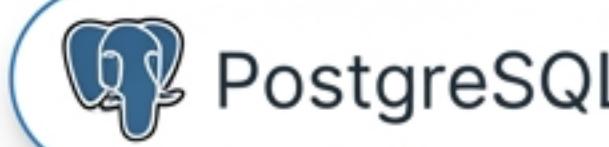
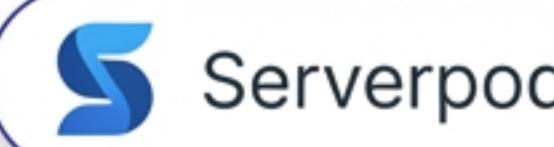


Recall Butler

Your AI-powered memory assistant.

Dump anything now, retrieve and act later.



Built for Serverpod Hackathon 2026

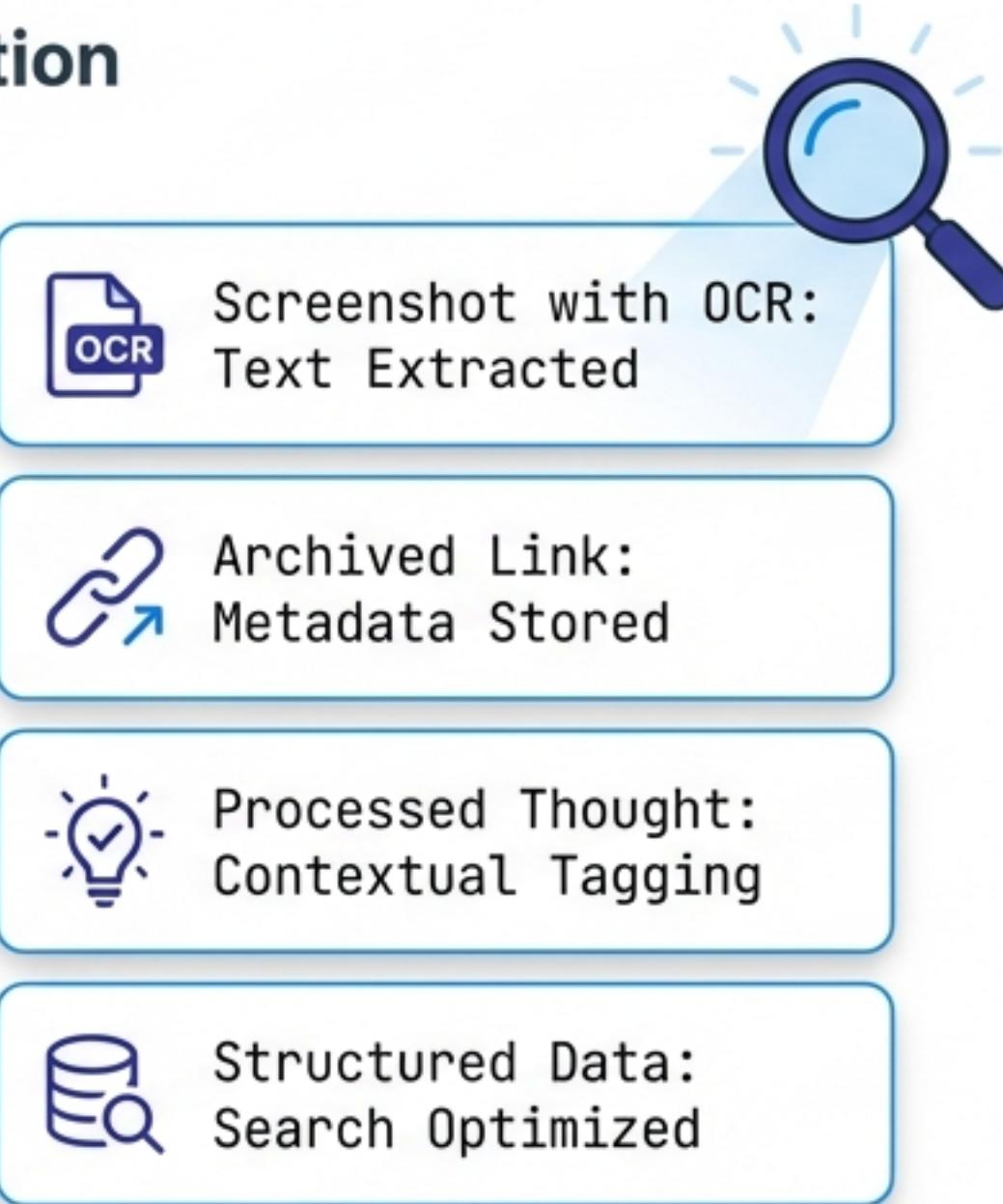
Turning Digital Noise into Actionable Intelligence

The Challenge



Our digital lives are cluttered with fragmented inputs—unorganized screenshots, saved links, and fleeting thoughts. Retrieval is manual, slow, and often fails.

The Solution



Recall Butler transforms unorganized input into a searchable memory bank. It doesn't just store data; it understands it.

Goal: Move from 'Dump & Forget' to 'Dump & Act'.

Three Pillars of the Digital Butler



1. Smart Ingest

Capture everything. Upload files (PDFs, Images), paste clipboard text, or save URLs. The system handles immediate processing.



2. Semantic Recall

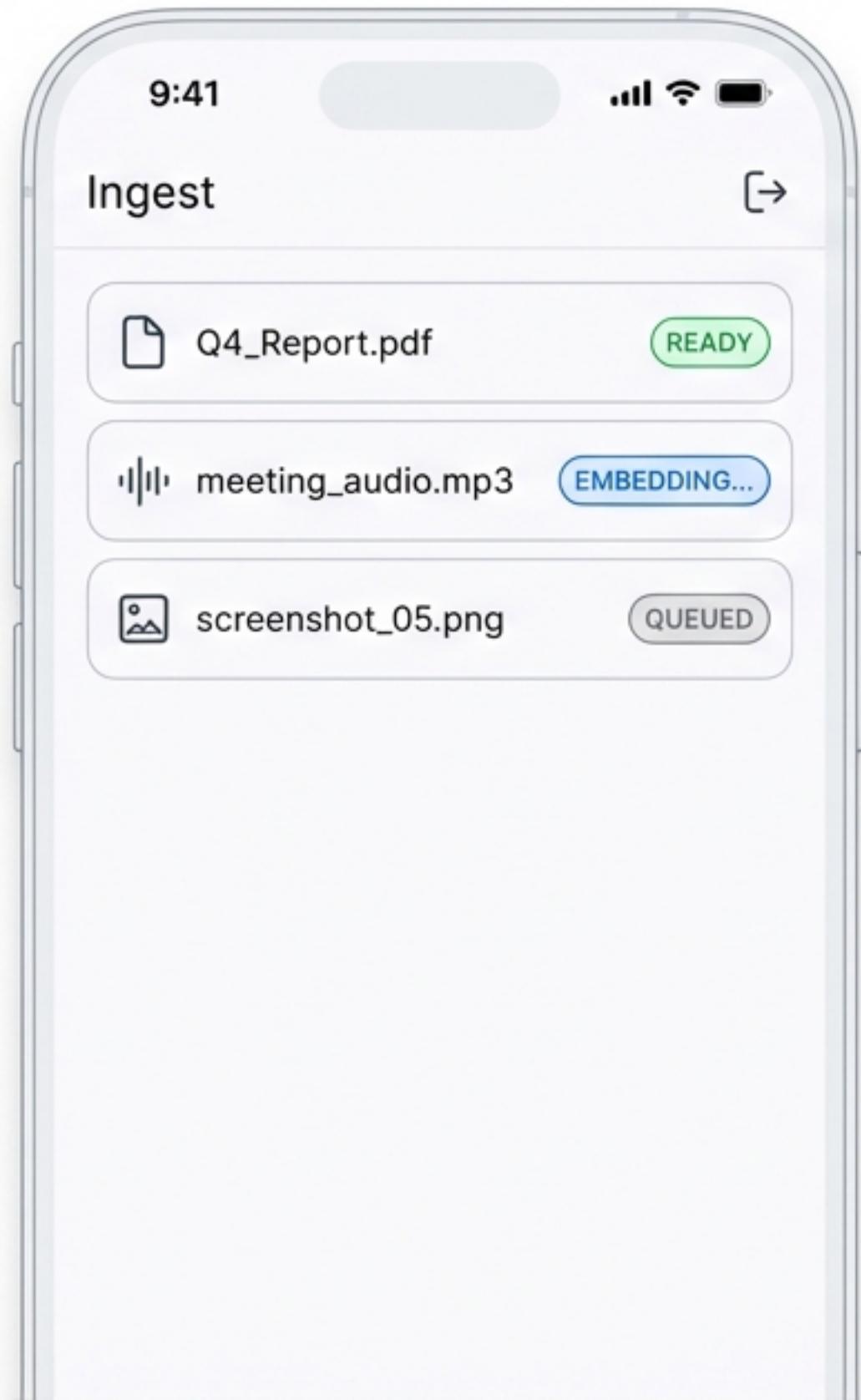
Ask questions in natural language. The system searches across all memories to provide answers grounded in your specific data, not general knowledge.



3. Butler Actions

Proactive intelligence. The AI analyses content types to suggest utility—turning invoices into reminders and meetings into follow-up tasks.

Smart Ingest & Real-Time Feedback Loop



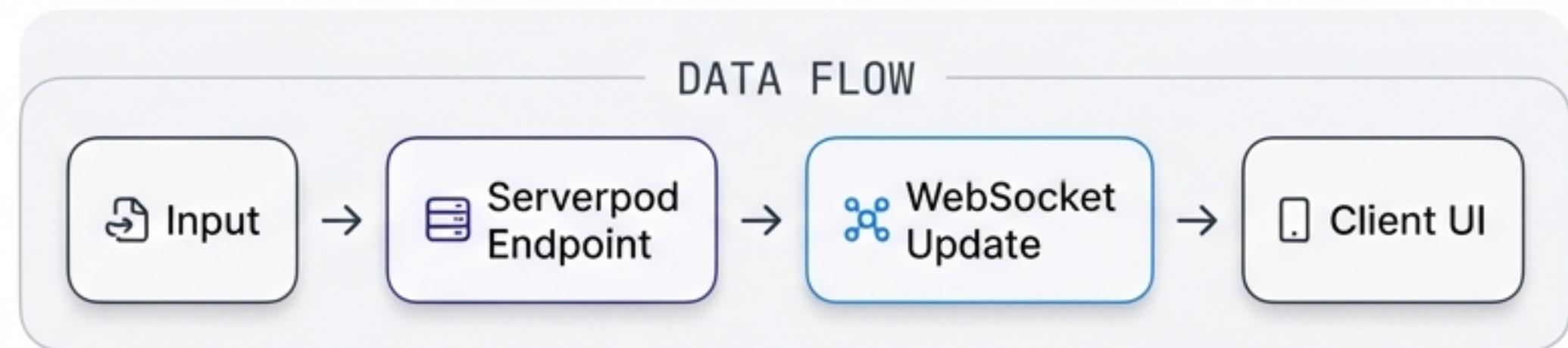
Versatile Input

Supports PDFs, Documents, Images, Clipboard text, and URL extraction.

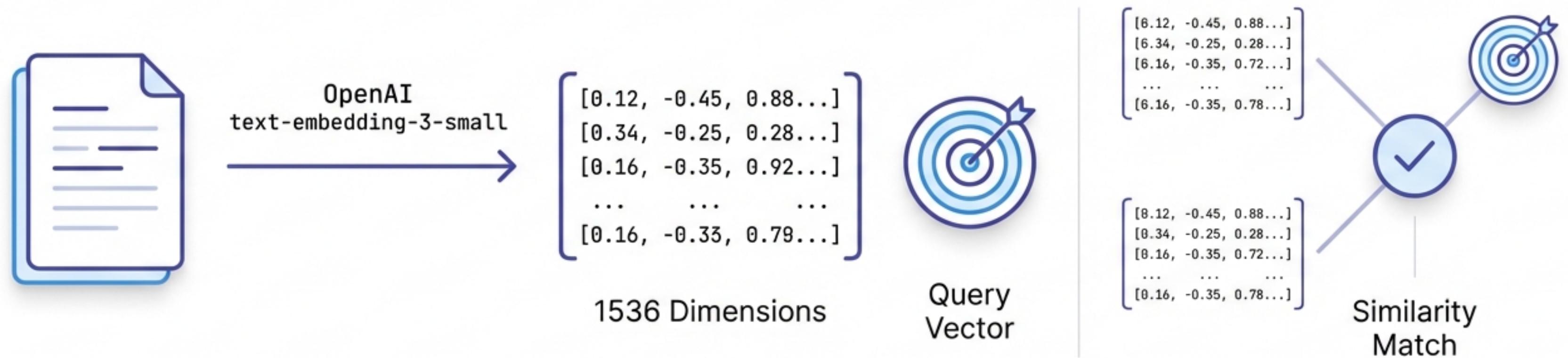
The Feedback Loop

Users are never left guessing. Status transitions:

QUEUED → **EXTRACTING** → **EMBEDDING** → **READY**.



Semantic Recall: Search by Meaning, Not Keywords



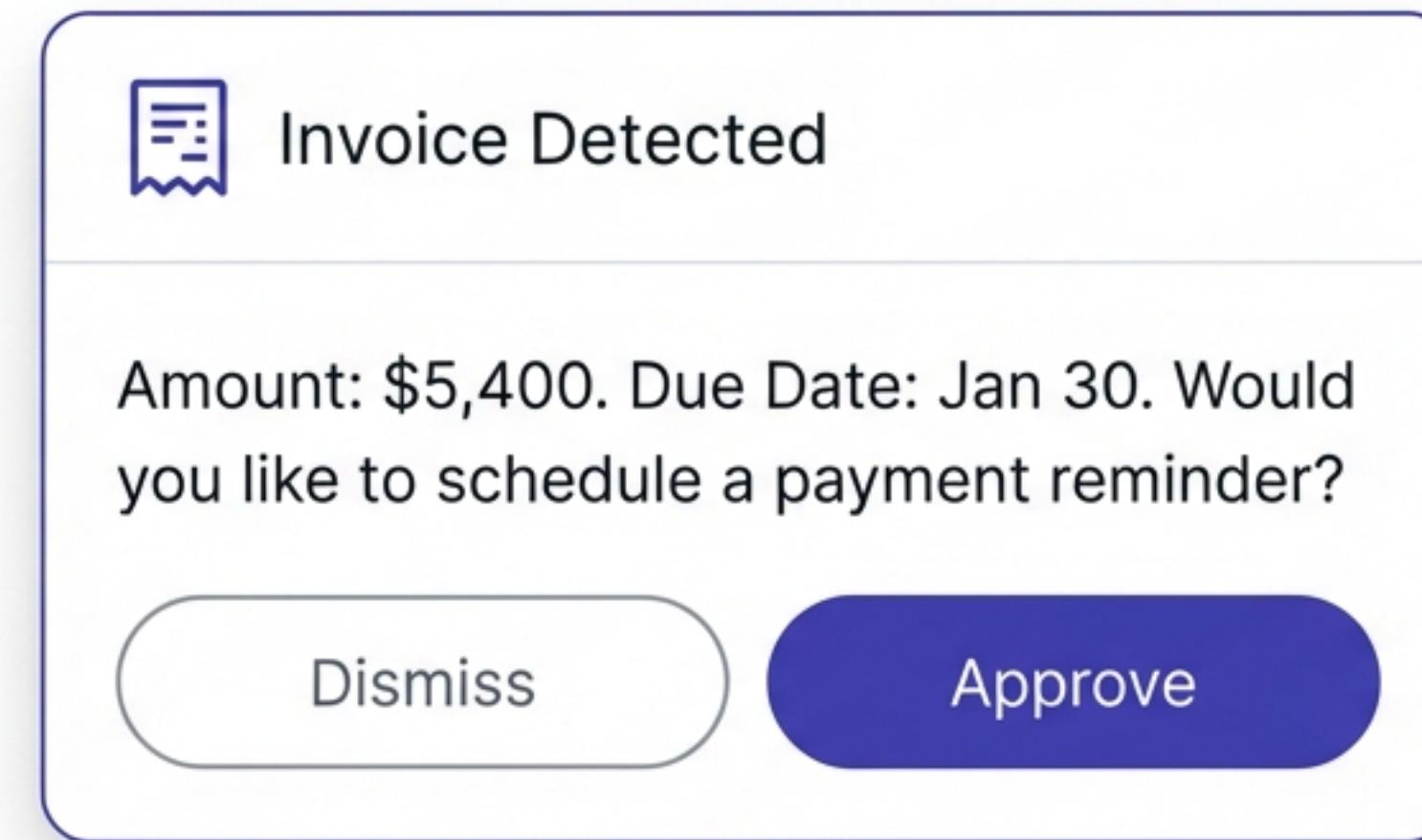
Grounded Answers:

The AI returns the top 3 source snippets with relevance scores. It answers based *only* on sites. It answers based *only* on these snippets to prevent hallucinations.

→ Navigation:

Includes one-click jumps to the original source document.

Butler Actions: Proactive Intelligence



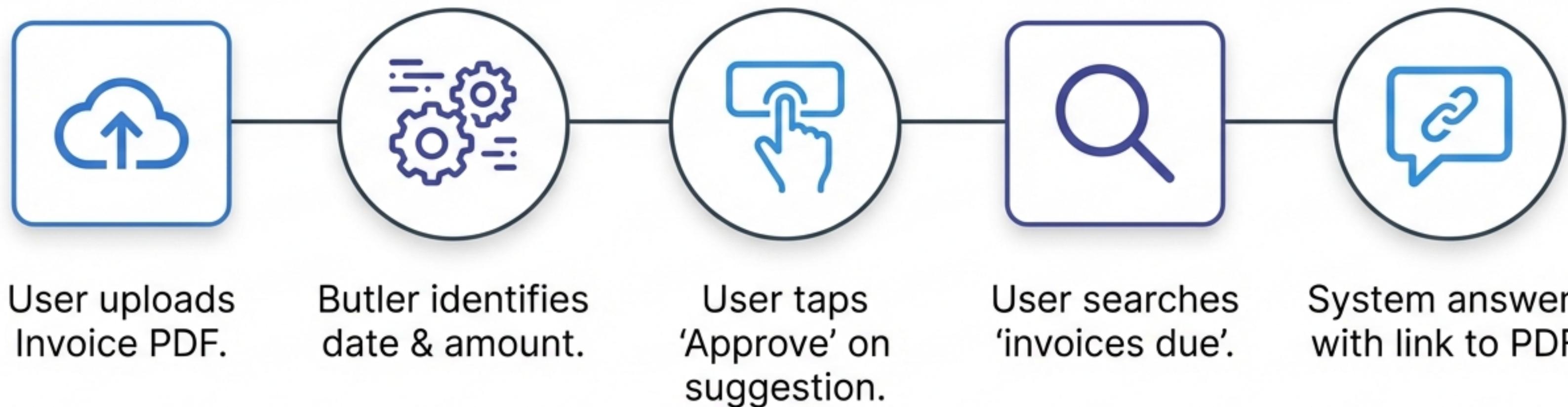
Proactive Intelligence: The system analyses the content ***type*** to suggest utility.

✉️ Invoice → Payment reminder 🎙️

✈️ Itinerary → Check-in alert 🚲

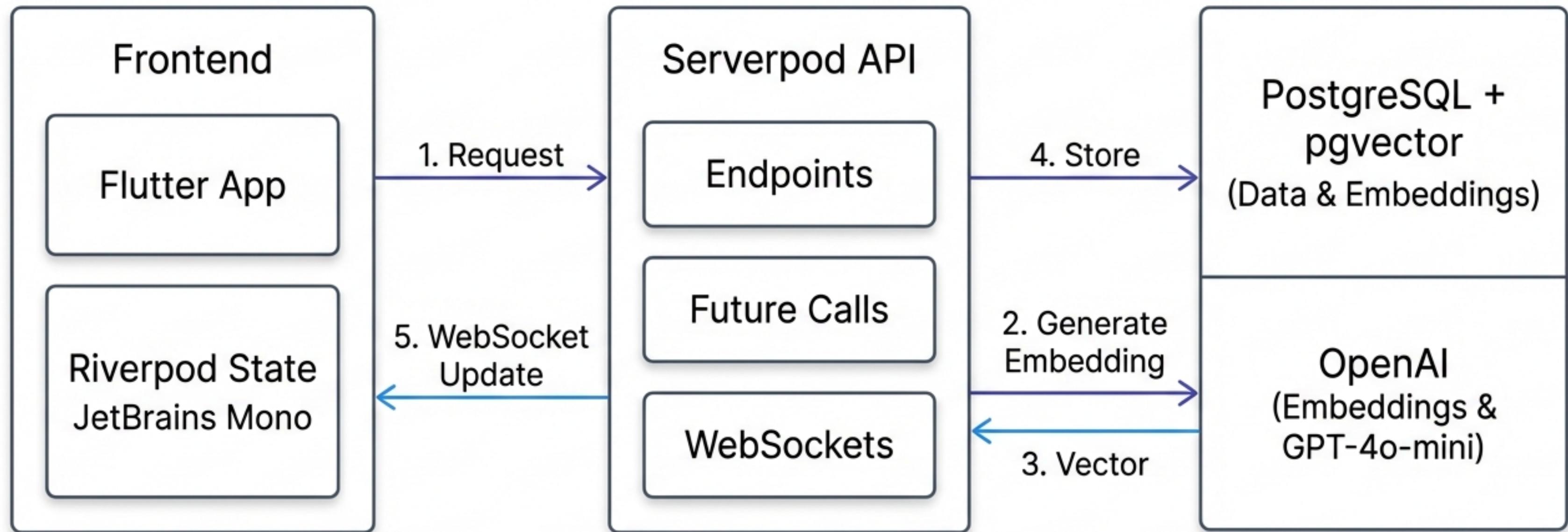
📝 Meeting Notes → Follow-up list 📋

Demo Flow: The Invoice Journey



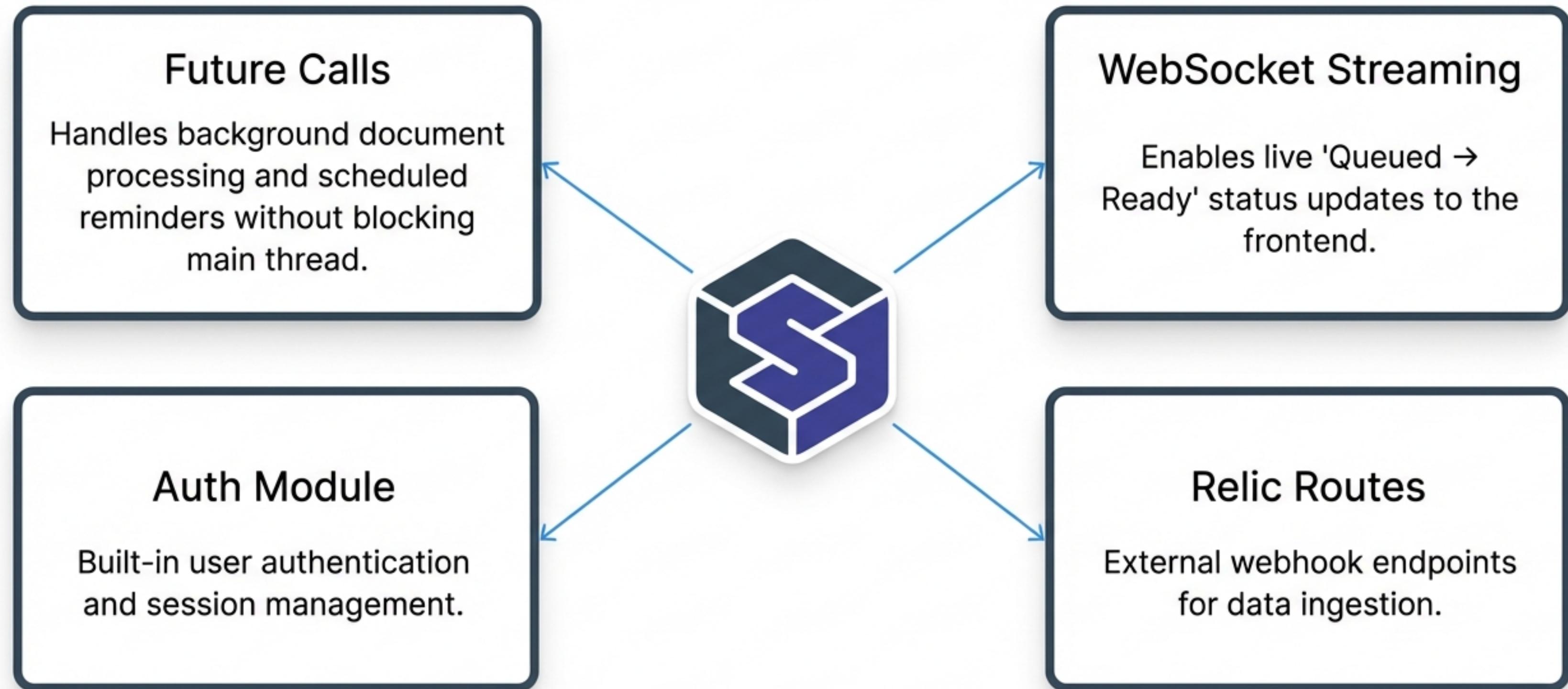
Scenario: Processing a Q4 Invoice (\$5,400 due Jan 30).

System Architecture



Storage: Local or S3-compatible (MinIO)

Leveraging Serverpod Capabilities



Database Strategy: PostgreSQL & pgvector



Unified Storage

Relational business data and vector embeddings live in the same database.
No need for Pinecone or Weaviate.

Configuration (pgvector)

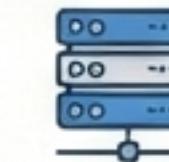
- Model: OpenAI text-embedding-3-small
- Chunk Size: 500 tokens
- Overlap: 100 tokens
- Dimensions: 1536

Comprehensive Tech Stack



Frontend

- Flutter 3.x
- Riverpod (State)
- flutter_animate



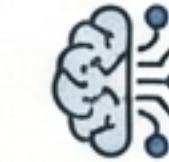
Backend

- Serverpod 3
- Dart Language



Database

- PostgreSQL 16
- pgvector extension

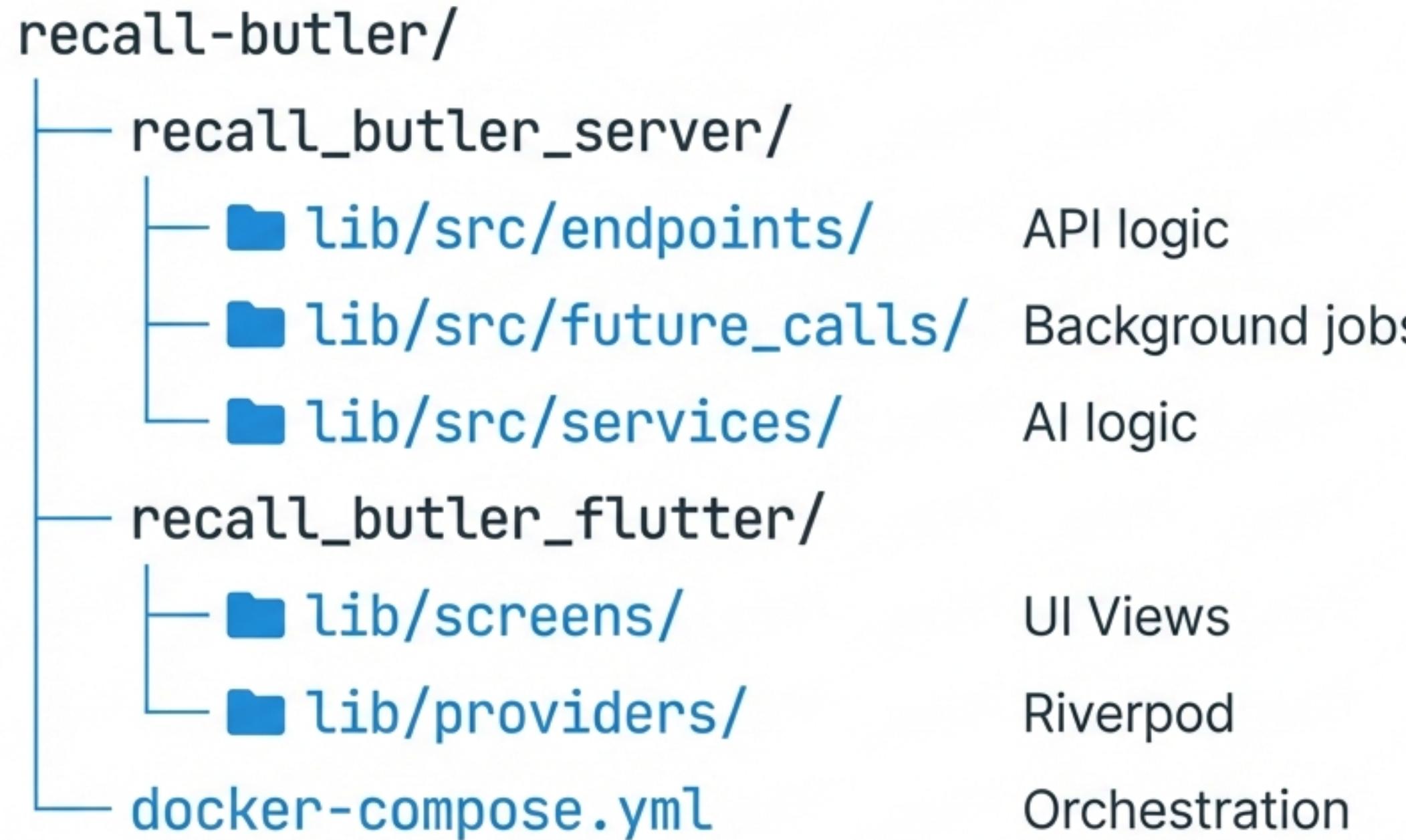


AI & DevOps

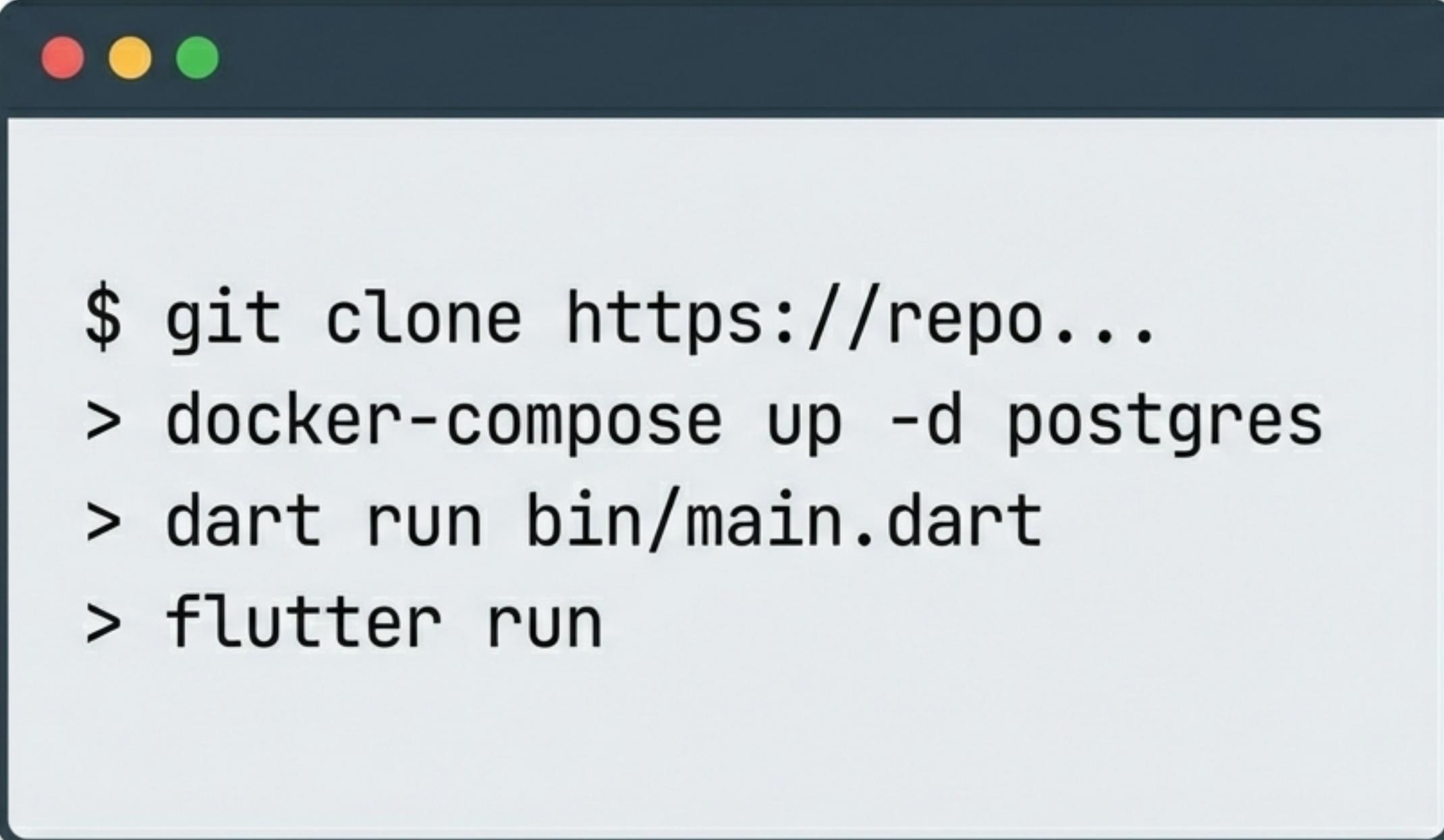


- OpenAI GPT-4o-mini
- Docker
- Serverpod CLI

Project Structure & Monorepo



Quick Start: From Zero to Butler

A terminal window icon with three colored circles (red, yellow, green) in the top-left corner, representing a Mac OS X window.

```
$ git clone https://repo...
> docker-compose up -d postgres
> dart run bin/main.dart
> flutter run
```

Prerequisites

- Flutter SDK
- Dart SDK
- Docker
- Serverpod CLI

Deployment & Configuration



Backend Strategy

Compile to EXE. Deploy to Fly.io or Render.

Requires: DATABASE_URL, OPENAI_API_KEY



Frontend Strategy

Build for web (flutter build web).
Deploy to Vercel/Firebase.

Environment Variables

OPENAI_API_KEY (Required)

POSTGRES_PASSWORD (Required)

S3_ENDPOINT (Optional)

Build Your Own Second Brain



Star on GitHub



Fork Repository

MIT License (Open Source)

Powered by Serverpod, OpenAI, and pgvector.
Built with ❤️ for Serverpod Hackathon 2026