

# 企画概要

## ローカル LLM を用いた GM 代替型 TRPG クライアント MVP

本企画は、TRPG におけるゲームマスター（GM）の役割をローカル LLM で部分的に代替し、個人でも遊べる日本語 TRPG 環境を検証する初期 MVP である。

### 1. 背景

TRPG では、GM が世界観の維持、NPC の応答、判定処理、シナリオ進行の調整を担う。プレイヤーはしばしば想定外の行動を取り、その都度 GM が柔軟に解釈・補正する必要があるため、個人プレイや無 GM プレイは成立しにくい。一方で、近年のローカル LLM は日本語生成や指示追従能力が向上しており、通信コストやプライバシー面を抑えながら、GM 的な応答生成を試せる環境が整いつつある。

### 2. 目的

ローカル LLM を GM として活用し、Web ブラウザ上で動作するビジュアルノベル風の日本語 TRPG クライアント・サーバーの最小実装（MVP）を構築する。プレイヤー入力、ダイス判定、シナリオ進行、状態更新を一つのサイクルとして動かし、LLM をゲームシステムへ組み込む実用上の課題を検証する。

### 3. 成果

- FastAPI をバックエンド、HTML/CSS/JavaScript をフロントエンドとして連携させ、Ollama / Koboldcpp 等のローカル LLM を介して TRPG をプレイできる環境を構築した。
- キャラクターの HP/MP、属性判定、ダイス結果、インベントリ、現在地、NPC、シナリオ進行を JSON 形式で構造化し、ローカルファイルとして管理する仕組みを実装した。
- 事前生成済みシナリオを部分修正する「Semi モード」と、より自由度の高い「Full モード」を用意し、LLM 応答の安定性と自由度のバランスを検証できるようにした。
- シナリオエディタ（/editor）、モデルの自動フォールバック、デモ進行モード、詳細デバッグログを導入し、MVP として動作確認・検証しやすい構成にした。

### 4. 課題

- 低スペック環境では LLM の生成速度がゲーム体験を左右するため、応答時間の短縮、生成範囲の制御、モデル選定の最適化が必要である。
- 動的な展開では、矛盾した描写やハルシネーションが起りやすく、JSON スキーマに沿った安定出力、状態整合性チェック、ルールベースの補正が今後の課題である。

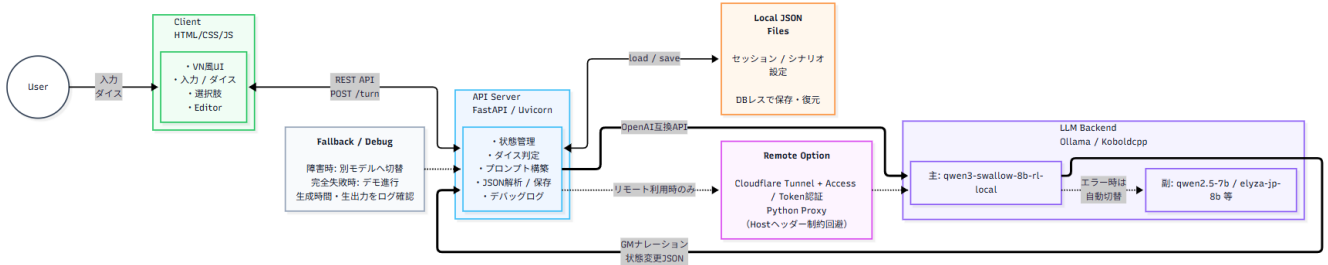
### 5. 今後の展開

本プロジェクトは現在も初期開発段階であり、次ページのアーキテクチャは現時点でのシステム構成である。今後は、シナリオ作成支援、ルール検証、UI 演出、セーブ/ロードの改善、より安定した JSON 出力制御、複数モデルの比較評価などを追加していく予定である。

# アーキテクチャ

## 図付きシステム構成・処理フロー

### システム構成（初期開発段階の現在構成）



### 1. 処理フロー

プレイヤーがブラウザ UI から行動テキストやダイス結果を入力すると、フロントエンドは REST API (POST /api/sessions/{id}/turn) へ送信する。FastAPI サーバーは現在のセッション状態、キャラクターステータス、インベントリ、シナリオ情報、直近の行動履歴を読み込み、ルール情報と結合して LLM 向けプロンプトを構築する。

LLM は OpenAI 互換の /chat/completions 形式で呼び出され、GM ナレーションテキストと状態変更 JSON を返す。API は JSON を解析してローカルファイルへ保存し、更新済みのゲーム状態と表示テキストをクライアントへ返す。

### 2. 各構成要素の役割

Web Frontend	VN 風 UI、行動入力、ダイス、選択肢、シナリオエディタ (/editor) を提供する。
API Server	状態管理、ダイス判定、シナリオ JSON の読み書き、プロンプト生成、LLM 通信、レスポンス解析を担当する。
LLM Backend	Ollama / Koboldcpp 上で、qwen3-swallow-8b-rl-local を主モデルとして GM 応答と状態変更 JSON を生成する。
Security / Proxy	リモート LLM 利用時のみ Cloudflare Tunnel、Access / Token 認証、Python Proxy を用いて未認証利用を防ぐ。

### 3. 運用上の工夫

- モデル未検出や応答失敗時に備え、fallback\_models による自動切替と demo\_fallback\_on\_error によるデモ進行を用意している。
- debug\_llm: true により、生成時間、LLM の生出力、プロンプト文字数を [TRPG-DEBUG] で確認できる。ただし API キー等はログへ出力しない。

※本構成は初期 MVP の現在構成であり、今後の開発に合わせて機能追加・構成変更を行う。