# AI接続基盤準備：ChatGPTまたはClaude API統合構成設計書

## 1. 概要

本書は、ChatGPTまたはClaude APIを利用したAI接続基盤の構築設計書である。  
本構成では、APIを通じて自然言語処理モデルとの通信を実現し、Webサービスやバックエンド環境から安全かつスケーラブルにAI機能を利用可能にすることを目的とする。

## 2. システム構成概要

AI接続基盤は以下の3層構造で構成される：  
  
1. フロントエンド層：ユーザーインターフェース（React, Next.js など）  
2. バックエンド層：API Gateway および AI接続モジュール（Node.js, Python FastAPI など）  
3. 外部AI接続層：OpenAI ChatGPT API または Anthropic Claude API  
  
全通信はHTTPS経由で行われ、APIキーはAWS Systems Manager Parameter StoreまたはAWS Secrets Managerにて暗号化・管理される。

## 3. 技術要素

- API通信方式：RESTful / HTTPS (POST /v1/chat/completions)  
- 認証方式：Bearer Token方式  
- データ形式：JSON (UTF-8)  
- 通信暗号化：TLS 1.2以上  
- ログ監視：AWS CloudWatch / Datadog連携

## 4. 設計詳細

1. API Gateway構成  
 - AWS API Gatewayを利用し、バックエンド層への安全なエンドポイントを提供。  
 - Lambda関数またはECS Fargate上のアプリケーションへリクエストを転送。  
  
2. AI接続モジュール  
 - Python版: FastAPIベースのAIリクエストハンドラを実装。  
 - Node.js版: Express + Axiosを利用した非同期リクエスト制御。  
  
3. エラー制御  
 - HTTP 4xx/5xx ステータスに応じて再試行・リトライロジックを構築。  
 - レート制限対応（429 Too Many Requests時に指数バックオフを実装）。  
  
4. セキュリティ設計  
 - Secrets ManagerでAPIキー暗号化保存。  
 - IAMポリシーによりアクセスを限定（最小権限の原則）。  
 - CloudTrailでAPI呼び出し履歴を監査。

## 5. デプロイ・運用

- インフラ環境：AWS Lambda + API Gateway または ECS Fargate + ALB  
- CI/CD構成：GitHub Actions または AWS CodePipeline による自動デプロイ  
- 監視・アラート：CloudWatchメトリクス監視、Slack通知設定  
- ステージング環境：VercelまたはAWS Amplifyを利用したテスト環境を構築  
  
※セキュリティ確保のため、内部ネットワーク（VPC）内で通信経路を完結させる。