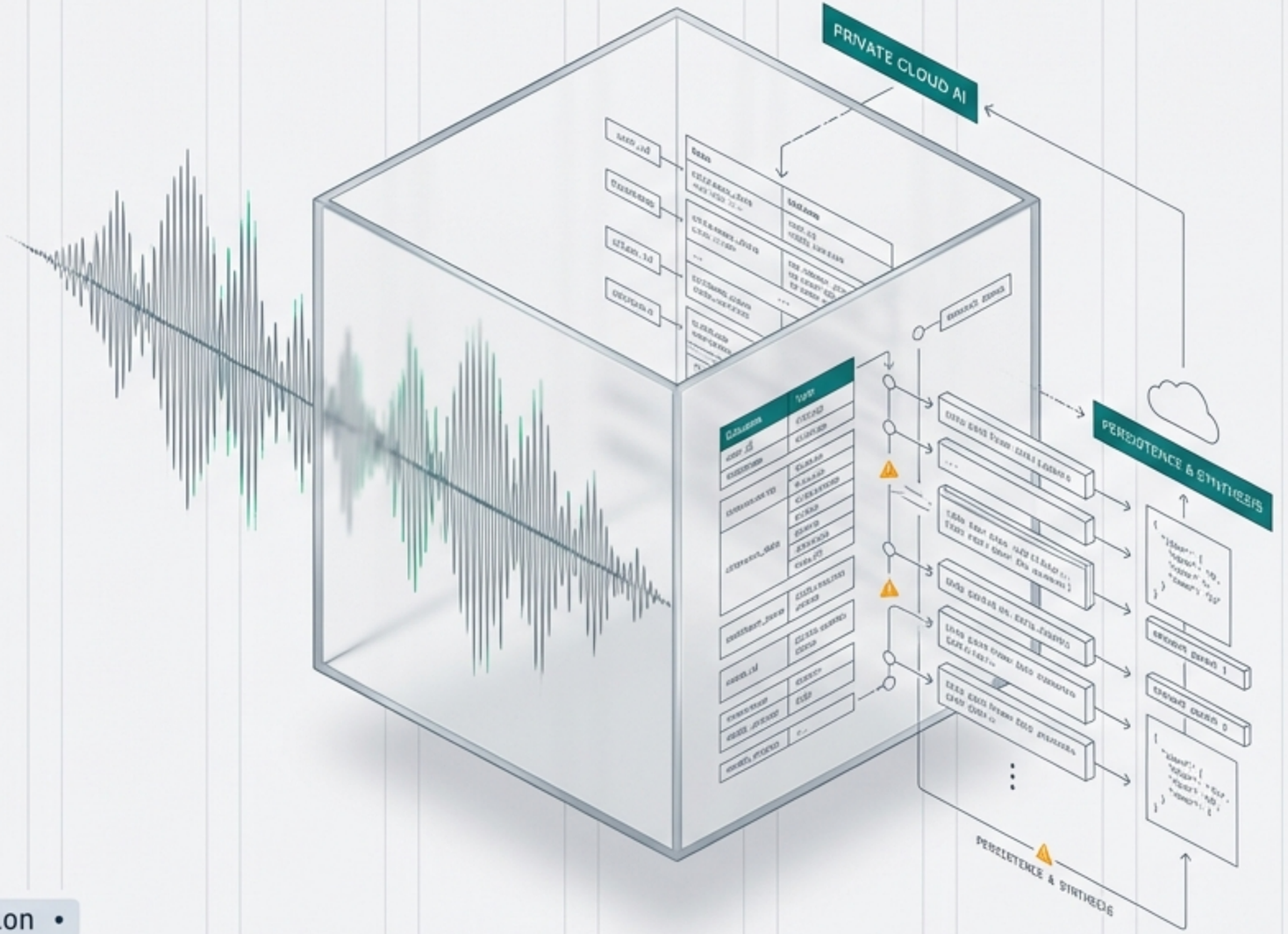
HPE
HPE AI Voice Agent v4.0

System Architecture & Capabilities

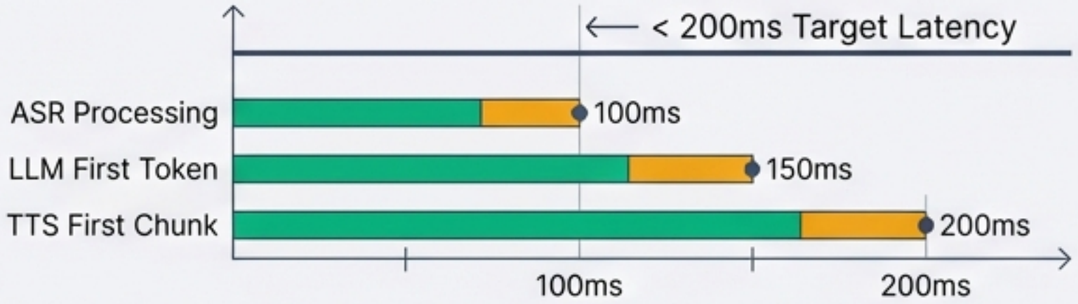
A technical dissection of an end-to-end voice pipeline powered by HPE Private Cloud AI.



`Frontend VAD • WebSocket Pipeline • LLM Integration • PostgreSQL Persistence • XTTS v2 Synthesis`

Engineering Targets & System Capabilities

Real-Time Latency



Metrics tracking for ASR Processing, LLM First Token, TTS First Chunk.

Data Persistence

Transactional CRUD

- Full `asyncpg` connection pooling to `PostgreSQL`.
- Create, Read, Update, Delete tickets & invoices.

Multilingual Core

Global Language Support

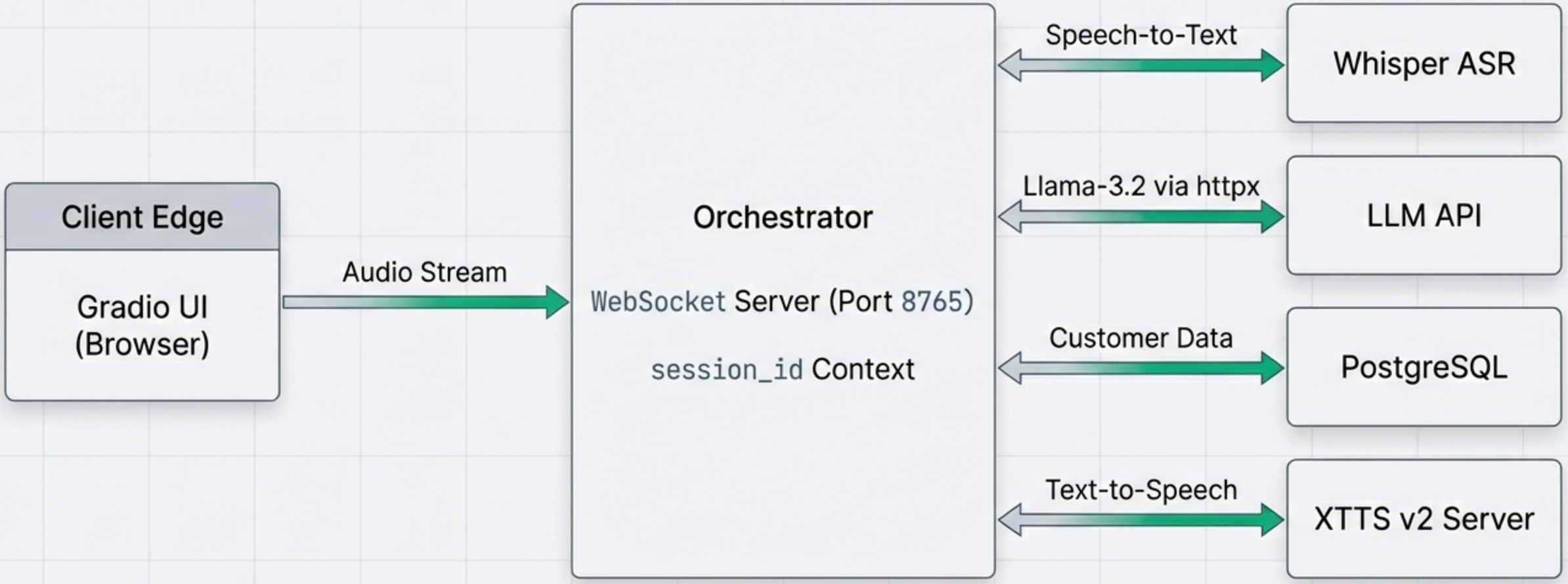
- 99+ Input Languages (Whisper ASR).
- 16+ High-Fidelity Output Languages (XTTS v2).

Resiliency

v4.0 Architecture

- Circuit Breakers. Connection Watchdogs.
- Service Health Checks to prevent cascading failures.

The Anatomy of a Voice Transaction



The pipeline uses a persistent WebSocket connection to stream audio, maintain session context, and push real-time visual updates back to the client.

Frontend Logic: Voice Activity Detection (VAD)


```
// Adaptive Threshold Calculation
const newThreshold = Math.round(noiseFloor * 2 + 8);
window.vad.threshold = newThreshold;

// Barge-In Logic
function handleBargeIn() {
  if (window.vad.currentAudio && !window.vad.currentAudio.paused) {
    console.log('Barge-in detected - stopping AI audio');
    window.vad.currentAudio.pause();
    window.vad.currentAudio.currentTime = 0;
    // Reset state to listening
    window.vad.paused = false;
  }
}
```

Technical Blueprint meets High-End Editorial

The Intelligence Layer: Personas & Prompts

Smart Assistant

Constraint: Keep it SHORT - 1-3 sentences max.

Tone: Natural, intelligent.

```
"template": '''You're having a
natural voice conversation.
Be warm, smart, and genuinely
helpful.
Reply naturally:'''
```

Customer Service Pro

Constraint: Use customer's name. Sound real.

Tone: Warm, efficient.

```
"template": '''You're a
friendly agent on a live call.
Get to the point quickly.
What would a great service rep
say?'''
```

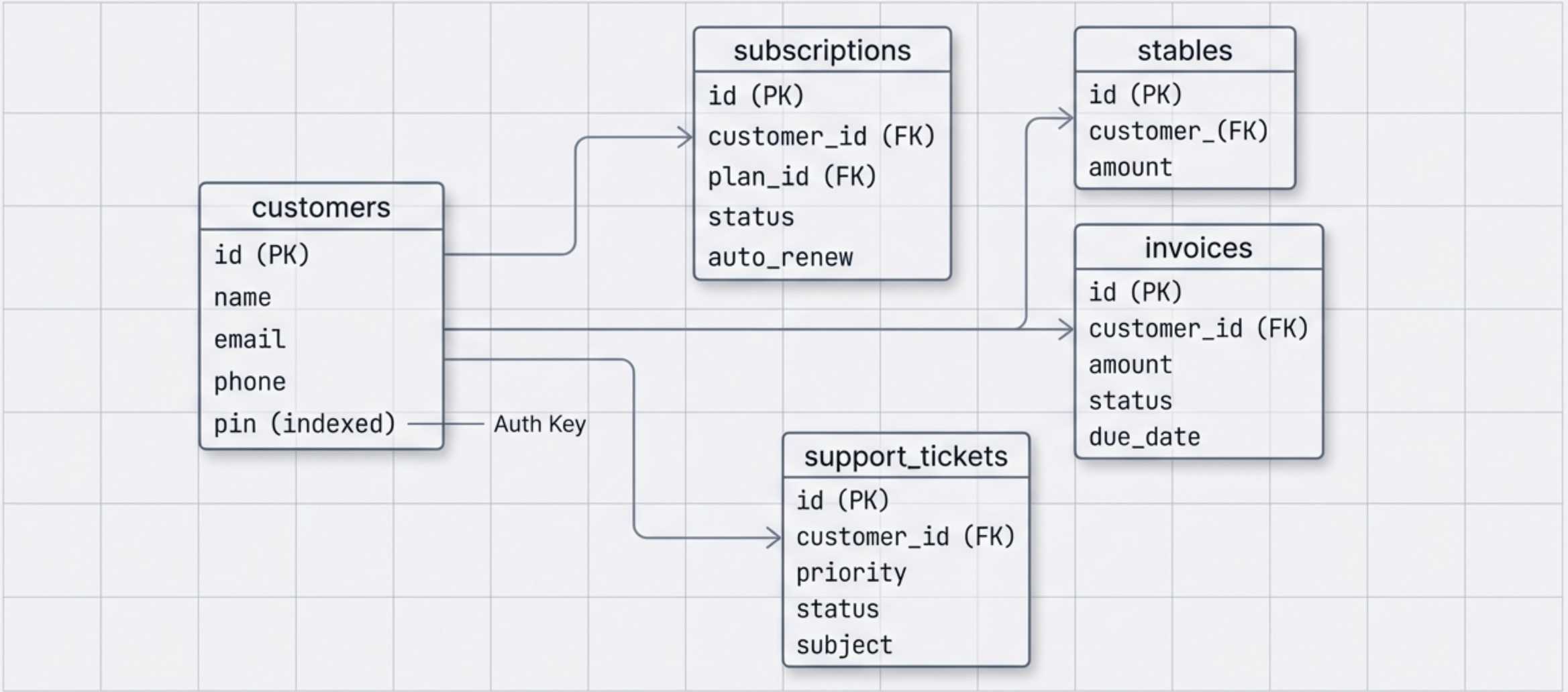
Language Injection Logic

System appends instructions based on output language selection.

```
# If target is 'en' (English)
"instruction": "Even if user
speaks Hebrew, reply in
English."

# If target is 'it' (Italian)
"instruction": "DEVI rispondere
SOLO in italiano."
```

Database Integration: Relational Schema



Transactional integrity ensured via indices on `'pin'`, `'status'`, and `'customer_id'` for sub-millisecond lookups.

Real-Time Data Actions (CRUD)

Context Injection (Read)

```
def get_extended_customer_info(
db_conn, customer_id):
  # Fetch overdue invoices to
prep the LLM
  sql = "SELECT count(*) FILTER
(WHERE status = 'overdue')..."
```

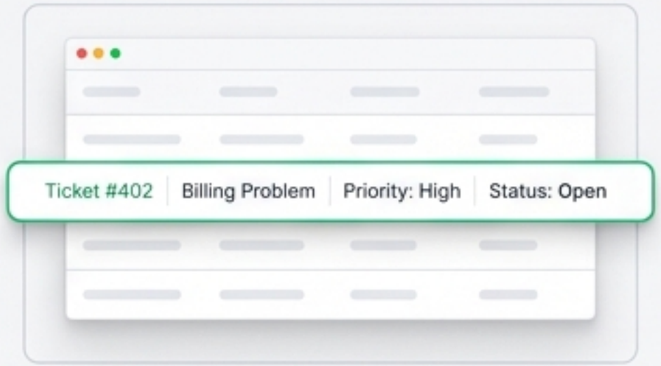
Context: You have 2 overdue invoices.

Action Execution (Write)

```
def add_ticket(customer_id,
subject, priority):
  # User says: "I have a billing
problem"
  sql = "INSERT INTO
support_tickets... RETURNING id"
```

Ticket #402 created.

UI Feedback



Advanced Analytics: Sentiment & Churn Detection

User Input

User: I am very frustrated with this service!

sentiment_alerts	
sentiment_score	0.15 (Negative)
churn_risk	HIGH
trigger_phrases	frustrated
recommended_action	Offer Discount

Supervisor Dashboard

⚠

New High-Risk Alert:
User ID 12345.
Sentiment: Negative (0.15).
Action Required.

Proactive Logic: The system logs 'High Risk' interactions into a persistent alert queue, enabling human intervention.

High-Fidelity Output: XTTS v2 Server

Engine Specs

- Sampling Rate: 24kHz
- Output Format: 16-bit PCM
- Latency Optimization: Preloaded models & cached embeddings.

Multilingual Routing

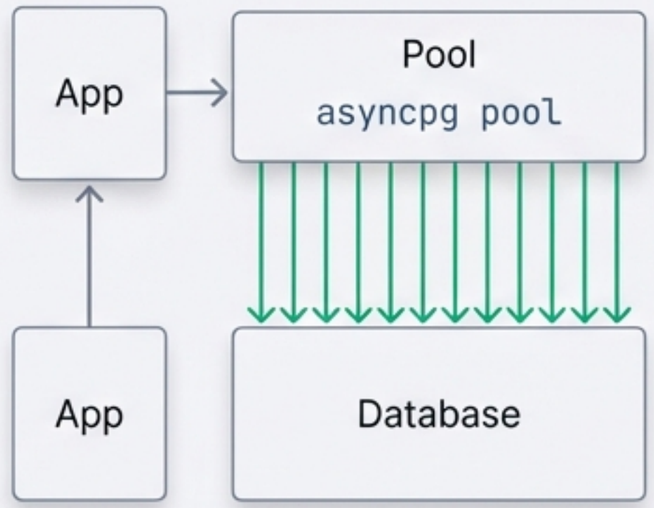
```
TTS_LANGUAGES = {
  "zh-cn": {"display": "Chinese",
    "instr": "Use Chinese..."
  },
  "es": {"display": "Spanish",
    "instr": "Responder en español..."
  }
}
```

Voice Cloning


```
# Endpoint: /tts/clone
xtts_manager.synthesize(
  text=text,
  speaker_wav=reference_audio
)
```

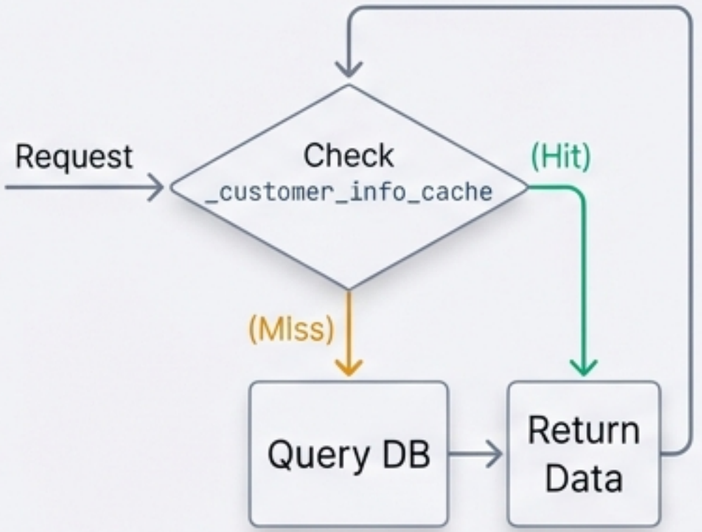
Performance Engineering: **v4.0 Optimizations**

Connection Pooling



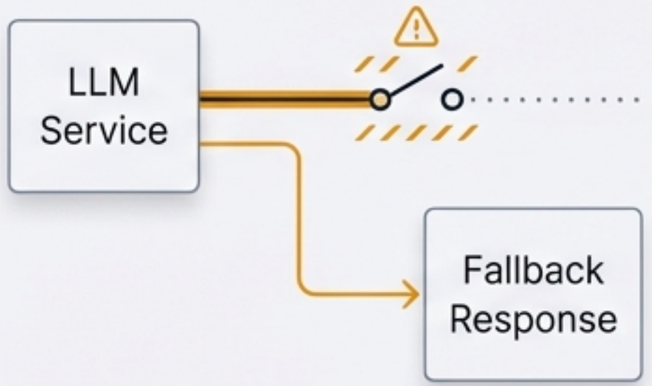
Reuses TCP/SSL connections.
Min 1, Max 10 connections.

Smart Caching



TTL (Time-To-Live) implementation.
Reduces DB hits during rapid conversation.

Circuit Breakers



Prevents cascading failures when external APIs (LLM/ASR) time out.

Security & Session State Management

Authentication

PIN Verification against `customers` table.

```
SELECT name FROM customers
WHERE pin = '1234'
```

Session Isolation

Unique Session Hash generation.

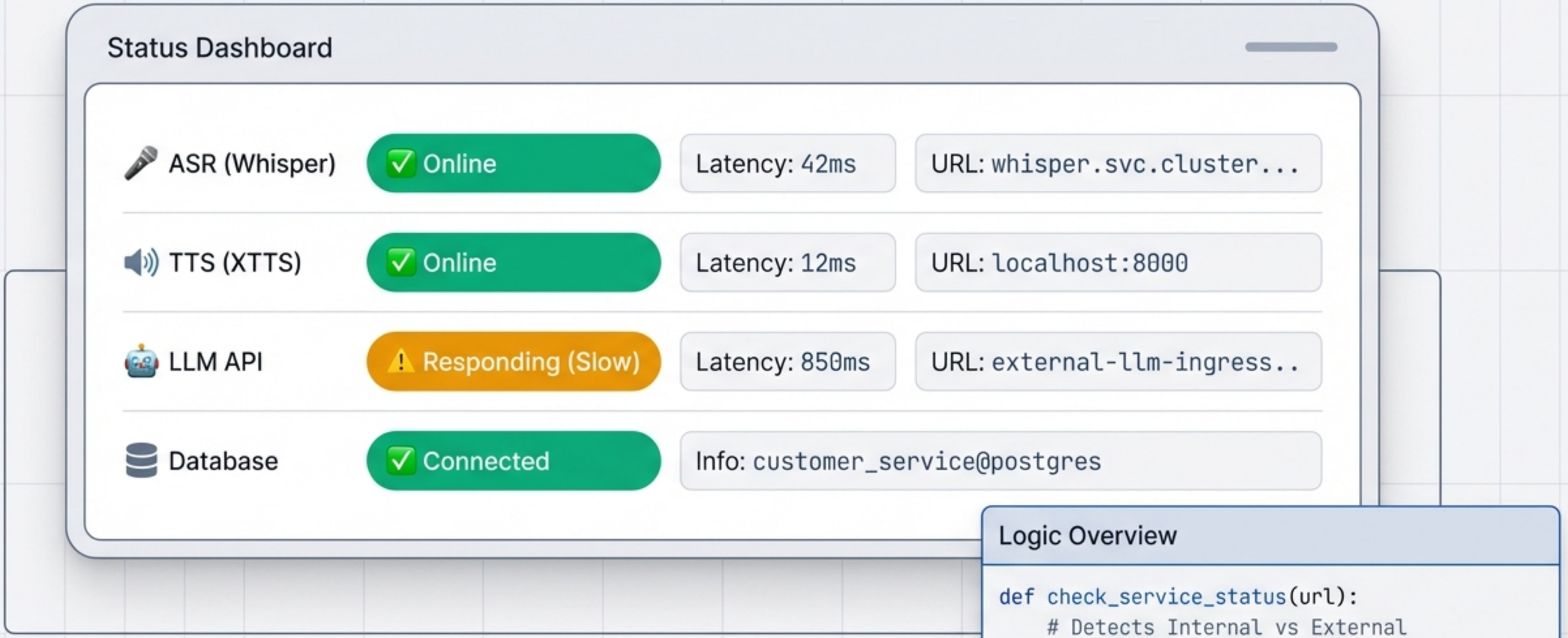
```
hashlib.md5(session_base.encode())
```

Environment Security

Credential Injection via `os.getenv`.

SSL Verification options (`urllib3.disable_warnings` for internal clusters).

Observability: Integrated Health Monitoring



Enterprise-Ready Architecture

- **Full-Stack Implementation**
  React-like frontend logic (Gradio), Python backend, SQL persistence.
- **Resilient & Robust**
  Connection pooling, Smart Caching, and Circuit Breakers.
- **Interactive & Responsive**
  Real-time visualizers, Barge-in capability, Sub-second latency.
- **Action-Oriented**
  Transactional capabilities: Reading/Writing Invoices & Tickets.

HPE AI Voice Agent v4.0 moves Voice AI beyond simple Q&A into a transactional interface for business logic.