Project Gemini

The Blueprint for an Intelligent Document Platform

Our Mission: Transform Any Document into Queryable Intelligence

We are building a platform to streamline the processing of diverse document formats. The core capability is to ingest any document—from text and images to audio and video—and convert it into structured, actionable insights. This is achieved through a combination of:

Advanced OCR & Multimedia Extraction

To accurately parse text, metadata, and content from any source.

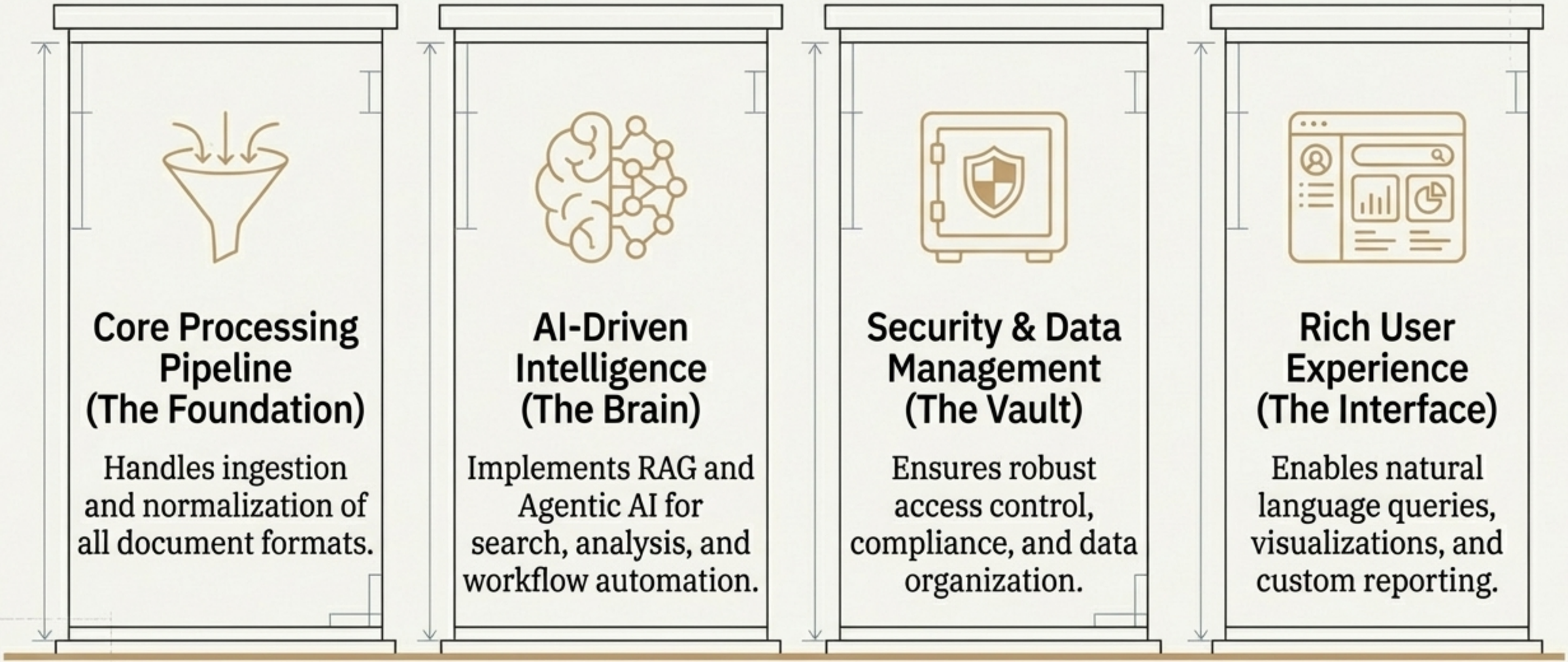
Retrieval-Augmented Generation (RAG)

For precise, on-demand information extraction, search, and summarization without model retraining.

Agentic AI

To automate complex workflows like content tagging, version control, and system integrations.

The Four Pillars of the Platform Architecture



Core Processing Pipeline (The Foundation)

Handles ingestion and normalization of all document formats.

AI-Driven Intelligence (The Brain)

Implements RAG and Agentic AI for search, analysis, and workflow automation.

Security & Data Management (The Vault)

Ensures robust access control, compliance, and data organization.

Rich User Experience (The Interface)

Enables natural language queries, visualizations, and custom reporting.

The Core Engine: How a Document Becomes an Insight

Ingestion & Unified Parsing

Input: Diverse Formats (PDFs, DOCX, Images, Audio, Video)

Process: Robust OCR, NLP, and multimedia extraction technologies.

Indexing & Augmentation

Process: Data is structured, vectorized, and stored for retrieval.

RAG & Agentic AI Layer

Process: Enables semantic search, validation, summarization, and automated workflows.

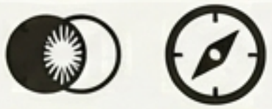
Actionable Intelligence

Output: Natural language answers, customized reports, and dynamic visualizations.

From Vision to Viable Product: The POC Blueprint

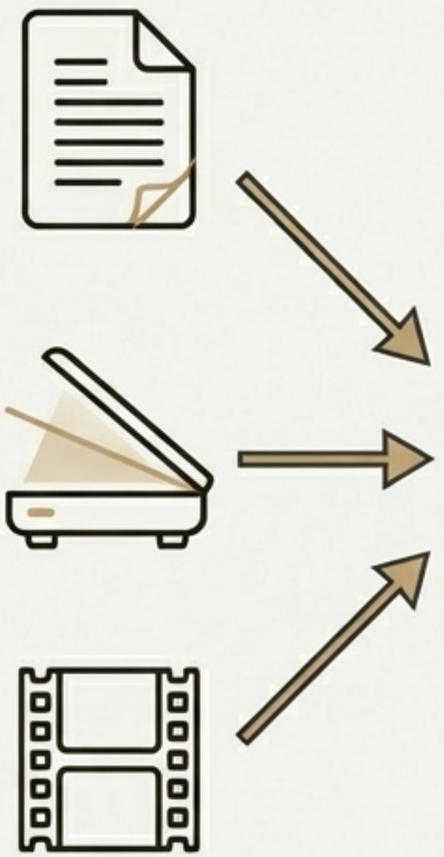
The following plan outlines a pragmatic path to a powerful Proof of Concept (POC). Our objective is to validate the core architecture using a focused, cost-effective, and scalable technology stack, prioritizing open-source and modular components to ensure flexibility and control.

The Proposed POC Technology Stack

Frontend & Visualization		React, D3.js
Security & Auth		Supabase Auth / Keycloak
AI Core (RAG & Agents)		LangChain / LlamaIndex
Vector Database		Chroma / pgvector
Data Storage (Structured)		DuckDB / PostgreSQL
Data Storage (Unstructured)		AWS S3 / Cloud Object Storage
Ingestion & Parsing		Unstructured, Tesseract, Whisper

Component 1: Handling Any Document with an Open-Source Ingestion Pipeline

We will leverage a suite of powerful, cost-effective open-source libraries to build a universal parser for text, image, and multimedia content. This approach avoids vendor lock-in and provides maximum flexibility.



For Document Parsing (PDF, DOCX, PPTX): ‘Unstructured’, ‘LangChain Document Loaders’

For Optical Character Recognition (OCR): ‘Tesseract’, **For Multimedia Transcription (Audio/Video):** ‘Whisper’, ‘ffmpeg’

Key Benefit: Establishes a robust foundation capable of processing a wide array of business and multimedia documents from day one.

Component 2: Building the Intelligence Layer with Modern AI Frameworks

RAG & Agent Frameworks

Analysis: Compared `LangChain`, `LlamaIndex`, and `LangGraph`.

POC Recommendation: `LangChain` for its mature ecosystem and comprehensive tooling for building complex chains and agentic workflows.

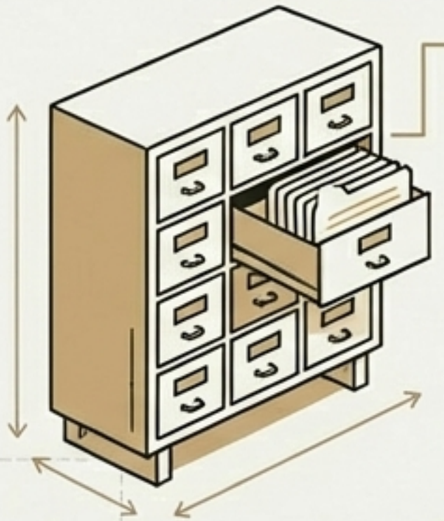
Vector Database

Analysis: Evaluated `Chroma`, `pgvector`, and `Weaviate`.

POC Recommendation: `Chroma` for its lightweight, in-memory architecture, making it ideal for rapid prototyping. `pgvector` is a strong alternative if tight integration with PostgreSQL is required.

Component 3: Designing a Smart and Scalable Storage Strategy

Our data strategy borrows concepts from enterprise warehouses but starts with lean, cost-effective solutions tailored for a POC. We will separate storage for structured metadata and unstructured document artifacts.



For Structured Data & Metadata

Strategy: Instead of a high-cost service like Snowflake, we will use a lightweight, efficient alternative.

Recommendation: DuckDB for fast, embedded analytics or PostgreSQL for a more robust, general-purpose relational database.

For Unstructured Artifacts (Original & Processed Files)

Strategy: Cloud object storage provides near-infinite scalability and cost-efficiency.

Recommendation: AWS S3, Google Cloud Storage, or Azure Blob Storage.

Component 4: Ensuring Enterprise-Grade Security from Day One

To protect sensitive documents and ensure compliance, we will implement a robust security framework with out-of-the-box features for authentication and access control.

Robust Authentication & Encryption

Compliance Features
(e.g., GDPR-ready, Audit Logs)

Role-Based Access Controls (RBAC)

Recommended Solutions:

- `Supabase Auth`: Integrates well with a PostgreSQL backend and provides a comprehensive suite of security features.
- `Keycloak`: A powerful open-source option for self-hosting and full control over identity and access management.
- `Auth0`: A mature, managed service for rapid implementation.

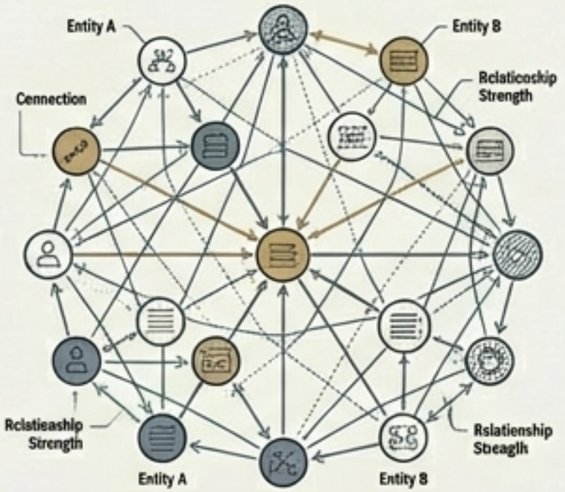
Component 5: Bringing Insights to Life with a Rich User Experience

The user interface will enable natural interaction with the data. Users can query documents in plain language, generate custom reports, and visualize connections and trends within the content.

Key UI Features for POC:

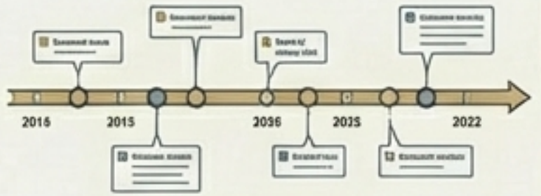
- Natural Language Query Interface: “Find contracts with clause X.”
- Custom Report & Summary Generation: Exportable in preferred formats.
- Dynamic Data Visualizations: To reveal patterns and key metrics.

Visualization Examples



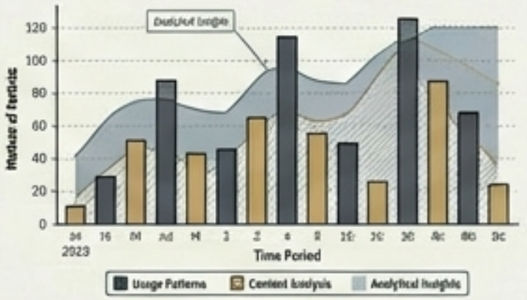
Network Graphs

To show relationships between entities.



Timelines

To track events or document versions.



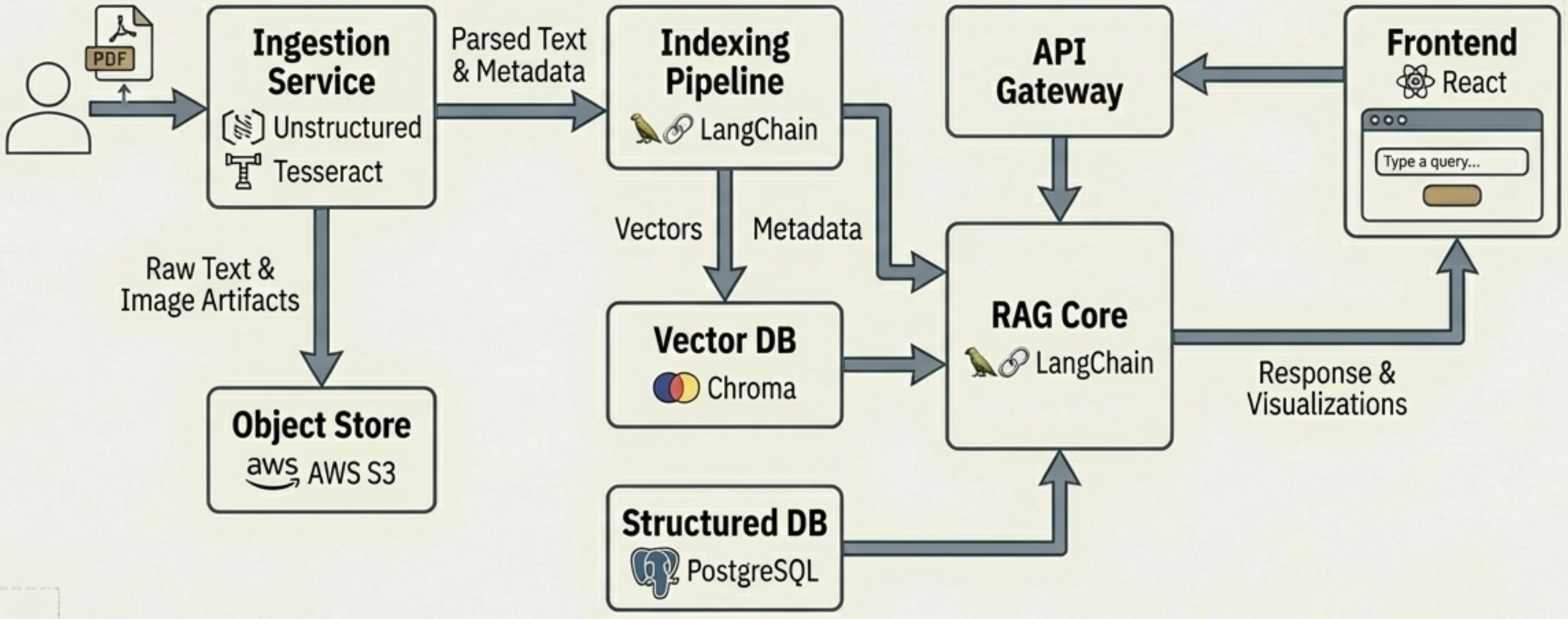
Timelines

To track events or document versions.

Dynamic Charts

For usage patterns and content analysis.

The Synthesized POC Architecture: A Complete Blueprint



A Phased Roadmap for POC Implementation

Phase 1: The Foundation (Weeks 1-2)

- **Goal:** Establish the core data processing capability.
- **Tasks:** Set up the document ingestion and parsing pipeline. Implement basic RAG chain with vector storage.
- **Key Tech:** Unstructured, Tesseract, Whisper, LangChain, Chroma.

Phase 2: Interaction & Security (Weeks 3-4)

- **Goal:** Build the user-facing elements.
- **Tasks:** Implement user authentication and RBAC. Develop a basic UI for document upload and natural language querying.
- **Key Tech:** Supabase Auth, Frontend Framework.

Phase 3: Automation & Insight (Weeks 5-6)

- **Goal:** Introduce advanced features.
- **Tasks:** Develop initial agentic workflows for automated tagging. Build the first set of dynamic data visualizations (e.g., a network graph).
- **Key Tech:** LangChain Agents, Visualization Libraries.

Key Decisions and Immediate Next Steps

Key Decisions Solidified

- **Prioritize Open-Source:** Leverage robust, community-backed libraries for core functions to maximize flexibility and control costs.
- **Cloud-Native Storage:** Utilize scalable object storage for unstructured data from the outset.
- **Modular, Phased Architecture:** Build components independently to de-risk development and validate the core pipeline first.
- **Pragmatic Tech Choices:** Select lightweight, POC-appropriate tools (e.g., Chroma, DuckDB) over heavy enterprise solutions.

Immediate Next Steps

- Finalize specific library versions for the tech stack.
- **Set** up the development environment and cloud infrastructure (S3, PostgreSQL).
- **Begin Sprint 1:** Focus on building the Phase 1 ingestion and parsing pipeline. Ingestion compons and practice, taggger 2, Phase 1 ingestion and parsing pipeline.
- **Define** initial test documents and success metrics for the core RAG functionality.

Q & A

Project Gemini

Thank you.