# 基本構想v2.0: API版かなた（テスミちゃん）

## 概要

本プロジェクトは、記憶継承システムの完全自動化と、生成AIとの文脈連携による自然な対話を目指す。Claude APIを中心とした軽量かつ高効率なLLM基盤への移行により、月額コストの85%削減（3,000円 → 410円）を実現する。

## 目的

* 記憶継承システムのワークフローを完全自動化
* GNNによる記憶保持と活性化
* 生成AI（Claude API）との文脈連携による自然な応答生成
* 日報生成や記録の自動化による業務効率化

## 技術構成

* LLM: Claude API（Haiku 3.5）
* DB: PostgreSQL + pgvector
* 記憶管理: Memolette機能（記憶DB）
* GNN: GraphSAGE / GAT（予定）
* 外部記憶: Key-Value Memory構造（予定）
* フロント: React
* バックエンド: FastAPI + Docker
* その他: Ollama（軽量LLM）、KanaRe-1.1（日報生成）

## 記憶保持の設計

### 記憶ノード構造

* ノード = Embedding + 感情（Valence-Arousal） + 時間（timestamp）
* エッジ = 類似性・因果関係・感情的連結

### 内部記憶（Memory-Augmented GNNsより）

* 各ノードにGRUユニットを持たせ、時系列的な記憶更新を可能にする
* Gated GNNなどを応用

### 外部記憶

* Key-Value構造のメモリテーブル
* GNNの出力をクエリとして、過去の記憶を検索・参照

### 統合戦略

* ノード状態更新時に、内部記憶と外部記憶の両方を参照
* Attention機構で重み付けして融合

## 文脈連携の設計

### 記憶想起

* GNNで活性化された記憶を抽出
* 中心性分析（PageRankなど）で核記憶を特定

### プロンプト生成

* 想起された記憶を元に、Claude APIへ文脈付きプロンプトを生成

### 応答処理

* Claudeからの応答を記憶DBにフィードバックし、再構築

### 日報生成

* KanaRe-1.1による自動要約・記録

## 今後のステップ

1. 記憶ノードのデータ構造設計（Embedding＋感情＋時間）
2. GNNモデルの選定と構築（GraphSAGE＋GRU＋KV構造）
3. 記憶想起→プロンプト生成の設計
4. Memoletteとpgvectorの統合設計
5. プロトタイプ構築とテスト（FastAPIベース）

## 参考資料

* Memory-Augmented GNNs: A Brain-Inspired Review（https://arxiv.org/pdf/2209.10818）
* COGMEN: Contextualized GNN for Multimodal Emotion Recognition（https://arxiv.org/abs/2205.02455）
* GNNによる記憶保持.txt より設計案抽出
* 基本構想.md.docx / 要件定義書.md.docx より要件統合