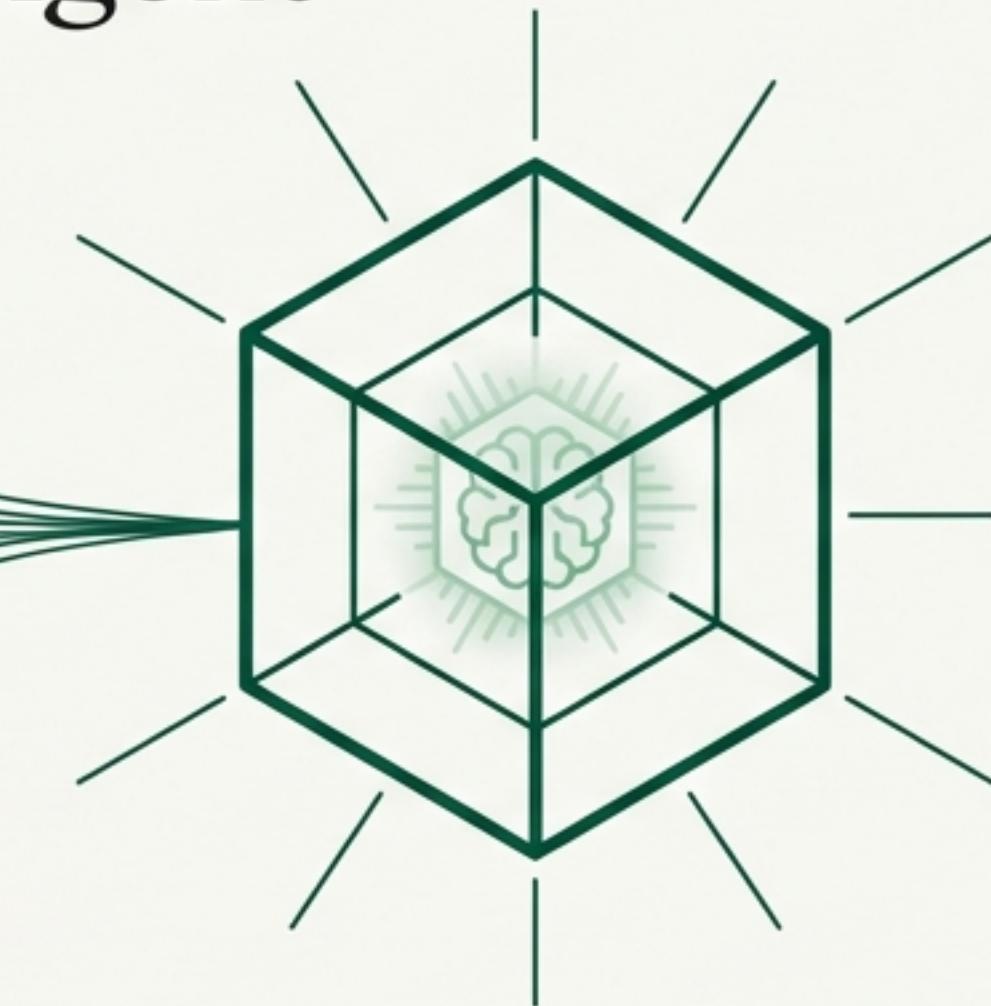
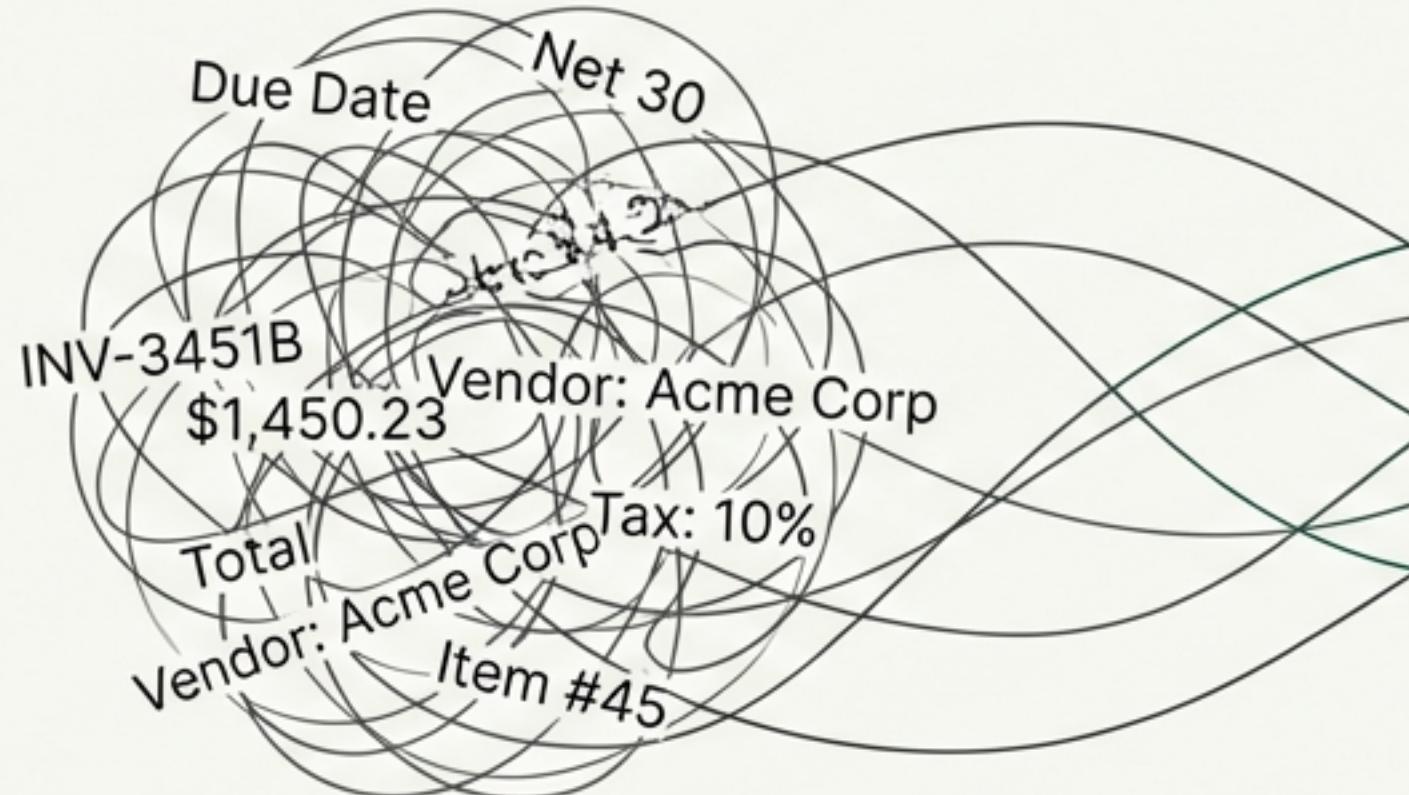


# Beyond Extraction: The Blueprint for an Autonomous Finance Agent



The challenge of invoice processing isn't just about managing diverse formats; it's about overcoming the brittle, complex, and expensive technical stacks built to handle them.

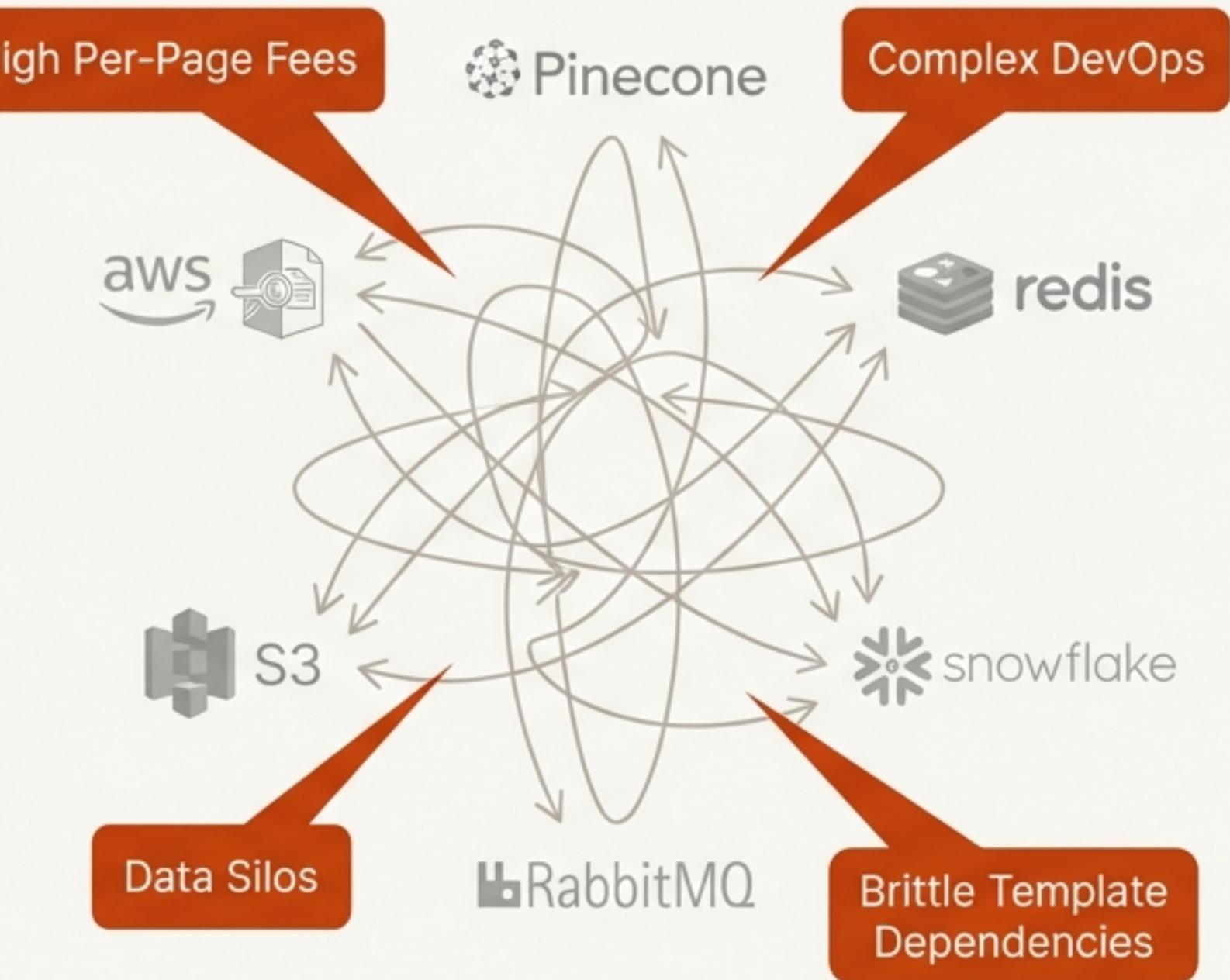
This presentation outlines a new architectural philosophy—one that creates a true AI agent capable of reading, reasoning, and reconciling financial documents with unprecedented efficiency and intelligence.

# The Status Quo: A Tax on Complexity

**Format Chaos:** Vendors send invoices in emails, portals, scans, and spreadsheets. Traditional OCR breaks whenever a layout changes.

**Manual Fatigue:** AP clerks spend 70% of their time on data entry, not high-value auditing.

**High Cost of Incumbents:** Solutions like AWS Textract or ABBYY force a choice between high per-page fees, expensive licensing, or linearly scaling costs with volume.

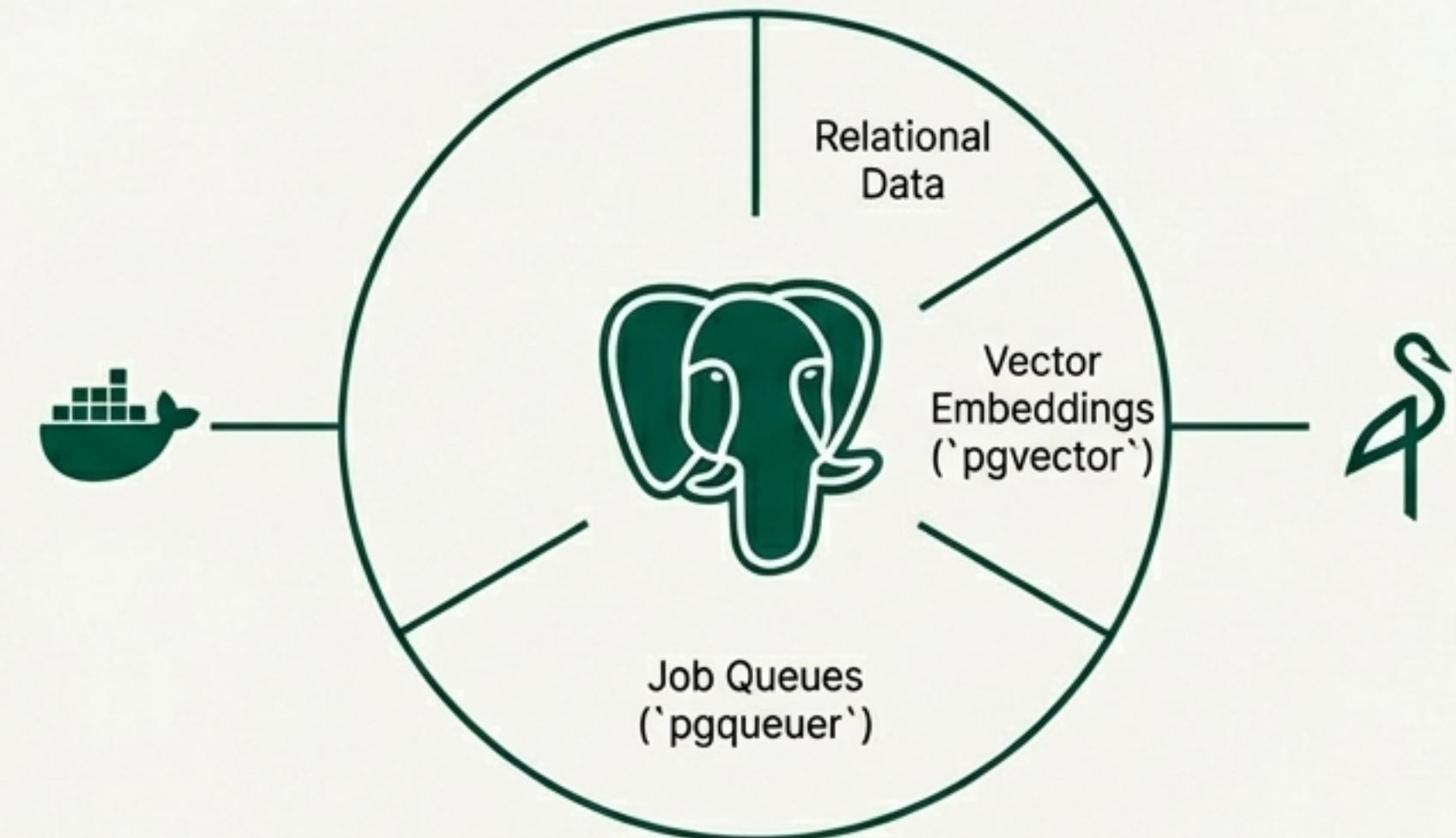


# Our Thesis: The ‘Complexity Collapse’ Model

**Core Goal:** Minimize DevOps overhead and monthly cloud spend by unifying the data and workflow layers.

**The Stack:** A single PostgreSQL instance manages relational data, vector embeddings, and asynchronous task queuing. MinIO provides S3-compatible object storage. The entire system is containerized with Docker.

**The Benefit:** This model eliminates the need for managing separate Redis, Pinecone, or Snowflake instances, reducing MVP infrastructure costs by an estimated 60%.



# The Data Journey Pt. I: The Universal Ingestion Funnel



The system acts as a universal entry point, accepting Excel, PDF, images, and emails without manual pre-sorting.

**Intelligent Routing:** A key USP. The system automatically directs different file types to the most appropriate processing agent.

This is the first step of the 'Sensory Layer,' understanding *\*what\** it has received before it tries to *\*read\** it.

# The Complete Open-Source Tech Stack Ecosystem

## Compute

Docker on a Cloud Instance  
(e.g., AWS EC2,  
DigitalOcean)

## Storage

MinIO (S3-Compatible  
Object Storage)

## Database & Orchestration

PostgreSQL (with `pgvector`  
for embeddings and  
`pgqueuer` for jobs)

## AI - Extraction & Reasoning

Docling / PaddleOCR,  
LlamaIndex, LangGraph,  
Pydantic Schemas

## AI - Interaction

Vanna.ai

## User Interface

Streamlit (for Human-in-the-  
Loop)

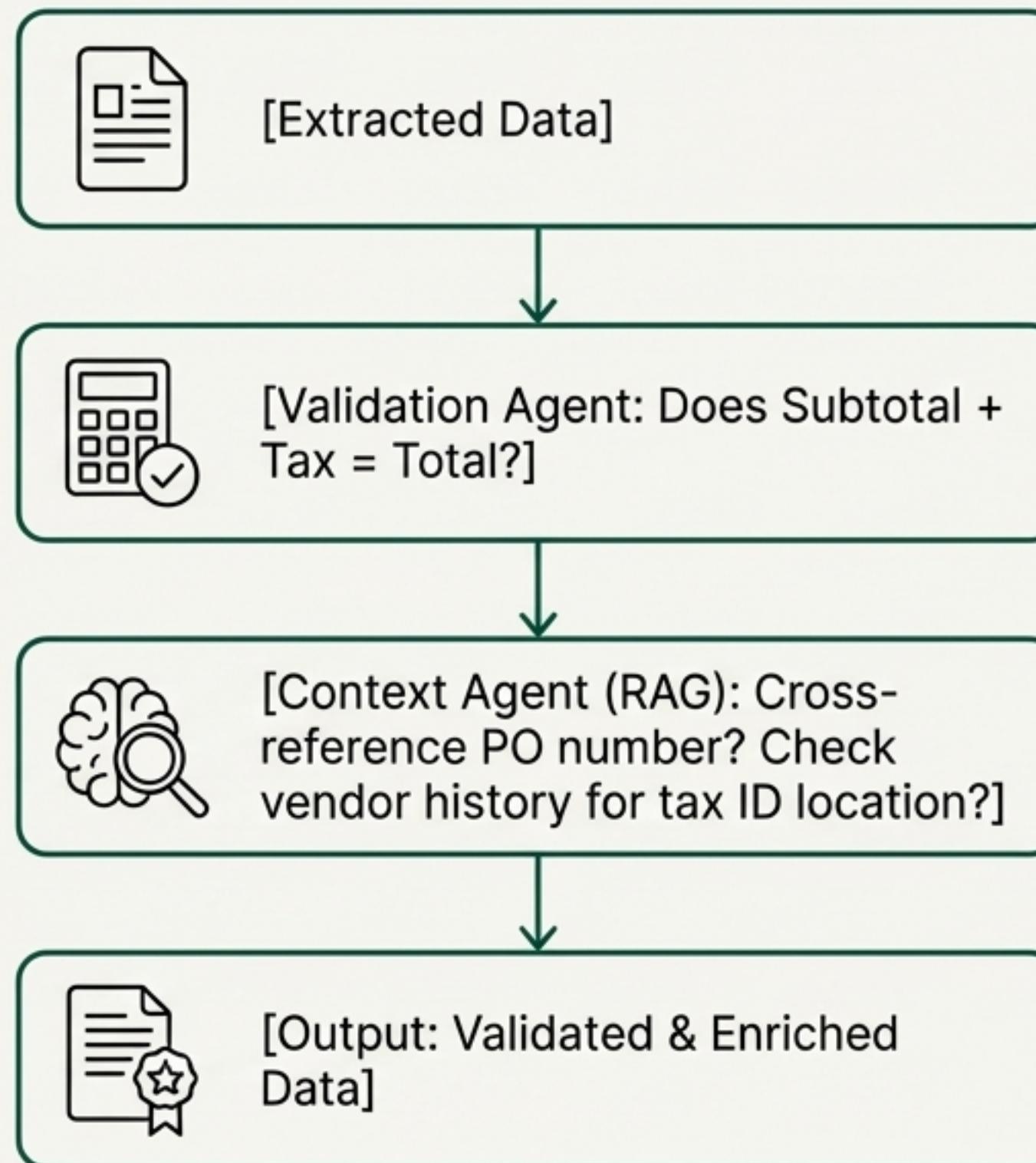
A cohesive, **cost-effective** stack chosen for low operational overhead and high intelligence. Each component is open-source, avoiding vendor lock-in and the 'snowflake tax' of proprietary cloud APIs.

# Step 1: Zero-Template Extraction with Vision Models



- Instead of relying on brittle, layout-dependent templates, our system uses advanced vision models (DocLing/PaddleOCR) to read documents like a human.
- This “Zero-Template” approach means the system is resilient to format changes and can process invoices from new vendors without any manual setup.

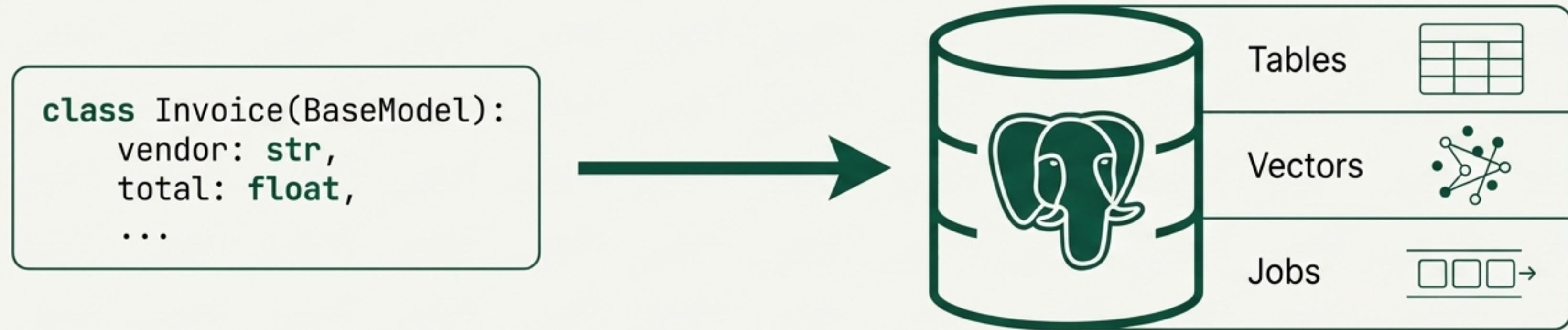
# Step 2: From Extraction to Agentic Reasoning



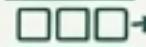
The system doesn't just extract data; it reasons about it. This is the "Agentic" difference, powered by frameworks like LlamaIndex and LangGraph.

- **Self-Correction:** If the math on an invoice doesn't add up, the agent can flag it or re-attempt extraction with a different strategy before alerting a human.
- **Retrieval-Augmented Generation (RAG):** The agent uses historical data to understand context, like learning that 'This vendor usually puts the tax ID in the footer.'

# Step 3: Structuring Intelligence in the Unified Backend



All validated and enriched data is cast into strict Pydantic schemas, ensuring data quality and consistency. This structured data is then persisted in our unified PostgreSQL backend, which handles:

- 
- Relational Data:** The core invoice fields and metadata.
-  **Vector Embeddings:** For RAG and semantic search via `pgvector`.
-  **Job State:** The processing queue itself via `pgqueuer`.

# The Interaction Layer: Human-in-the-Loop & Conversational Query

Streamlit Dashboard

Vendor	Date	Total	Status
ACME Corp	10/25/2023	\$1,200.00	Approved
Globex Inc	10/26/2023	\$500.00	Flagged
INV-1234	10/26/2023	\$500.00	Approved

**Invoice Details:** Globex Inc

Vendor: **Validation Failed:** Line items do not sum to total.

Total ~~\$500.00~~ **\$550.00**

**Confirm**

Conversational Query (Vanna.ai)

"Show me invoices that look like duplicates."

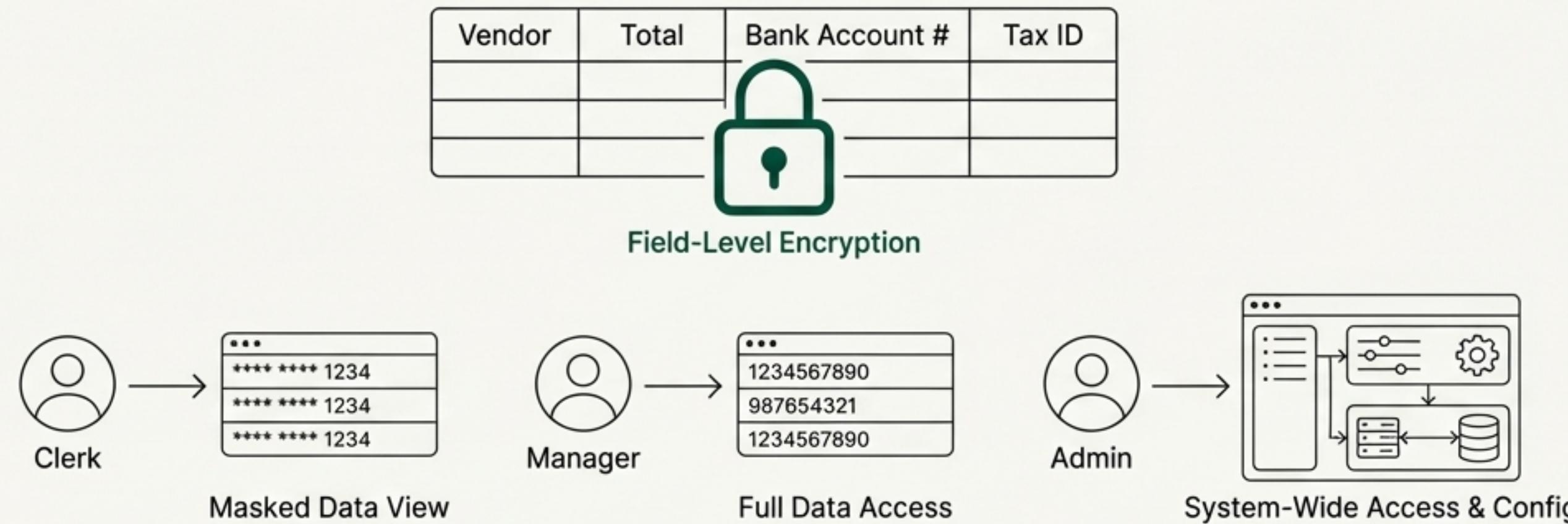


"Based on similar amounts and vendors, here are potential duplicates: INV-1234 (\$500) and INV-5678 (\$500) from ACME Corp"

**Human-in-the-Loop:** A Streamlit dashboard allows for efficient review and correction of exceptions, ensuring 100% accuracy while maximizing automation.

**Conversational Intelligence:** Non-technical finance managers can query the data using natural language, democratizing access to insights.

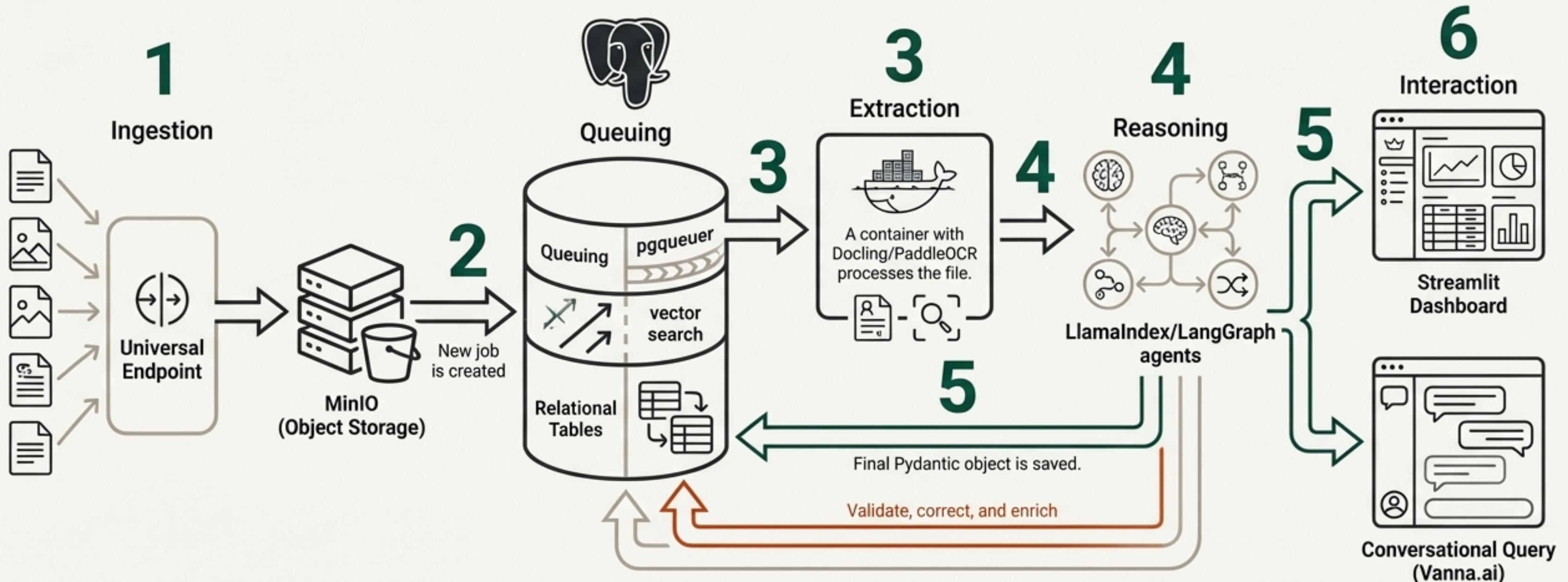
# Ensuring Security & Data Privacy by Design



Security is not an afterthought; it is built into the core architecture.

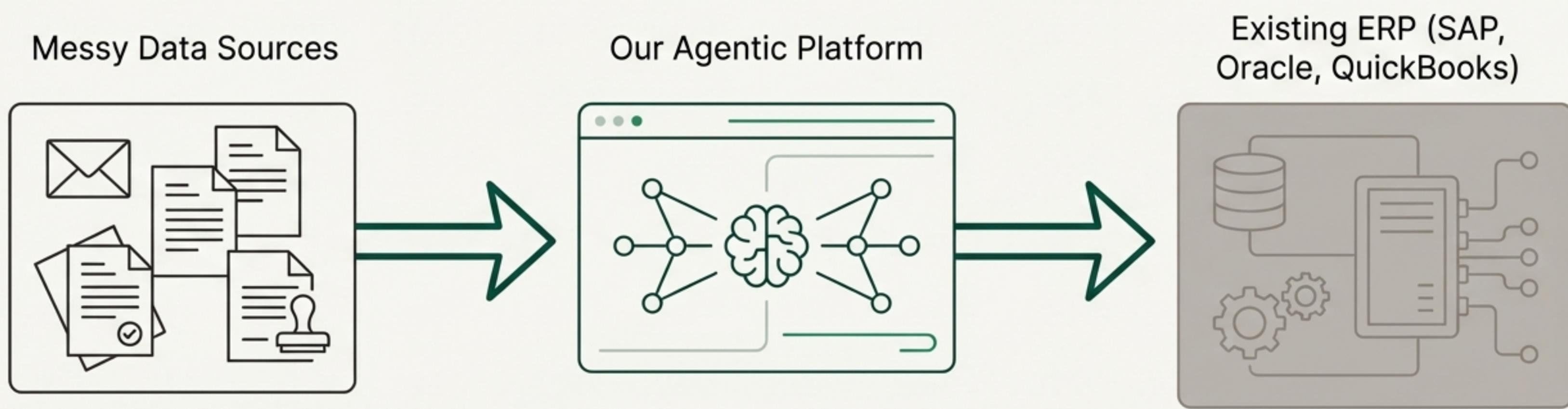
- **Field-Level Encryption:** Sensitive PII (Personally Identifiable Information) is encrypted at the field level using an envelope encryption strategy.
- **Role-Based Access Control (RBAC):** Guarantees that users can only view and interact with the data they are authorized to see.
- **Authentication:** Secure access is enforced using standard mechanisms like OAuth2 and JWT.

# The Full Architectural Blueprint: End-to-End Workflow



This architecture integrates the Sensory, Brain, and Interaction layers into a cohesive, efficient pipeline, orchestrated around a single, powerful data backend.

# Go-to-Market Strategy: The Intelligent Inbox for the ERP



We don't try to replace the entire ERP. Our strategy is to be the "Trojan Horse" that perfects the data first.

**The Pitch:** "We are the intelligent inbox for your ERP. We are a pre-processor that cleans, validates, and structures messy invoice data *before* it hits your rigid accounting system." This approach reduces customer friction and accelerates time-to-value.

# Measuring Success: The Key Performance Indicators

>80%



Straight-Through  
Processing (STP) Rate

<5 mins



Time-to-Value for  
New Users

~60%



Lower MVP  
Infrastructure Cost

Our success is defined by efficiency, speed, and cost-effectiveness.

- **STP Rate:** The percentage of invoices processed with zero human intervention.
- **Time-to-Value:** How quickly a new user can upload a batch of invoices and receive structured data.
- **Cost Efficiency:** Directly attributable to the 'Complexity Collapse' architecture.

# The Phased MVP Development Roadmap

## Month 1: The Sensory Layer

- ➡ Build the universal ingestion pipeline.
- 📄 Integrate Docing/PaddleOCR.
- ในฐาน Store raw extraction results in Postgres.

## Month 2: The Brain Layer

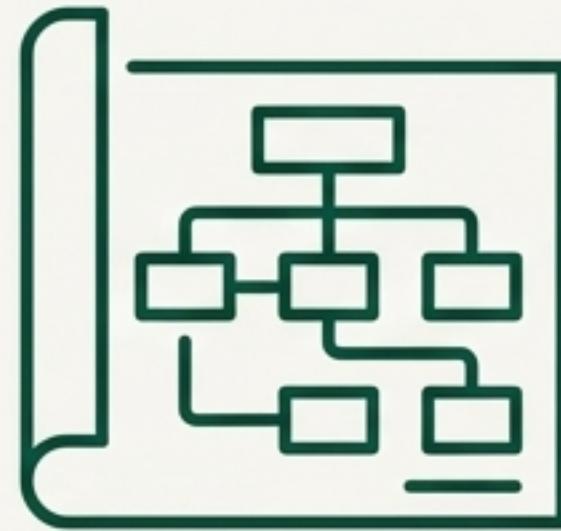
- 🧠 Implement LlamaIndex extraction & reasoning logic.
- 📦 Define Pydantic schemas.
- ✅ Build the first “Validation Agent” to check math.

## Month 3: The Interaction Layer

- 👤 Deploy Streamlit dashboard for Human-in-the-Loop review.
- 👤 Alpha release to 3-5 friendly pilot users.

# Immediate Next Steps to Build the Foundation

The path forward is clear. We begin by establishing the core components of our system.



SCHEMA BLUEPRINT

1. **\*\*Freeze the Schema\*\***: Finalize the Pydantic models for the `Invoice` object to define our ground truth.



INFRASTRUCTURE STACK

2. **\*\*Deploy Infrastructure\*\***: Launch the `docker-compose` environment with Postgres and MinIO.



GOLD STANDARD DATA

3. **\*\*Curate Test Data\*\***: Gather 50-100 diverse, anonymized invoices to create our 'Gold Standard' test set for measuring accuracy.