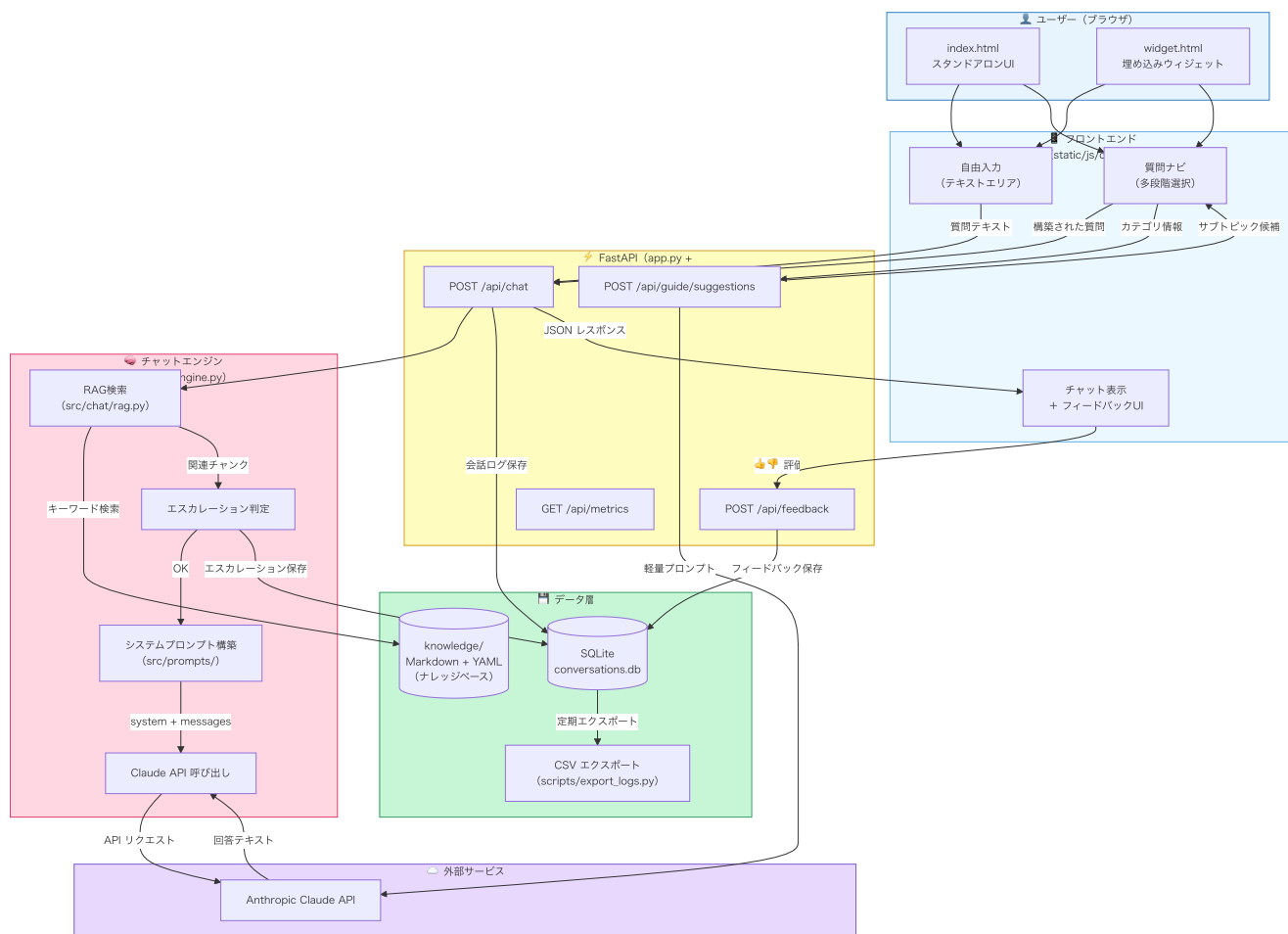


牧野生保塾 AI伴走システム - プロセスフロー図

入力 → 処理 → 出力 → データ格納の全体フロー

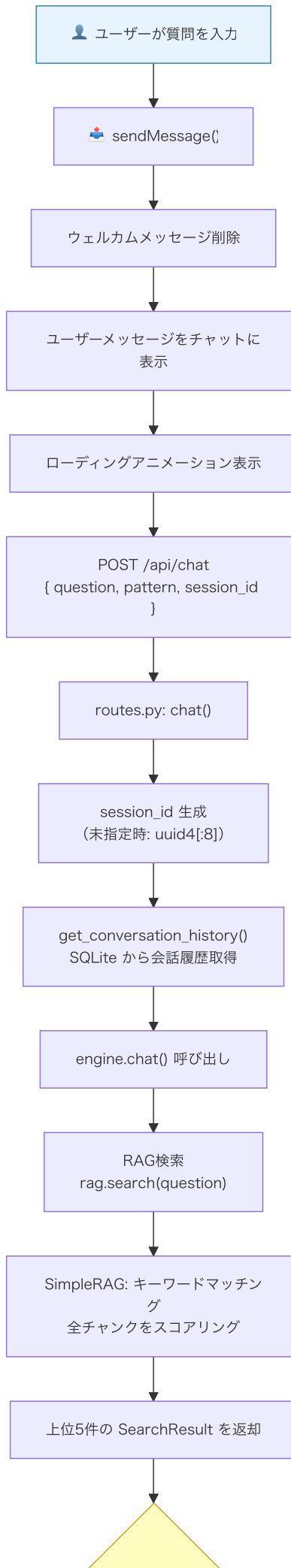
1 全体アーキテクチャ概観

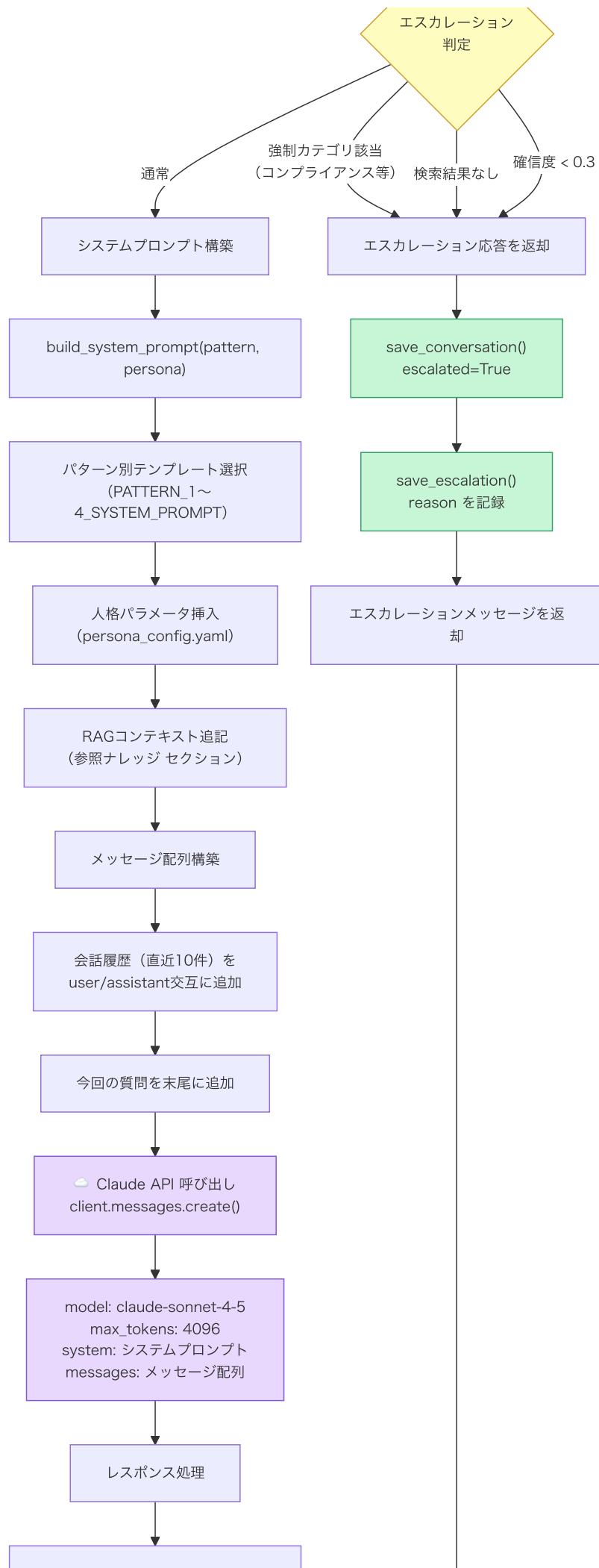
システムを構成する全レイヤーと、データの流れ方向を示す。



2 メインチャットフロー (自由入力 → 回答表示)

ユーザーが質問を入力してから回答が画面に表示されるまでの詳細な処理ステップ。





```
answer =  
response.content[0].text
```

関連ファイル: static/js/chat.js src/api/routes.py src/chat/engine.py src/chat/rag.py
src/prompts/system_prompts.py src/database/operations.py

```
tokens_used = input + output  
tokens
```

3 質問ナビフロー（ガイド付き入力 → 質問生成）

```
confidence = RAGスコア平均値
```

質問ナビの3ステップで質問が組み立てられ、最終的にメインチャットフローに合流する流れ。

```
category = キーワード簡易分類
```

```
save_conversation()  
SQLite conversations テーブル
```

```
ChatResponse を返却
```

```
JSON レスポンス → フロントエ  
ンド
```

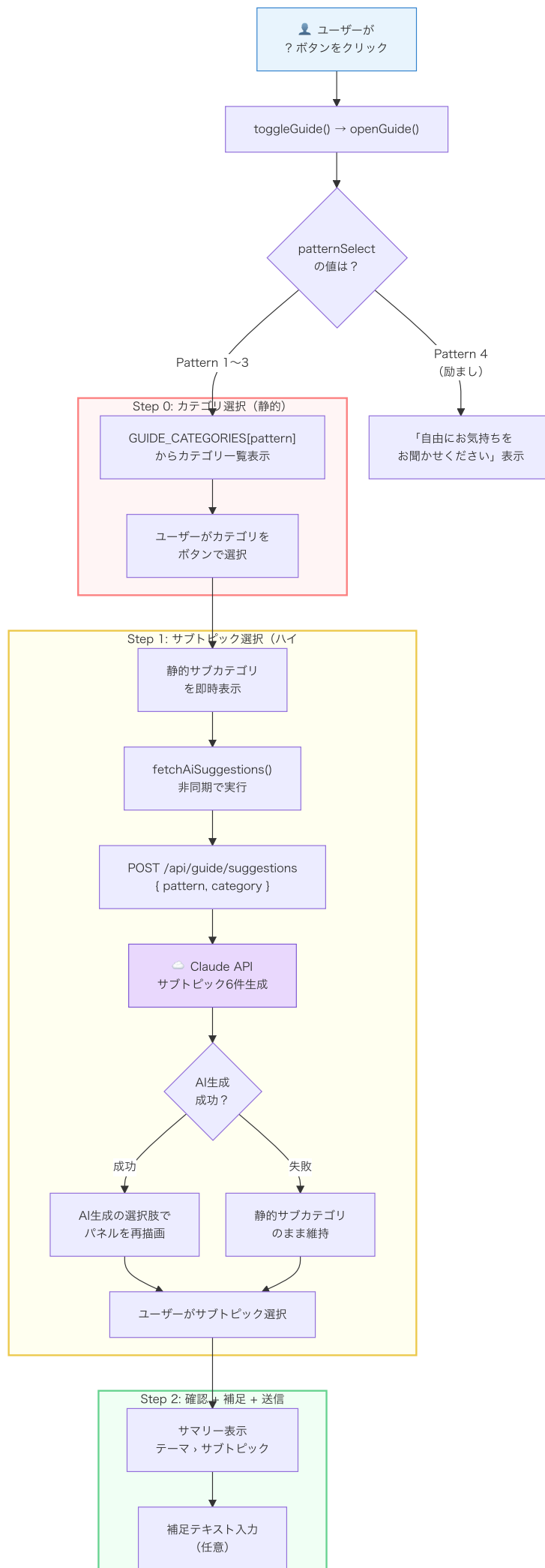
```
ローディング除去
```

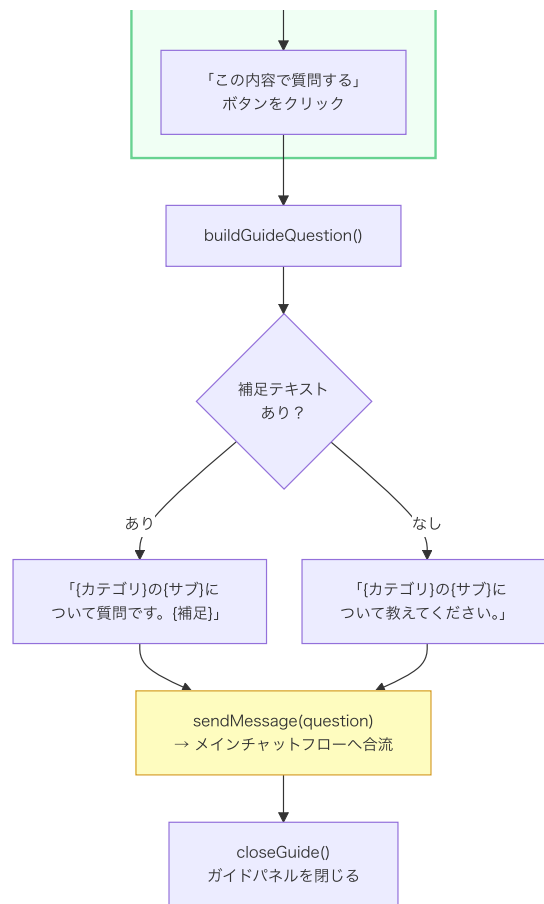
```
appendMessage('assistant',  
answer)
```

```
回答バブル表示
```

```
出典表示 (details/summary)
```

```
フィードバックボタン表示  
(👍👎)
```

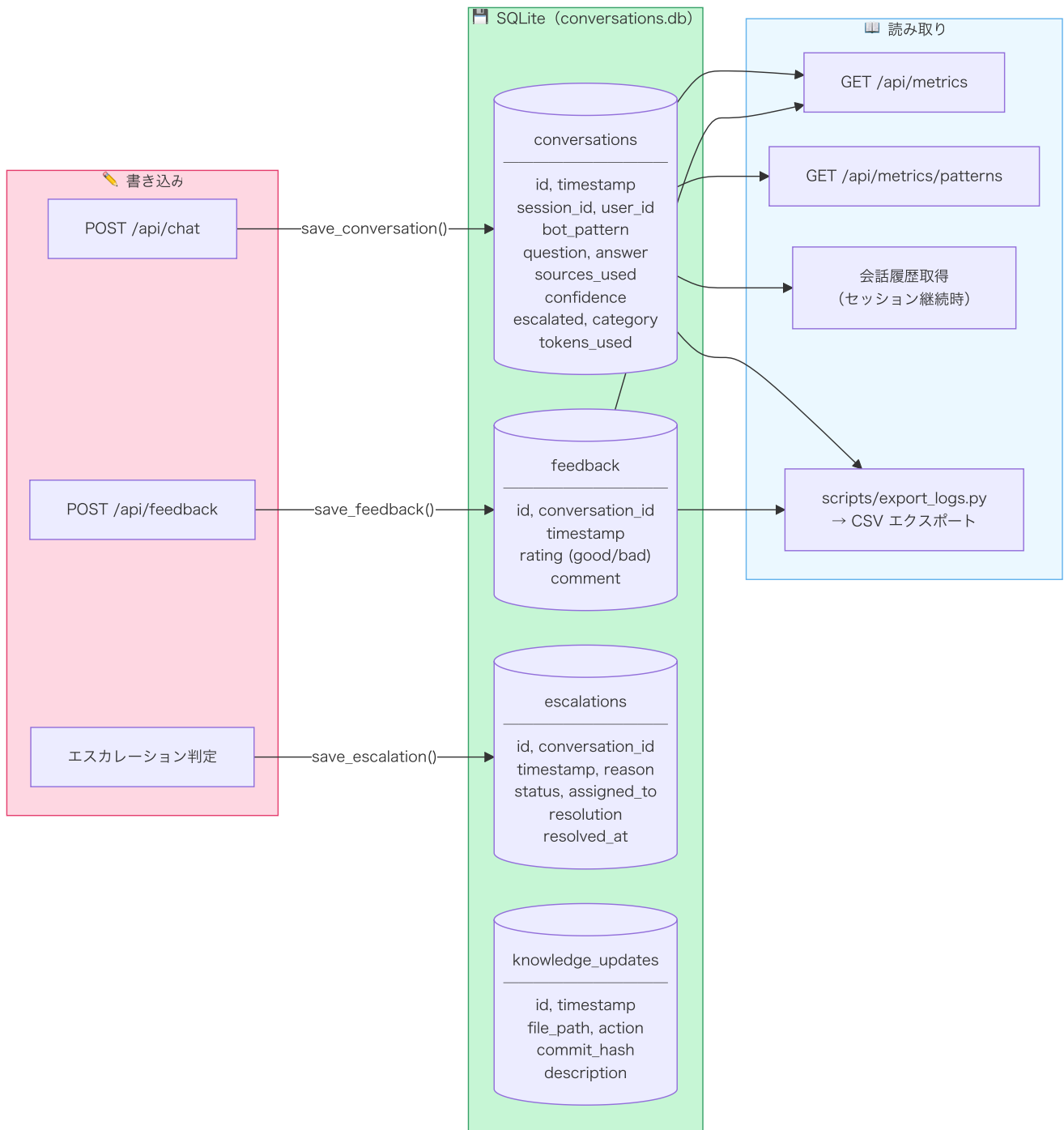





関連ファイル: `static/js/chat.js` (`GUIDE_CATEGORIES`, `renderGuideStep`, `fetchAiSuggestions`, `buildGuideQuestion`) `src/api/routes.py` (`guide_suggestions` エンドポイント)

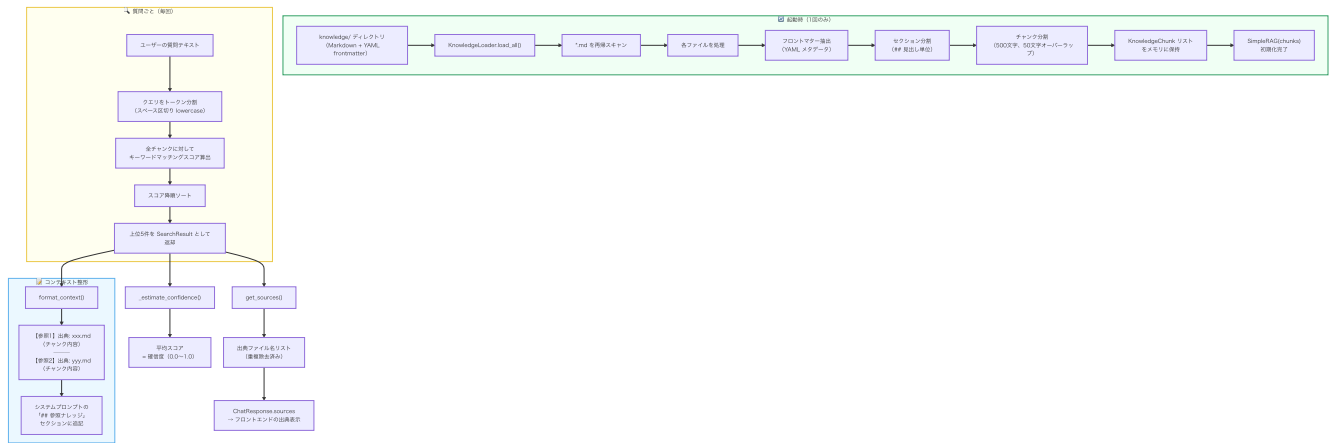
4 データフロー (SQLite 書き込み・読み取り)

どのアクションがどのテーブルにデータを書き込み、どこから読み取るかの全体像。



5 RAG処理 (ナレッジベース → コンテキスト生成)

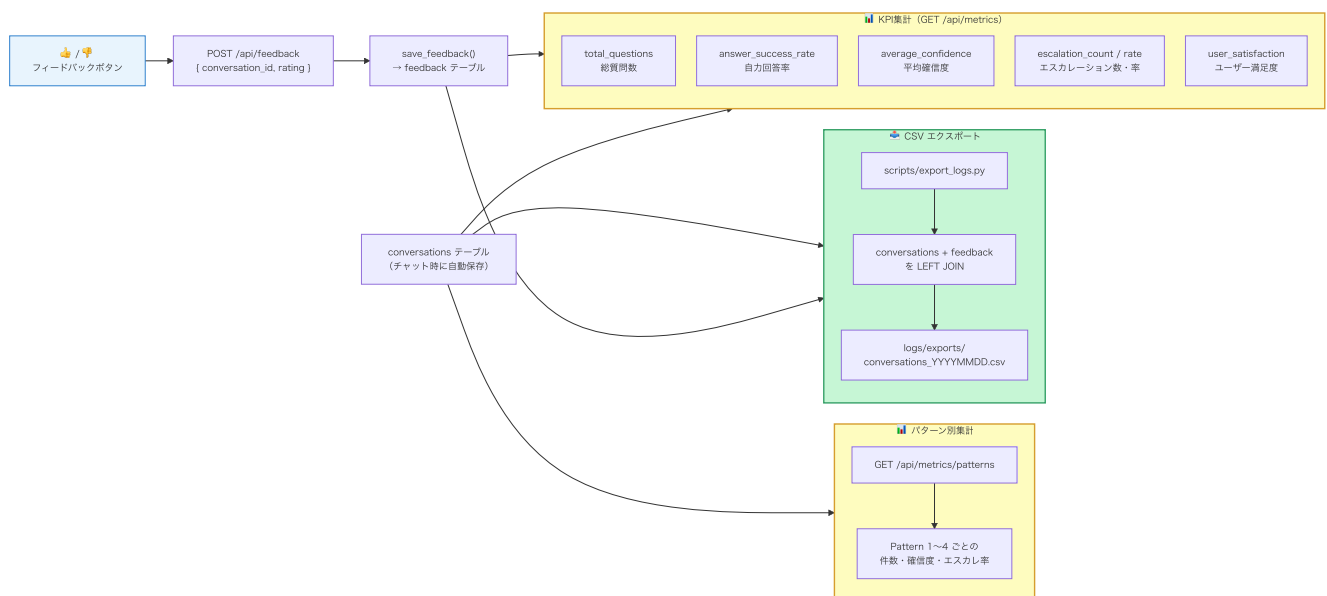
MarkdownナレッジベースがRAG検索でどう処理され、Claude APIに渡されるまで。



関連ファイル: `src/chat/rag.py` (KnowledgeLoader, SimpleRAG) `knowledge/` (ナレッジMarkdownファイル群)

6 フィードバック → KPI メトリクス → CSV エクスポート

ユーザー評価の記録から、KPI集計、定期エクスポートまでのデータ活用フロー。



7 ファイル × 処理ステップ 対応表

処理ステップ	担当ファイル	主要関数 / クラス	入力
UI表示・入力受付	templates/index.html static/js/chat.js	sendMessage()	ユーザー入力 キスト

処理ステップ	担当ファイル	主要関数 / クラス	入力
質問ナビ (フロント)	static/js/chat.js	renderGuideStep() buildGuideQuestion()	カテゴリ・サブトピック選択
質問ナビ (AI生成)	src/api/routes.py	guide_suggestions()	pattern, category
API受付・バリデーション	src/api/routes.py	chat()	ChatRequest (question, pattern, session_id)
会話履歴取得	src/database/operations.py	get_conversation_history()	session_id
RAG検索	src/chat/rag.py	SimpleRAG.search()	質問テキスト
エスカレーション判定	src/chat/engine.py	_check_escalation()	質問 + 検索結果
プロンプト構築	src/prompts/system_prompts.py	build_system_prompt()	pattern, persona_config
Claude API 呼び出し	src/chat/engine.py	client.messages.create()	system + messages + model + max_tokens
DB保存	src/database/operations.py	save_conversation() save_feedback() save_escalation()	各種データ
KPI集計	src/database/operations.py	calculate_metrics()	日付範囲 (任意)
CSVエクスポート	scripts/export_logs.py	export_to_csv()	SQLiteデータベース