omakase.ai
Technical Analysis Report

技術分析ドキュメント

Version 1.0

2025 年 12 月 6 日

HAR 解析に基づく VAPI/Daily.co プロトコル分析
競合比較・技術スタック評価

目次

1 エグゼクティブサマリー

omakase.ai は EC（電子商取引）サイト向けの音声 AI ショッピングアシスタントプラットフォームである。本レポートでは、プロダクション HAR ファイルの解析に基づき、技術スタック、外部依存関係、独自開発部分を詳細に分析する。

1.1 主要な発見

- **音声 AI 基盤**: VAPI（Voice AI Platform）に完全依存
- **通信基盤**: Daily.co の WebRTC SFU を使用
- **認証**: Clerk によるセッション管理
- **独自開発**: Widget UI、プロンプト設計、バックエンド統合

1.2 技術依存度

コンポーネント	提供元	依存度
音声認識 (STT)	VAPI/Deepgram	高
音声合成 (TTS)	VAPI/ElevenLabs	高
LLM	VAPI/OpenAI/Claude	高
WebRTC	Daily.co	高
認証	Clerk	中
Widget UI	独自開発	-
バックエンド API	独自開発	-

表 1: 技術依存マトリクス

2 システムアーキテクチャ

2.1 全体構成

omakase.ai は以下の 4 層構造で構成される：

1. **Presentation Layer**: React Widget
2. **Transport Layer**: Daily.co WebRTC
3. **AI Platform Layer**: VAPI
4. **Application Layer**: Express.js Backend

2.2 アーキテクチャ図

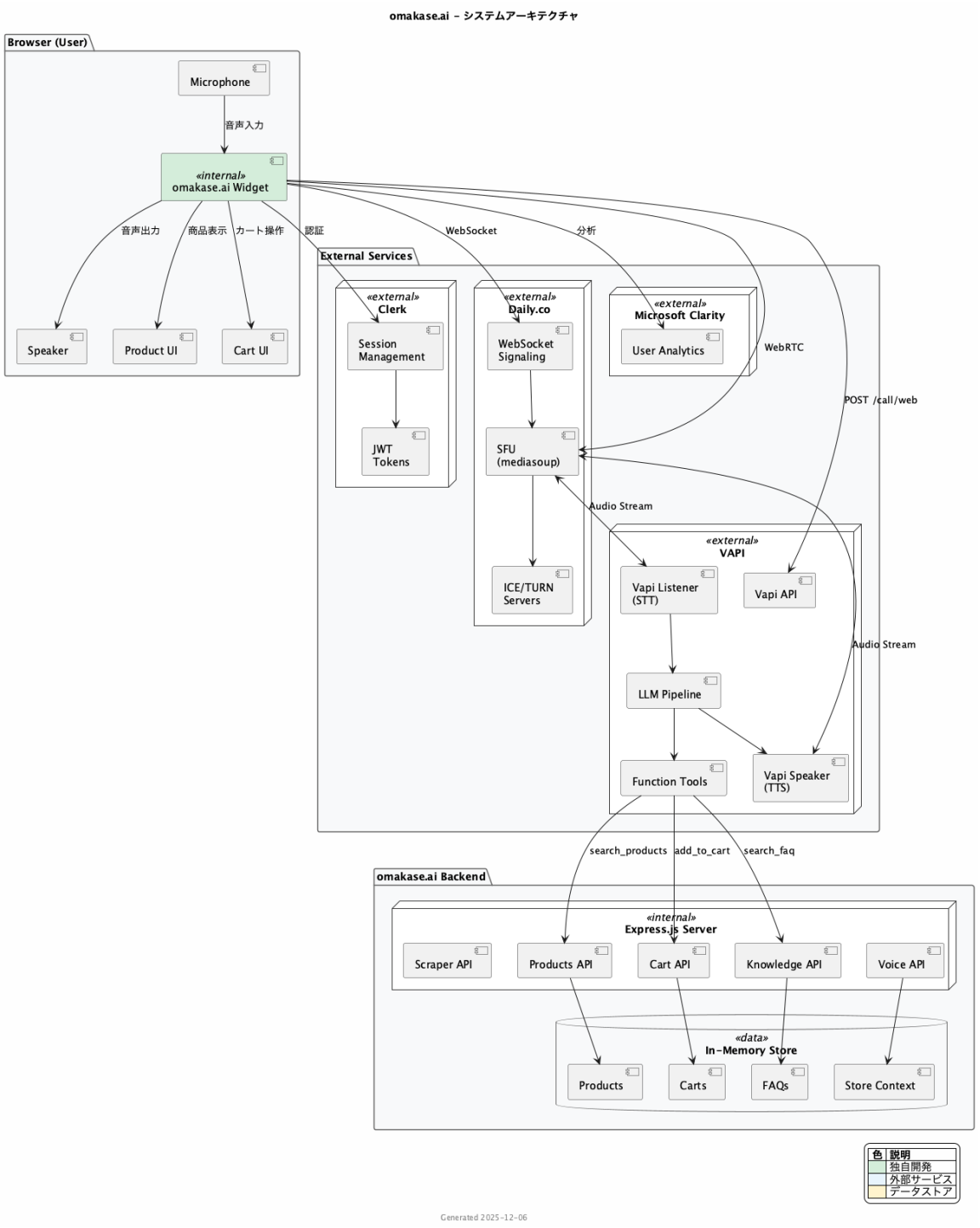


図 1: omakase.ai システムアーキテクチャ

2.3 データフロー

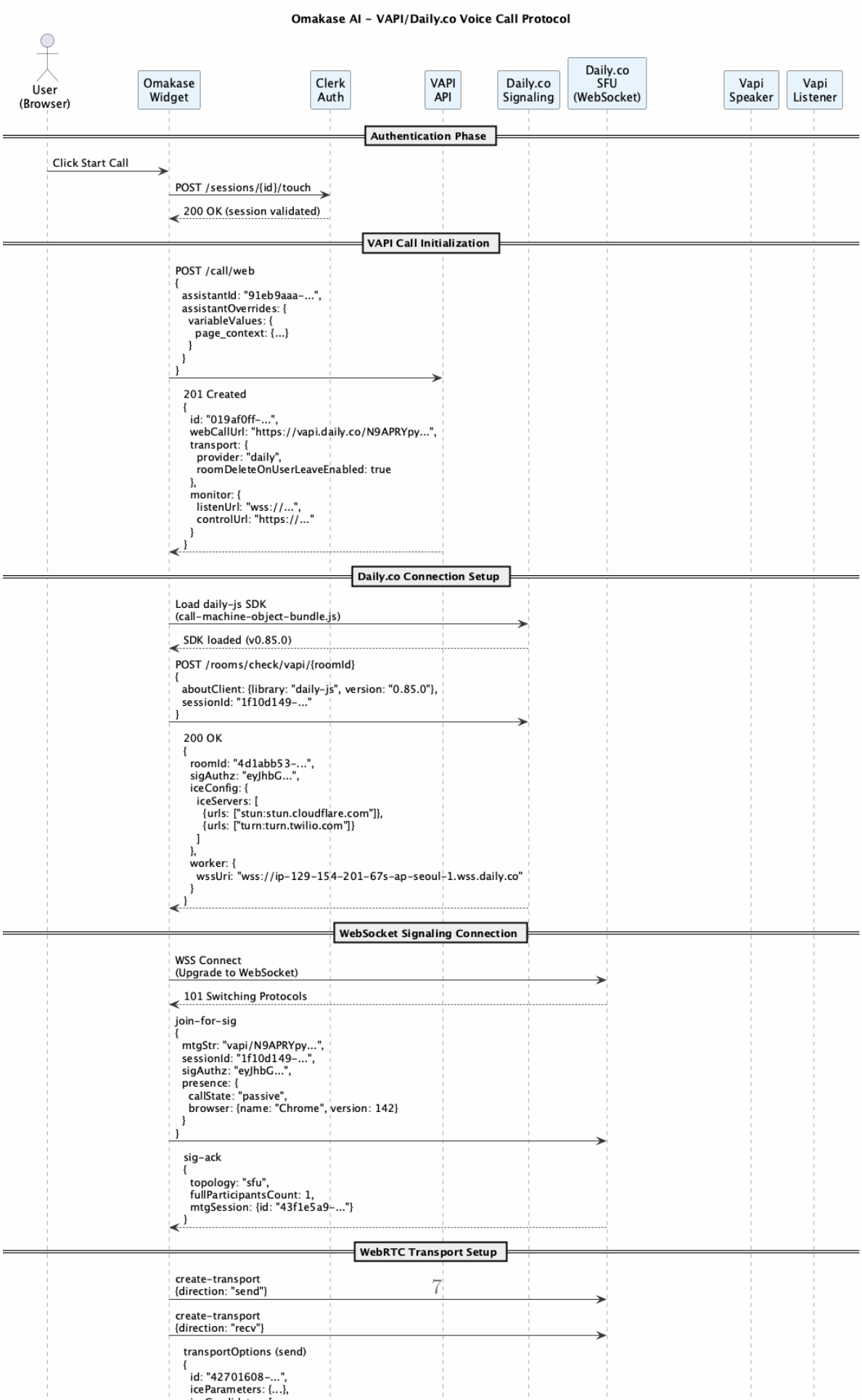
1. ユーザーが Widget で通話開始をクリック
2. Clerk でセッション JWT を検証

3. VAPI に通話開始リクエスト（assistantId + page_context）
4. Daily.co で WebRTC ルーム作成・接続
5. 音声ストリームが Vapi Listener/Speaker と双方向通信
6. VAPI が Function Tools でバックエンドを呼び出し

3　音声通話プロトコル

HAR 解析により、VAPI/Daily.co の詳細なプロトコルフローを特定した。

3.1 通話開始シーケンス



3.2 主要フェーズ

フェーズ	説明
認証	Clerk で `POST /sessions/{id}/touch` を呼び出し、セッション検証
VAPI 初期化	`POST /call/web` で assistantId と page_context を送信
Daily.co 接続	`POST /rooms/check/vapi/{roomId}` で ICE 設定取得
WebSocket	SFU への WebSocket 接続、`join-for-sig` メッセージ
WebRTC	`create-transport`、`connect-transport`、`send-track`
エージェント参加	Vapi Speaker/Listener が `sig-presence` で参加
音声会話	RTP パケットの双方向ストリーミング

3.3 技術仕様

- **プロトコル**: WebSocket + WebRTC
- **トポロジー**: SFU (Selective Forwarding Unit)
- **コーデック**: audio/opus, 48kHz
- **ICE**: Cloudflare STUN, Twilio TURN
- **参加者**: User, Vapi Speaker, Vapi Listener

4 外部サービス依存分析

4.1 VAPI (Voice AI Platform)

項目	詳細
エンドポイント	`https://api.vapi.ai/call/web`
assistantId	`91eb9aaa-17dc-4aa0-862b-e28fa21c0df4`
機能	STT, TTS, LLM, Function Calling
レイテンシ	500ms 未満
処理実績	150M+ 通話

表 3: VAPI 技術仕様

4.1.1 リクエスト例

```
POST /call/web
{
  "assistantId": "91eb9aaa-...",
  "assistantOverrides": {
    "variableValues": {
      "page_context": {
        "product_name": "...",
        "price": 2500,
        "stock": "available"
      }
    }
  }
}
```

4.2 Daily.co (WebRTC Infrastructure)

項目	詳細
シグナリング	`https://gs.daily.co/`
SFU WebSocket	`wss://ip-xxx.wss.daily.co`
グローバル拠点	75+
レイテンシ	13ms 中央値
SDK	daily-js v0.85.0

表 4: Daily.co 技術仕様

4.3 Clerk (Authentication)

項目	詳細
エンドポイント	`https://clerk.omakase.ai/`
機能	セッション管理, JWT 発行
トークン更新	約 45 秒間隔

表 5: Clerk 技術仕様

5 独自開発部分の分析

5.1 価値提供マトリクス

コンポーネント	独自性	差別化	説明
プロンプト設計	高	高	EC 販売特化シナリオ
Widget UI	高	中	ブランディング対応
バックエンド API	中	高	商品/カート連携
Function Tools	中	高	カート追加等
認証統合	低	低	Clerk 使用
音声 AI 基盤	低	低	VAPI 依存

表 6: 独自開発価値マトリクス

5.2 独自開発ファイル構造

```
frontend/src/
  components/
    preview-phone.tsx       # プレビュー表示
    agent-selector.tsx      # エージェント選択
    config-panel.tsx        # 設定パネル
    ec-context-form.tsx     # EC連携設定
  lib/
    api.ts                  # APIクライアント
    realtime.ts             # WebSocket処理

src/server/
  routes/
    products.ts             # 商品API
    voice.ts                # 音声API
    knowledge.ts            # ナレッジAPI
  services/
    store.ts                # データストア
    prompt-generator.ts     # プロンプト生成
```

6 競合分析

6.1 主要競合比較

機能	omakase.ai	VAPI	Pipecat	Rep AI	Retell
音声対話					
Web ウィジェット			-		-
EC 特化		-	-		-
日本語				?	
カート統合		-	-		-
OSS	-	-		-	-

表 7: 競合機能比較

6.2 ポジショニング

omakase.ai は「EC 特化」×「日本市場」の垂直特化ポジションを取る。汎用プラットフォーム（VAPI, Pipecat）との差別化は、EC 販売に最適化された会話設計とカート統合にある。

6.3 Pipecat 移行の可能性

メリット	デメリット
VAPI 料金削減	開発工数増加
カスタマイズ自由度	インフラ管理必要
ベンダーロックイン回避	保守負担増

表 8: Pipecat 移行の評価

7 推奨事項

7.1 短期 (3-6 ヶ月)

1. **プロンプト・シナリオ強化**: 業種別テンプレート、購買心理考慮
2. **分析ダッシュボード**: 会話分析、コンバージョン追跡
3. **A/B テスト基盤**: プロンプト・声・フローの最適化

7.2 中期 (6-12 ヶ月)

1. **Pipecat 検証**: PoC 実施、コスト比較
2. **データ蓄積**: 会話ログからの学習、レコメンド改善
3. **データベース導入**: In-Memory → PostgreSQL + Redis

7.3 長期 (1-2 年)

1. **自社音声 AI**: Fine-tuned モデル、独自 STT/TTS
2. **プラットフォーム化**: SDK 公開、マーケットプレイス

8 付録

8.1 参考ドキュメント

- `docs/API.md` - API 仕様書
- `docs/omakase-ai-protocol.puml` - プロトコルシーケンス
- `docs/architecture.puml` - アーキテクチャ図
- `docs/TECHNICAL_ANALYSIS.md` - 技術分析詳細
- `docs/COMPETITIVE_ANALYSIS.md` - 競合分析詳細

8.2 外部リンク

- VAPI: https://vapi.ai/
- Daily.co: https://www.daily.co/
- Pipecat: https://docs.pipecat.ai/
- Clerk: https://clerk.com/

— End of Document —