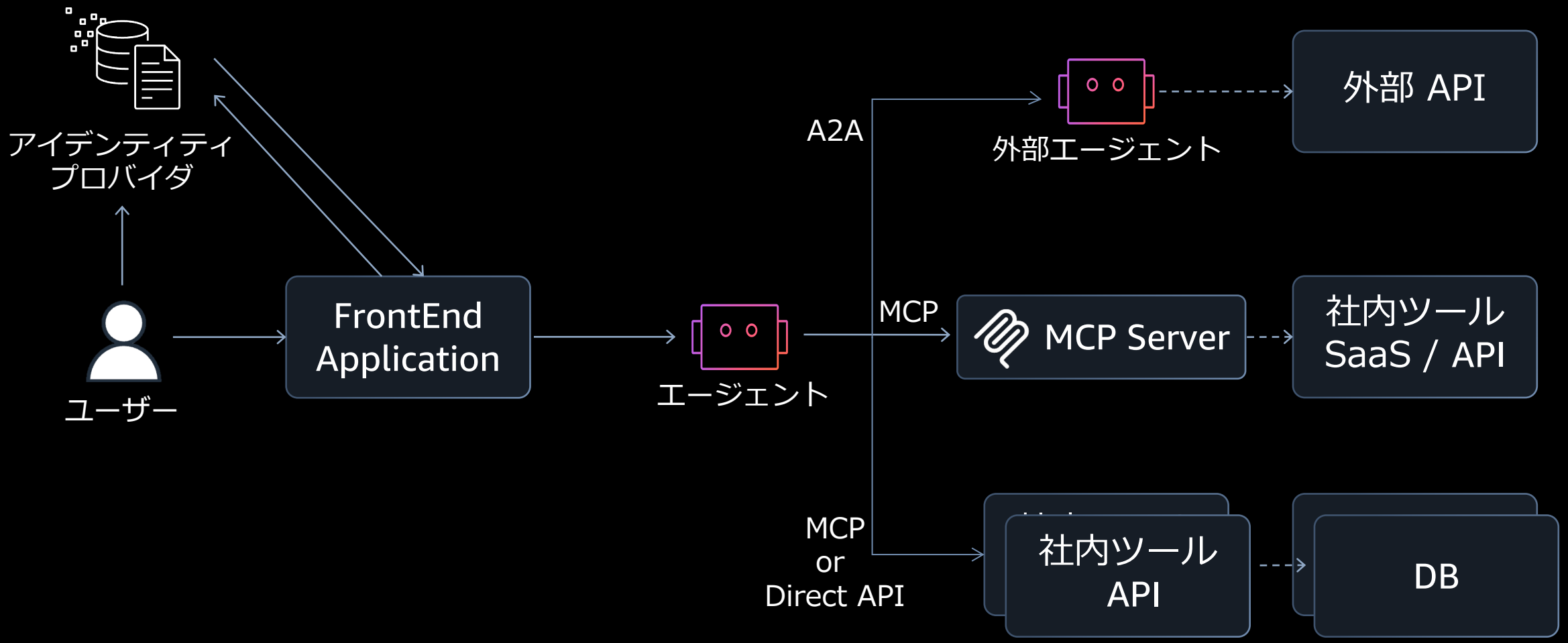
1. 認証 (Authentication)
エージェントを実行しているのは誰か
ユーザーやエージェントの属性は何か

2. 認可 (Authorization)
最小権限の実現：scope の制限、
コンテキストによる認可の実現

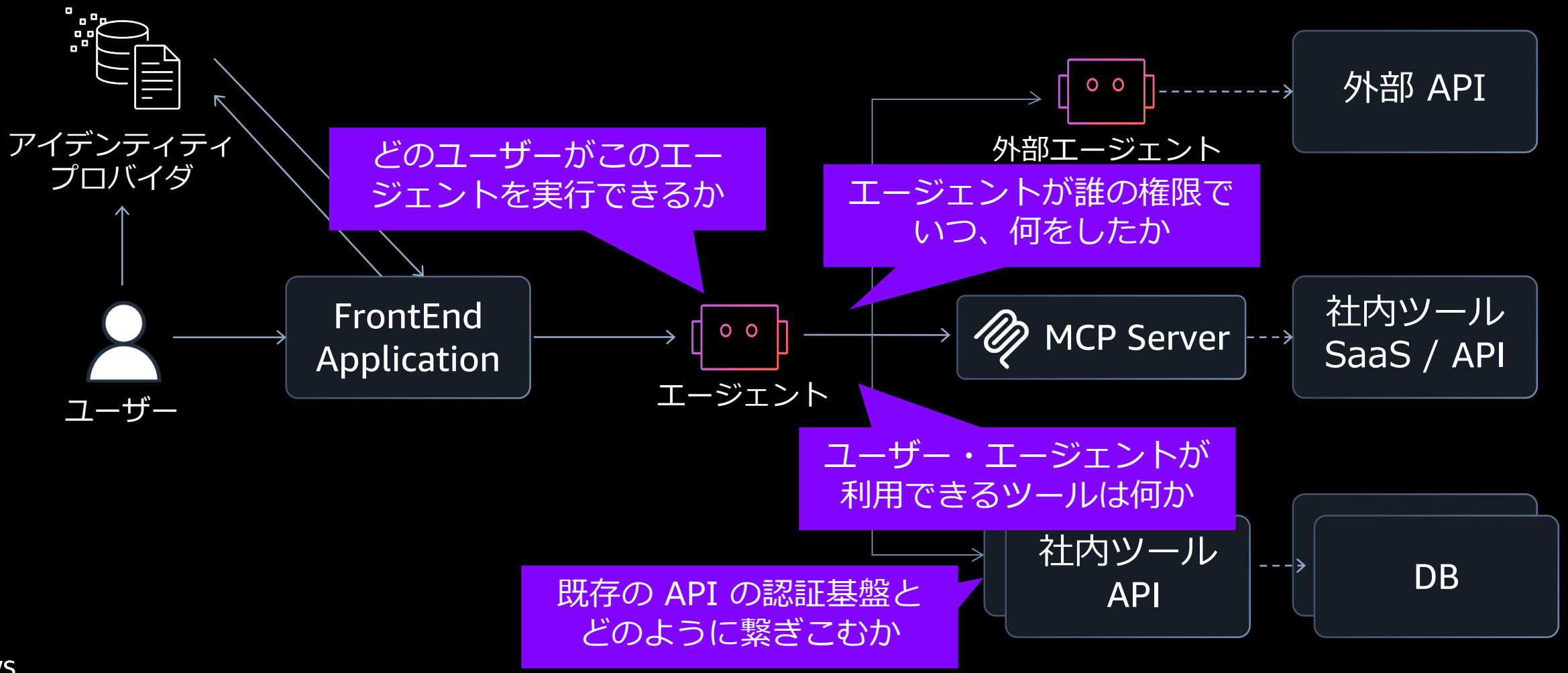
3. 監査可能性 (Auditability)
行動のログ保全：ユーザーの属性や
判断のためのコンテキストは何か

エージェントに求められる認証・認可・監査可能性

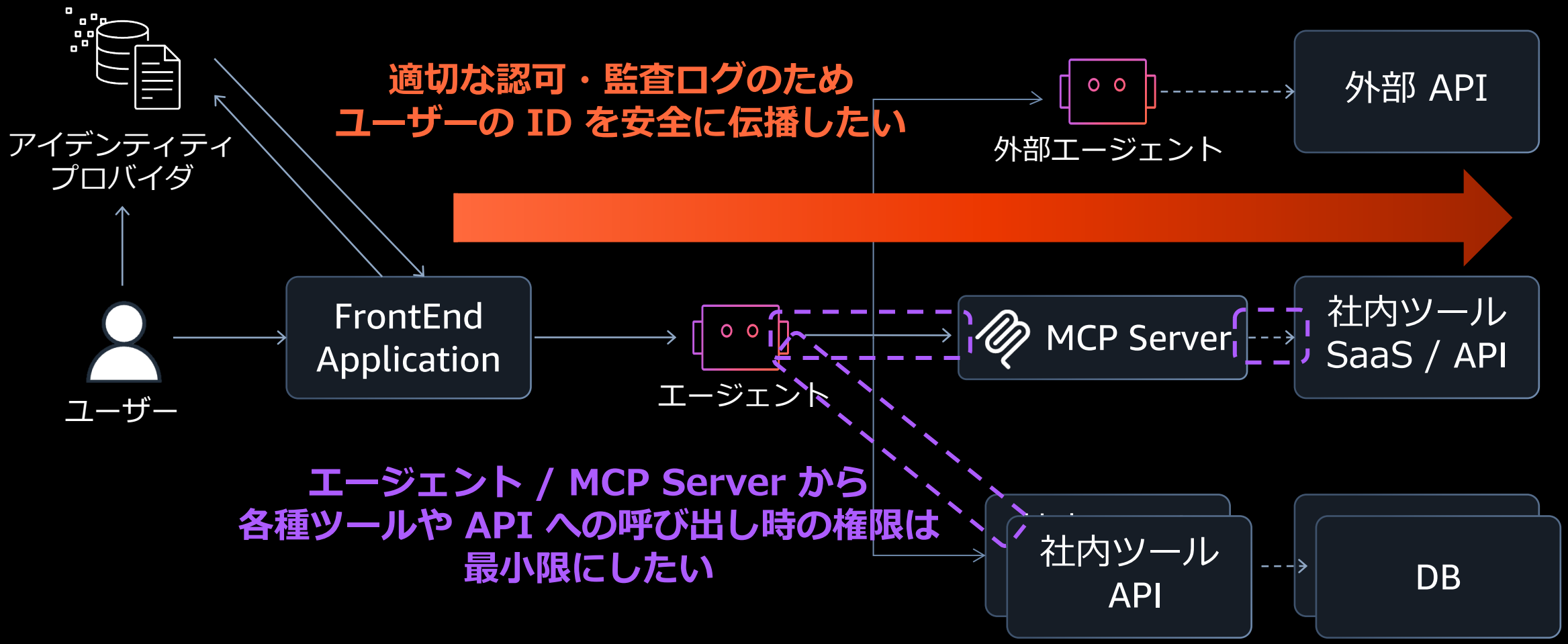
AI エージェントからの操作の流れ



エージェントや MCP サーバーでの認証認可の課題



ユーザーアイデンティティ伝播と最小権限原則の両立

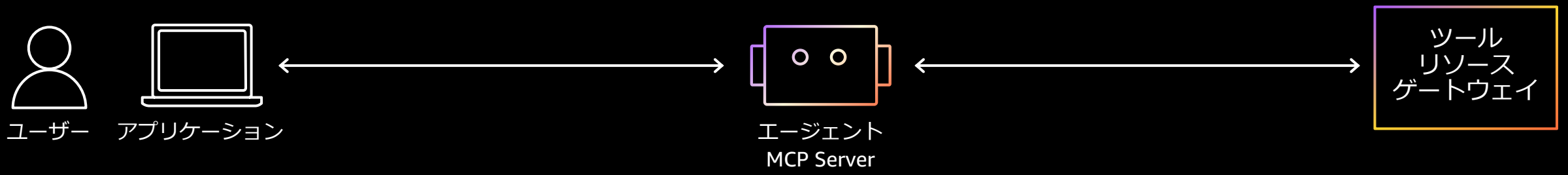


エージェント型 AI に必要な認証認可の整理

AI エージェントを構築するにあたって必要なのは…

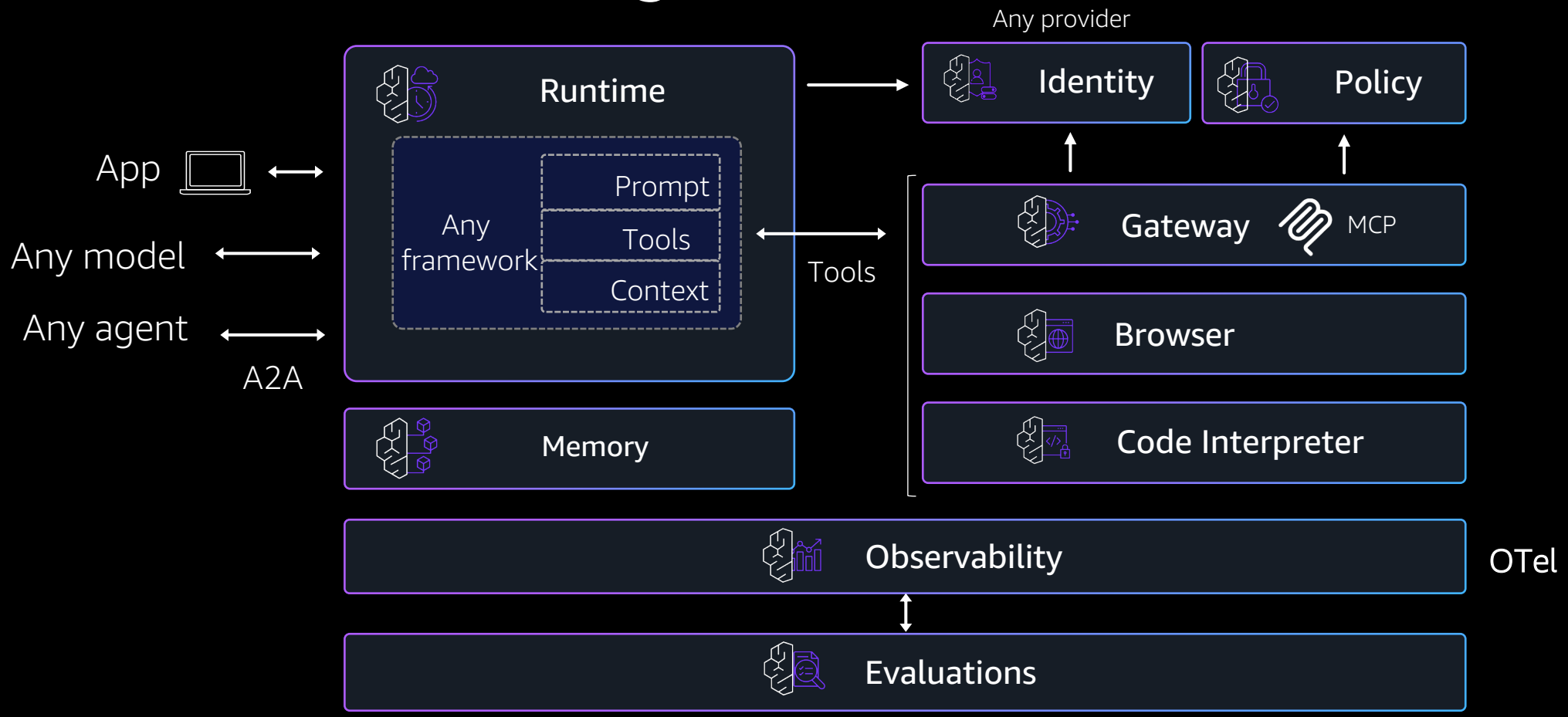
許可されたユーザーが
アプリケーションを通じて
エージェントにアクセス

呼び出し元の Identity (ユーザーまたはエージェント自身) の権限で
リソースに安全にアクセス



エージェントの認証・認可・監査可能性を支える AWS サービス

Amazon Bedrock AgentCore



AgentCore Identity で実現できる認証認可

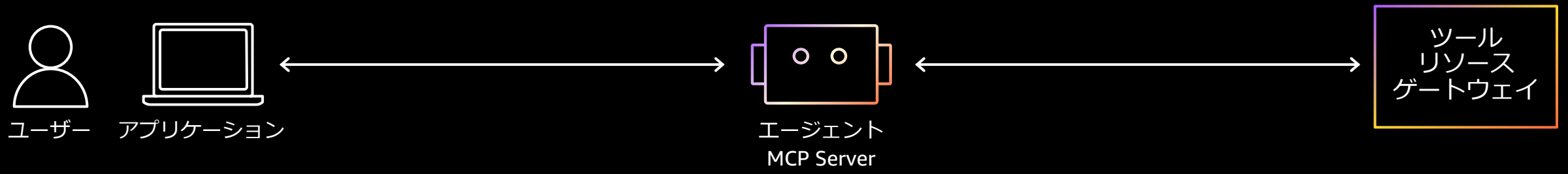
AI エージェントを構築するにあたって必要なのは…

許可されたユーザーが
アプリケーションを通じて
エージェントにアクセス

呼び出し元の Identity (ユーザー
またはエージェント自身) の権限で
リソースに安全にアクセス

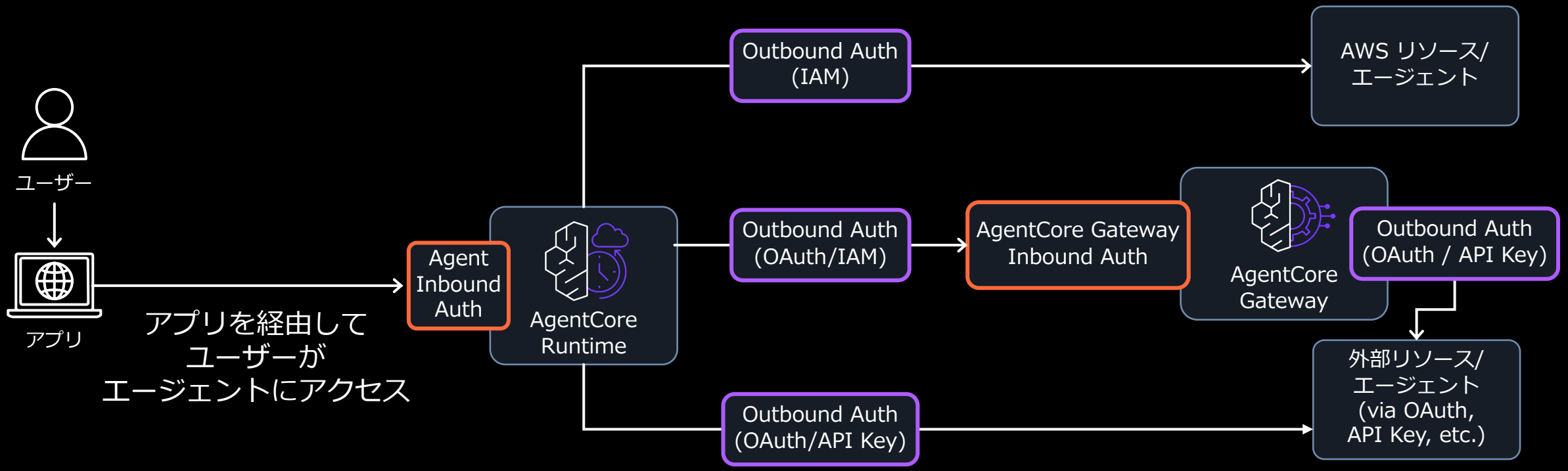
Inbound Auth

Outbound Auth



AgentCore Identity を使った認証・認可の流れ

エージェント / MCP Server に対するアクセスを制御 (**Inbound Auth**) して、
エージェントから外部のリソースへのアクセスのための認証情報を安全に交換 (**Outbound Auth**)



Inbound Auth によるエージェントへのアクセス制御

IAM SigV4 による制御

デフォルトの認証方式

呼び出し元アプリが EC2 / ECS / EKS 等で
稼働している場合 IAM でアクセス制御が完結
IAM Role を使えば認証情報ローテーション不要

ユーザー情報を Outbound Auth に伝播させる場合
X-Amzn-Bedrock-AgentCore-Runtime-User-Id ヘッダーを付与 (*)

JWT による制御

アプリケーションの認証情報を利用してアクセス

aud, client_id, scope や任意のクレームを
利用したアクセス制御が可能

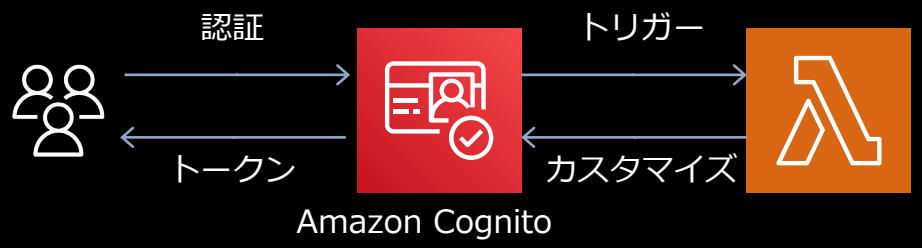
JWT に含まれるユーザーの ID は
自動的に OutboundAuth に伝播 (*)

(*) Outbound Auth ではなく、エージェントそのものにユーザー情報を渡す場合は条件を満たす任意のヘッダーが利用可能
https://docs.aws.amazon.com/bedrock-agentcore/latest/devguide/runtime-header-allowlist.html

アクセストークンにユーザーの属性を付与する

認可サーバーとしてAmazon Cognito を利用している場合、**Lambda トリガー**により
アクセストークンにユーザーの**ロール、部署、グループ、テナント**などの属性を付与可能

※ アクセストークンのカスタマイズには Essentials Tier 以上が必要

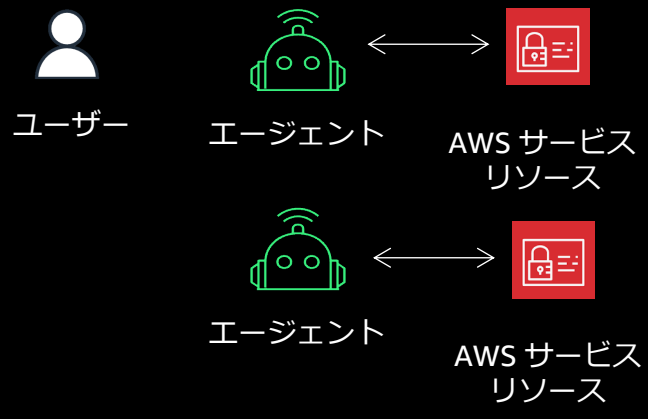



```
{
  "sub": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
  "iss": "https://cognito-idp.us-east-1.amazonaws.com/us-east-1_01EXAMPLE",
  "version": 2,
  "client_id": "1example23456789",
  "event_id": "01faa385-562d-4730-8c3b-458e5c8f537b",
  "token_use": "access",
  ”custom:division": ”sales",
  "scope": "openid profile email my-crm/read",
  "auth_time": 1702270800,
  "exp": 1702271100,
  "iat": 1702270800,
  "jti": "d903dcdf-8c73-45e3-bf44-51bf7c395e06",
  "username": ”alice"
}
```

AgentCore Identity がサポートする Outbound Auth

AWS リソースアクセス

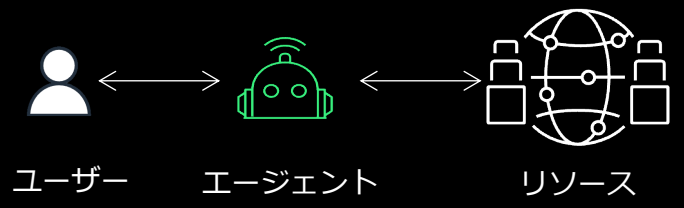
例: ユーザーがエージェントを呼び出して AWS サービスにアクセス



IAM

AWS リソース以外へのユーザー代理アクセス:

例: エージェントがユーザーに代わり SaaS や社内アプリにアクセス

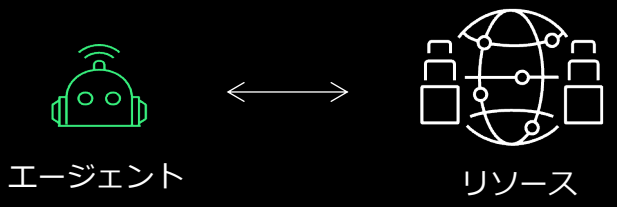


3LO Access
(Authorization Code Grant)
※初回のみユーザー同意が必要

On-Behalf-Of Access
(Token Exchange / JWT Bearer Grant)
※認可サーバーが対応している必要がある

AWS リソース以外へのマシンアクセス:

例: エージェントが外部 DB に定期クエリジョブを実行



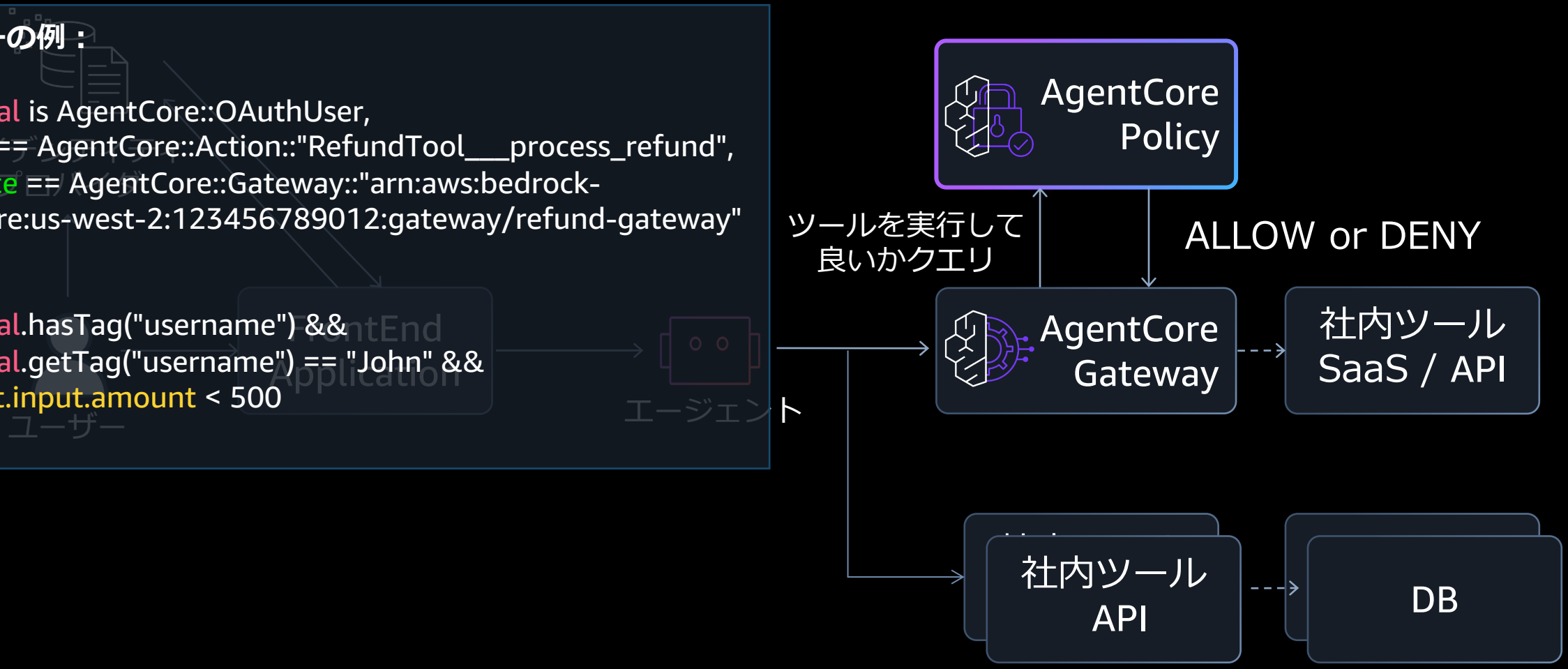
2LO Access
(Client Credentials Grant)

API Key

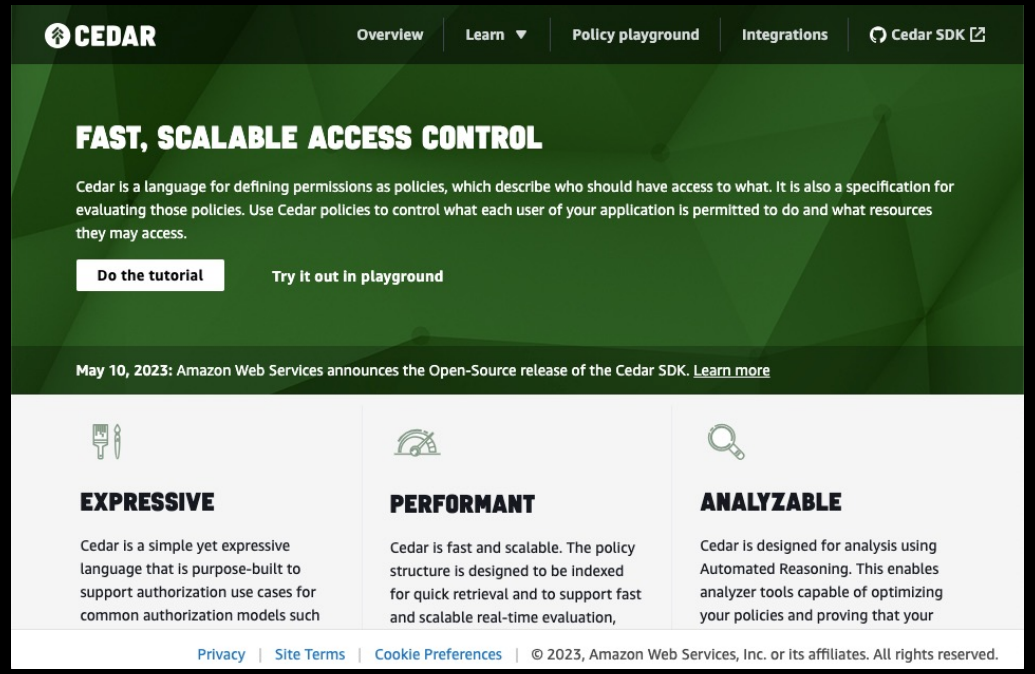
AgentCore Policy によるきめ細やかなツールの認可

ポリシーの例：

```
permit(
  principal is AgentCore::OAuthUser,
  action == AgentCore::Action::"RefundTool___process_refund",
  resource == AgentCore::Gateway::"arn:aws:bedrock-
agentcore:us-west-2:123456789012:gateway/refund-gateway"
)
when {
  principal.hasTag("username") &&
  principal.getTag("username") == "John" &&
  context.input.amount < 500
};
```



Cedar とは



https://www.cedarpolicy.com/en

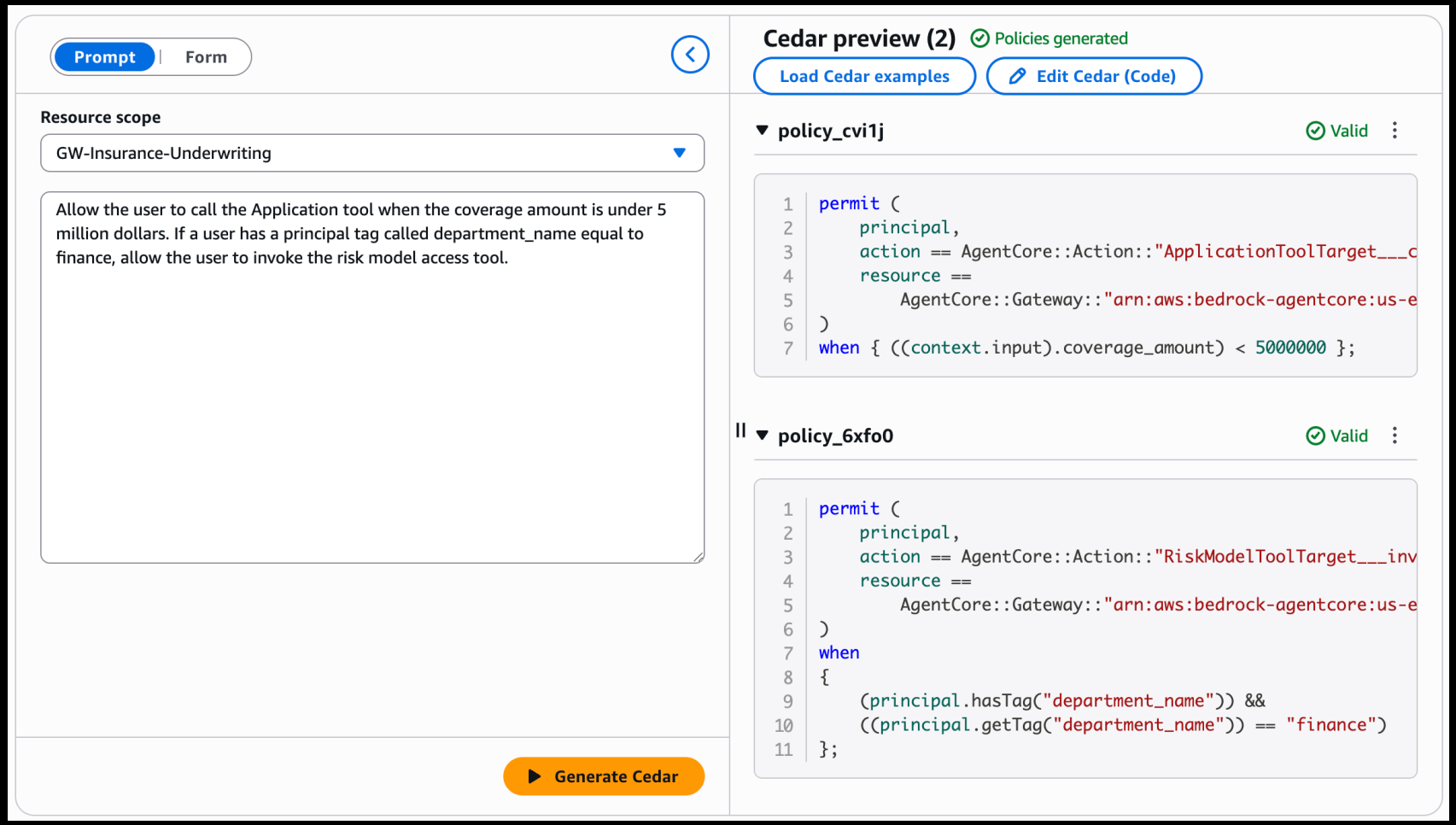
アプリケーションでの独自の認可を実現するための
AWS 製のオープンソースポリシー言語と認可エンジン

シンプルでありながら、ロールベースアクセス制御 (RBAC) や属性ベースのアクセス制御 (ABAC) もサポート

AWS Identity and Access Management (IAM) ポリシーとの類似性

- PARC (Principal, Action, Resource, Condition) を用いたアクセス許可
- デフォルトで拒否
- 明示的な拒否は許可よりも優先される
- ポリシー間に優先順位は存在しない

AgentCore Policy：自然言語による Policy 作成の支援



Principal:
AgentCore::OAuthUser::"<sub>"

Resource:
AgentCore::Gateway::"<ARN>"

Action:
AgentCore::Action::"<ToolName>"

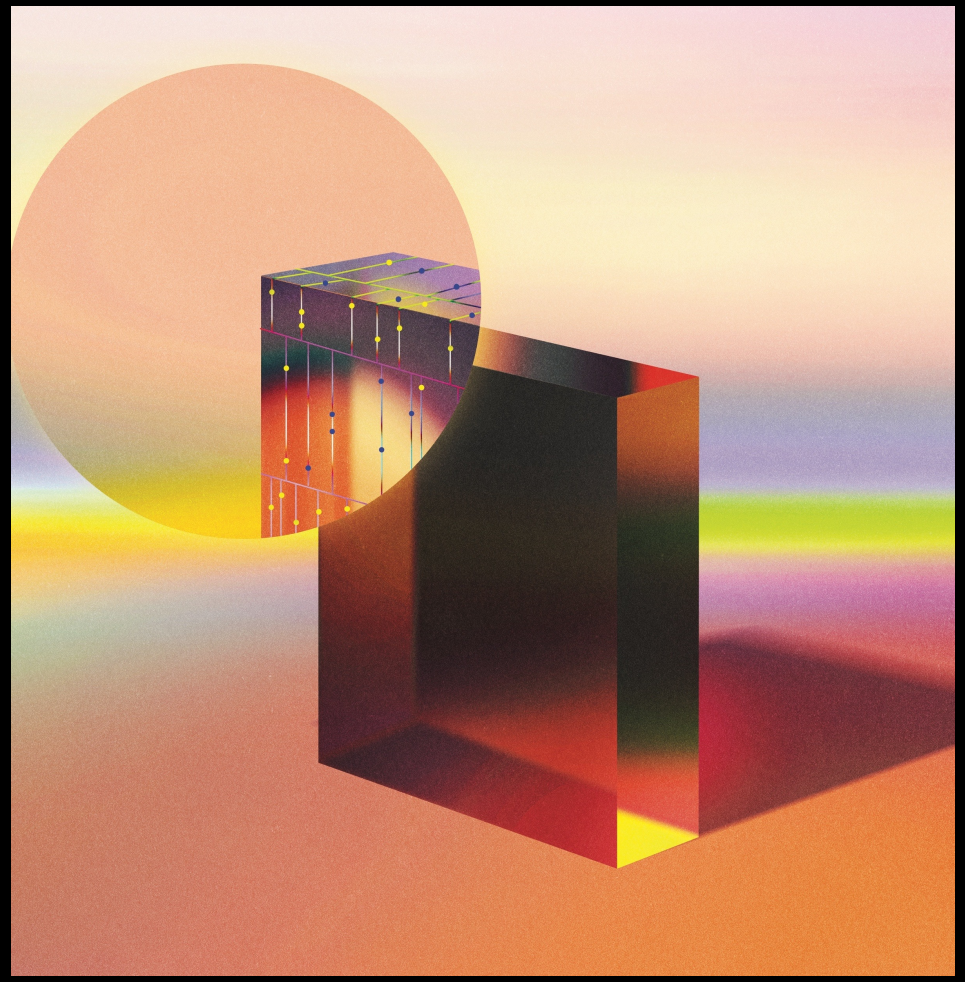
AgentCore Identity / Policy の監査可能性と可観測性

AWS CloudTrail 統合

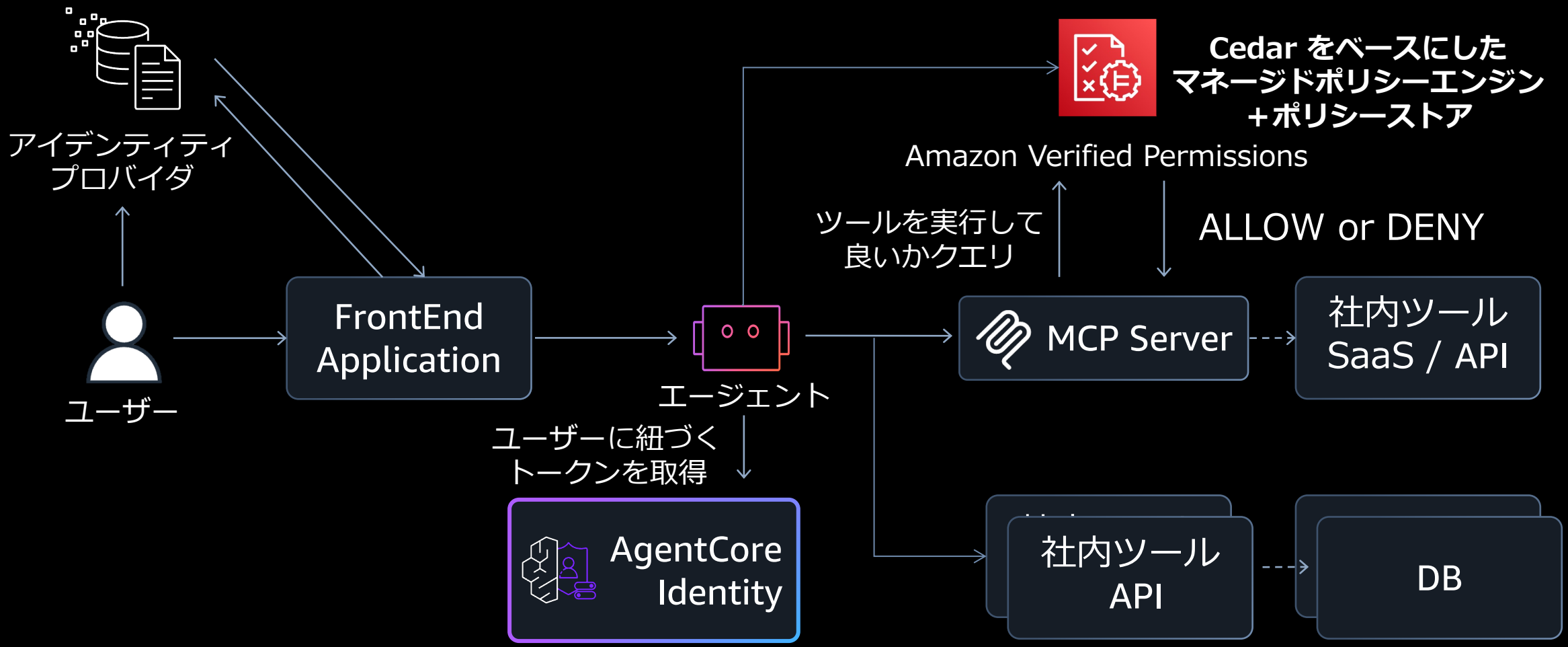
AgentCore Identity API の全 API コールと
AWS リソースへのエージェントアクションを
CloudTrail イベントとして記録可能

AgentCore Observability 統合

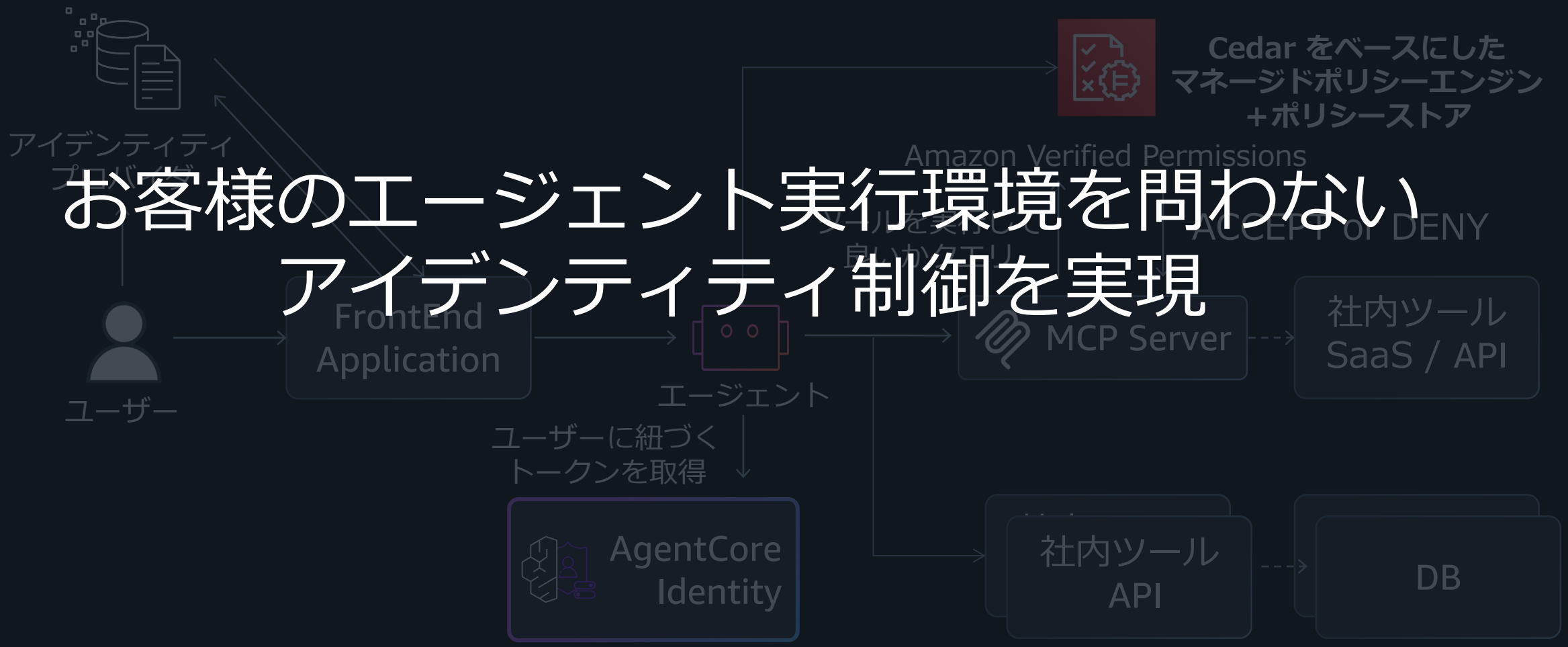
インバウンドの認証リクエスト、
アウトバウンドのトークン・API キー取得の
成功率、ポリシーの評価結果を
トレース・モニタリング



AgentCore Runtime / Gateway 以外の環境では・・・？



AgentCore Runtime / Gateway 以外の環境では・・・？



お客様のエージェント実行環境を問わない
アイデンティティ制御を実現

まとめ

アイデンティティ制御はエージェントセキュリティの基本

- 決定論的な外部コントロールがエージェントのセキュリティの出発点
- アイデンティティはエージェントに決定論的制御を追加する

AWS ではエージェントのアイデンティティ制御のための簡単な仕組みを提供

- AgentCore Identity でエージェントと MCP の認証認可をシンプルに管理
- AgentCore Policy や Amazon Verified Permissions によってツールやリクエストパラメーターレベルのきめ細やかな承認を実現

参考

AWS Security Reference Architecture – Generative AI agents

- https://docs.aws.amazon.com/ja_jp/prescriptive-guidance/latest/security-reference-architecture-generative-ai/gen-auto-agents.html

Four Security Principles for Agentic AI Systems

- https://aws.amazon.com/jp/blogs/security/four-security-principles-for-agentic-ai-systems/

Thank you!

Ryuhei Shibata
shibary@amazon.co.jp