

omakase.ai 技術分析

VAPI/Daily.co プロトコル解析レポート

Engineering Team

Confidential

2025 年 12 月 6 日

アジェンダ

1. 概要 - omakase.ai とは
2. 技術スタック - 外部サービス構成
3. プロトコル詳細 - 通話フロー解析
4. 独自開発部分 - 差別化ポイント
5. 競合分析 - 市場ポジション
6. 改善提案 - Pipecat 移行計画

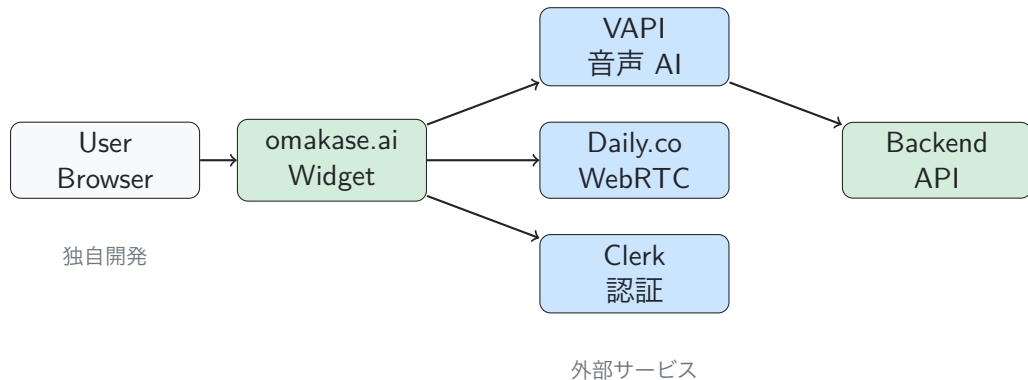
EC 特化の音声 AI ショッピングアシスタント

- ▶ 商品ページに埋め込む Widget を提供
- ▶ 音声で商品案内・カート操作
- ▶ 自然言語での質問応答
- ▶ 多言語対応（日本語中心）

ターゲット: EC 事業者



外部サービス構成



サービス	役割	依存度
VAPI	音声 AI (STT/LLM/TTS)	高
Daily.co	WebRTC SFU	高
Clerk	認証・セッション	中

VAPI - Voice AI Platform

提供機能

- ▶ 音声認識 (STT)
- ▶ 音声合成 (TTS)
- ▶ LLM 統合 (GPT/Claude)
- ▶ Function Calling
- ▶ 割り込み処理

実績

- ▶ 150M+ 通話処理
- ▶ 350K+ 開発者
- ▶ 500ms 未満レイテンシ

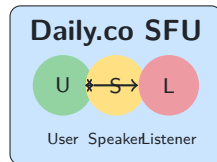
API 呼び出し例

```
POST /call/web
{
  "assistantId": "91eb...",
  "assistantOverrides": {
    "variableValues": {
      "page_context": {}
    }
  }
}
```

Daily.co - WebRTC Infrastructure

提供機能

- ▶ SFU (Selective Forwarding Unit)
- ▶ グローバル 75+ 拠点
- ▶ 13ms 中央値レイテンシ
- ▶ ICE/TURN サーバー



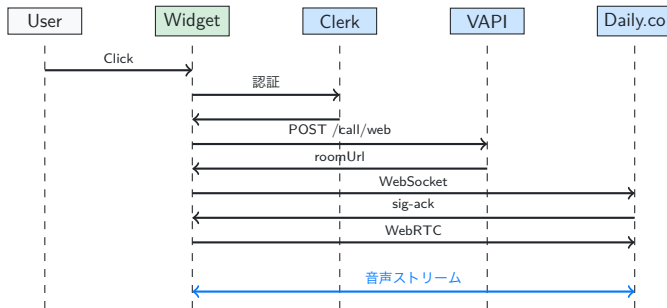
omakase.ai での使用

- ▶ VAPI が内部で使用
- ▶ daily-js SDK v0.85.0
- ▶ Opus 48kHz 音声コーデック

3 参加者構成:

- ▶ User (ブラウザ)
- ▶ Vapi Speaker (応答)
- ▶ Vapi Listener (聞き取り)

通話開始シーケンス (概要)



プロトコルフェーズ

#	フェーズ	詳細
1	認証	Clerk POST /sessions/{id}/touch
2	VAPI 初期化	POST /call/web + assistantId
3	Daily.co 接続	POST /rooms/check/vapi/{roomId}
4	WebSocket	SFU へ接続、join-for-sig
5	WebRTC	create-transport, send-track
6	エージェント参加	Vapi Speaker/Listener が sig-presence
7	音声会話	RTP パケット双方向ストリーミング
8	トークン更新	45 秒ごとに Clerk JWT 更新

WebRTC 詳細

トランスポート設定

- ▶ 送信 (send) + 受信 (recv)
- ▶ ICE: Cloudflare STUN, Twilio TURN
- ▶ DTLS 暗号化

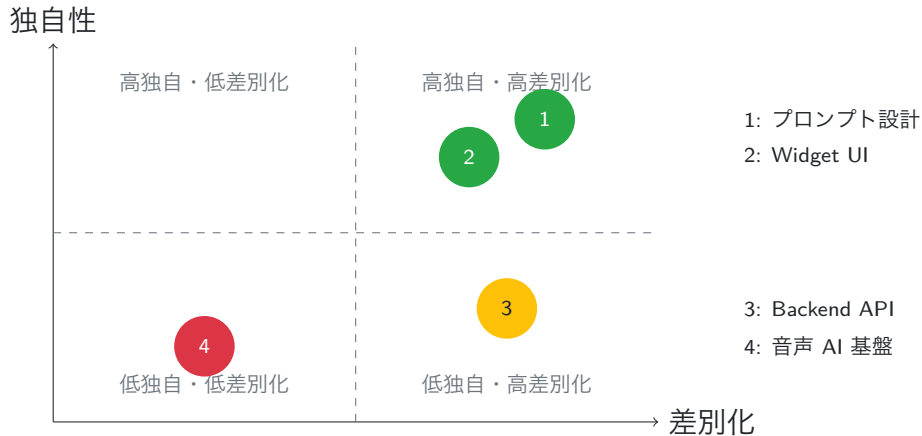
音声トラック

- ▶ コーデック: Opus 48kHz
- ▶ メディアタグ: cam-audio
- ▶ SSRC 割り当て

send-track メッセージ

```
{
  "transportId": "427...",
  "kind": "audio",
  "rtpParameters": {
    "codecs": [{
      "mimeType": "opus",
      "clockRate": 48000
    }]
  }
}
```

独自開発 vs 外部依存



独自開発コンポーネント

1. プロンプト設計 ✓

- ▶ EC 販売特化シナリオ
- ▶ 商品コンテキスト注入
- ▶ 日本語最適化

2. Widget UI ✓

- ▶ ブランドカスタマイズ
- ▶ 商品ページ連携
- ▶ カート UI 統合

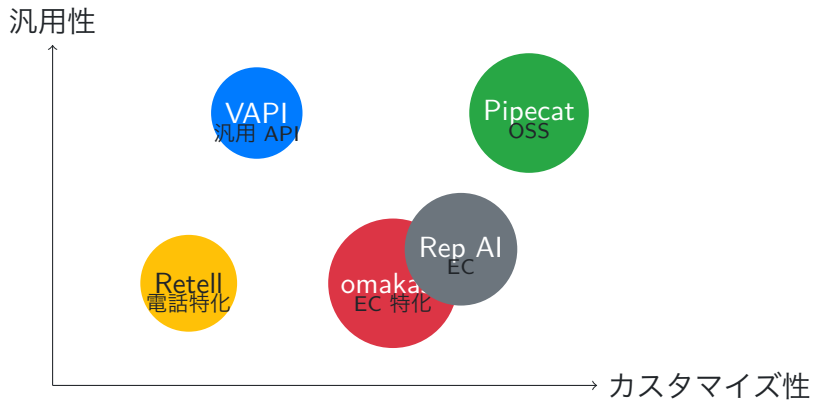
3. Backend API ✓

- ▶ 13 Function Tools
- ▶ 商品/カート/注文 API
- ▶ ナレッジベース

4. 音声 AI 基盤 ✕

- ▶ VAPI 完全依存
- ▶ カスタマイズ制限
- ▶ コスト課題

競合マップ



omakase.ai のポジション: EC 特化 × 日本市場 × 音声 AI

機能比較

機能	omakase	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	✗	✗	✓	✗	✗

課題と改善策

現状の課題

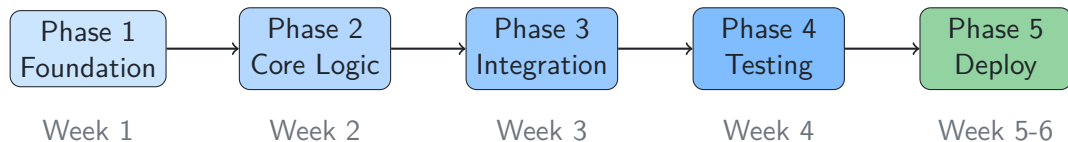
1. ⚠️ VAPI 完全依存
2. ⚠️ コスト構造
3. ⚠️ カスタマイズ制限
4. ⚠️ ベンダーロックイン

改善策: Pipecat 移行

1. ✅ 自社管理の音声 AI
2. ✅ コスト 50% 削減
3. ✅ 完全カスタマイズ
4. ✅ マルチベンダー対応



Pipecat 移行計画



Phase	内容	成果物
1	Pipecat 基本パイプライン	PoC 動作確認
2	Function Tools + Backend 連携	13 ツール動作
3	Widget 統合 + Feature Flag	切り替え可能
4	負荷テスト + 最適化	<600ms レイテンシ
5	Staging → Production	本番稼働

期待効果

コスト削減

- ▶ VAPI 料金: **-100%**
- ▶ 新規コスト: Deepgram + ElevenLabs
- ▶ **純削減: 30-50%**

パフォーマンス

- ▶ レイテンシ: 800ms → **<600ms**
- ▶ 中間サービス削減

カスタマイズ

- ▶ プロンプト完全制御
- ▶ STT/TTS 選択自由
- ▶ 独自機能追加可能

リスク軽減

- ▶ ベンダーロックイン解消
- ▶ 障害時の切り替え可能

まとめ

1. **現状分析:** VAPI/Daily.co に高度依存、独自価値は UI/プロンプト
2. **プロトコル:** HAR 解析で詳細フロー特定完了
3. **競合:** EC 特化 × 日本市場でのニッチポジション
4. **提案:** Pipecat 移行でコスト削減・カスタマイズ強化

Next Steps

1. Pipecat PoC 実施 (Week 1)
2. コスト比較・検証
3. 移行判断

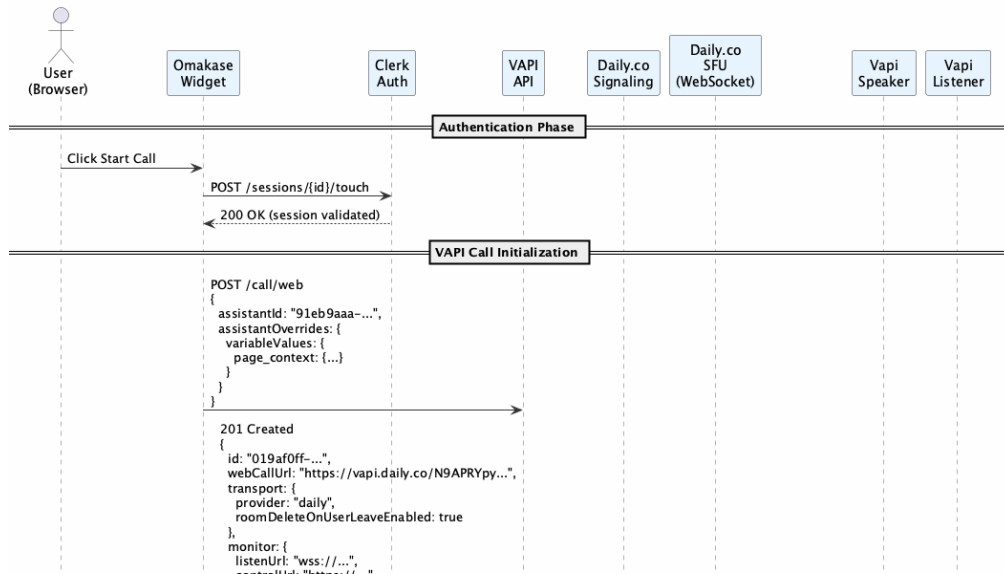
Thank You

Questions?

Confidential - Internal Use Only

Appendix: プロトコルシーケンス図

Omakase AI - VAPI/Daily.co Voice Call Protocol



Appendix: アーキテクチャ図

omakase.ai - システムアーキテクチャ

