# デプロイ環境およびAI接続基盤運用指針

## 1. デプロイ環境設計（Vercelステージング環境）

本構成では、Vercelをステージング環境として採用し、安定したCI/CDパイプラインを構築する。開発から本番への移行時におけるデプロイの自動化・検証・エラーログ管理を包括的に管理する設計とする。

主な構成要素:

・Vercel Git Integration（GitHub連携による自動デプロイ）

・Preview DeploymentによるPR単位の環境生成

・環境変数（Environment Variables）の暗号化管理

・Custom Domain設定（例: staging.arwr.net）

### 1.1 環境変数設定例

Vercelプロジェクト設定画面にて、以下の環境変数を追加する：  
  
- NODE\_ENV=production  
- NEXT\_PUBLIC\_API\_URL=https://api.arwr.net  
- OPENAI\_API\_KEY（またはClaude\_API\_KEY）: セキュリティ保護済みキー

### 1.2 デプロイワークフロー

1. GitHubにてmainブランチへのpushをトリガーとしてVercelが自動ビルドを実行。  
2. Vercelビルドサーバー上で依存パッケージをインストール。  
3. Lintおよび型検証（TypeScript）実行後にアプリケーションをビルド。  
4. 成功時にPreview URLを自動発行。  
5. 承認後、Production Deploymentを手動または自動で実行。

### 1.3 セキュリティ対策

・Vercel環境変数をチーム単位で制限（READ権限）

・自動生成Preview環境への認証ゲートを設定（Basic AuthまたはToken）

・ドメインキャッシュのDNS TTL短縮による迅速な切替制御

## 2. AI接続基盤構成およびAPIキー管理方針

本システムでは、ChatGPT（OpenAI API）またはClaude（Anthropic API）を利用する構成を前提とする。  
クライアント側でAPIキーを発行し、キーごとの利用制限およびローテーション方針を明確化する。

### 2.1 APIキー発行および構成

クライアントが各AIサービスプロバイダ（例：OpenAI Dashboard, Anthropic Console）上でAPIキーを発行。  
発行されたキーは次の形式で環境変数としてVercelに登録する：  
  
- OPENAI\_API\_KEY=sk-xxxxxxxxxxxxxx  
- CLAUDE\_API\_KEY=ak-xxxxxxxxxxxxxx

### 2.2 キー管理ポリシー

・APIキーは開発者個人ではなく組織単位で発行・管理する。

・キーは定期的にローテーション（推奨：30〜60日周期）。

・権限スコープを最小限に設定（読み取り専用または単一エンドポイント限定）。

・キーの使用状況をダッシュボードで監視し、異常アクセスを自動検知。

### 2.3 API利用制限設計

APIキーごとの利用制限は以下の通り設定する：  
  
- レートリミット：60 requests/minute  
- 1ユーザーあたり最大月間トークン数：50,000 tokens  
- 不正利用時には自動でキーを無効化し、アラート通知を送信

### 2.4 エラーハンドリング方針

API障害・レート超過時に備え、フロントエンドおよびバックエンドに以下のリトライ戦略を実装する：  
  
- HTTP 429応答時：指数バックオフ方式（最大3回）  
- HTTP 500系応答時：ユーザー通知＋ロギング  
- APIトークン失効時：自動キー再取得または手動再設定フローへ誘導

## 3. 運用監視およびアクセスログ管理

VercelとAWS CloudWatchを併用し、アプリケーション全体の稼働監視を行う。  
AI APIへのアクセス履歴は専用ログサーバーに転送し、週次で分析レポートを生成する。

アクセスログには以下の情報を含む：

　- API呼び出しユーザーID

　- 呼び出しエンドポイントおよびレスポンスステータス

　- 処理時間（latency）および利用トークン数