LexAtlas

Project Abstract

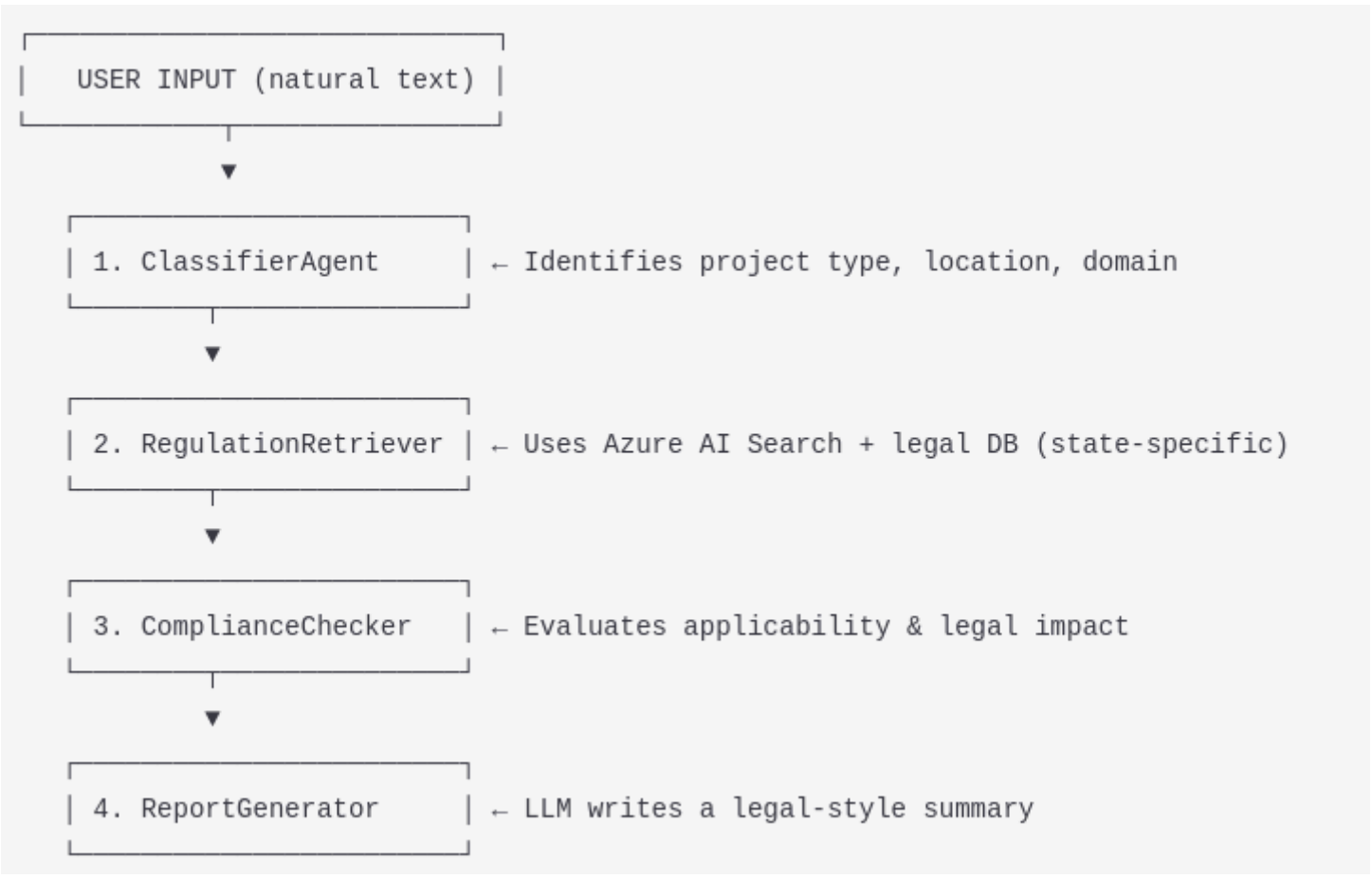
Enable an AI agent to analyze a user-submitted project (in natural language) and return:

- Relevant state-level regulations (US)
- Legal risk or compliance assessment
- Initial legal-technical summary report

Required hackathon components

1. **LLM** – Azure OpenAI (GPT-4o or model O) [done]
   - Processes user queries and legal text
   - Generates summaries and legal reports
2. **Orchestrator** – Azure AI Agent SDK + Semantic Kernel [done]
   - Manages task planning across sub-agents
   - Modular and scalable multi-agent logic
3. **Tooling System**
   - Azure AI Search – Semantic search over indexed legal corpora (RAG) [done]
   - Azure SQL or Cosmos DB – Structured data for law-by-state mapping
   - APIs – Legal data retrieval from Justia, LexisNexis, or mock services
4. **Memory Store** – Azure Agent Thread Storage
   - Tracks project inputs and agent interactions [done]
   - Enables follow-up queries or long-term memory

Intelligent Agent Flow



Trustworthy Agent Design

- **Context-Aware**: Adapts to different states and industries
- **Stateful**: Maintains history of past interactions/projects
- **Human-in-the-loop**: Can flag results for human review
- **Evaluated**: Logs steps and output for traceability
- **Repeatable**: Uses templates for system messages and logic

Example Demo Scenario

User Input:

> “We’re planning a carbon capture project in Wyoming using DAC technology.”

Agent Output:

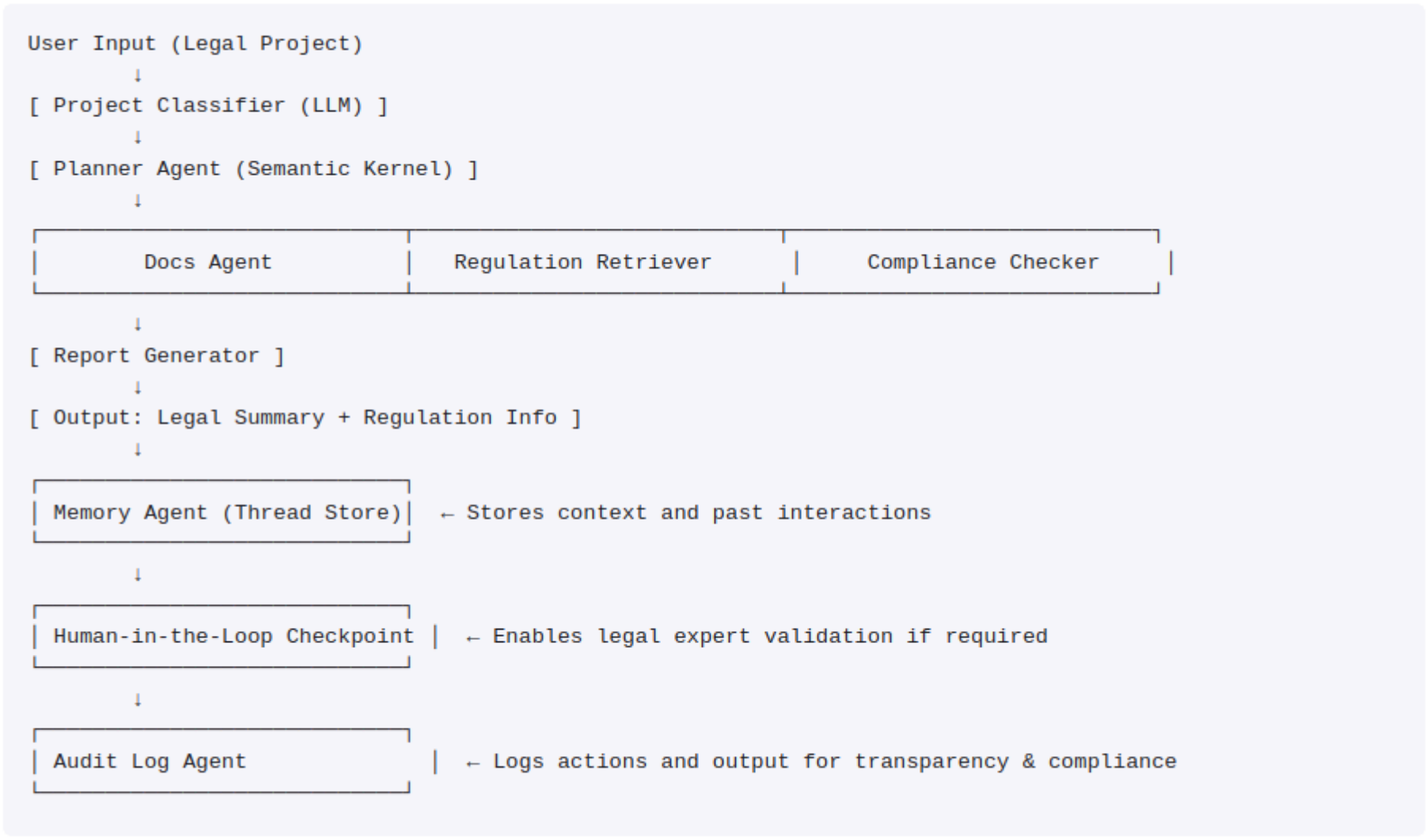
> Based on the input, this is a Wyoming-based industrial emissions project involving Direct Air Capture.
> Under the Wyoming Environmental Quality Act, carbon capture requires authorization from the Department of Environmental Quality (DEQ), Section 35-11.
> This project may qualify for incentives under the Wyoming Carbon Capture Program if geologic storage conditions are met.
> We recommend contacting the DEQ’s Division of Air Quality for pre-permitting consultation.

Flow Overview

1. User submits a project description.
2. **Classifier Agent** (powered by GPT-4 via Azure OpenAI) analyzes the type and location of the project.
3. **Regulation Retriever Agent** fetches relevant state laws using:
    - Azure AI Search for semantic document retrieval
    - Azure SQL or CosmosDB for structured regulation mapping
4. **Compliance Checker Agent** assesses if the project meets legal requirements.
5. **Report Generator Agent** produces a plain-English legal summary.
6. Memory and orchestration is managed by:
    - Azure AI Agent SDK (Semantic Kernel)
    - Azure Agent Service (Thread Storage)

Architecture diagram

```
User Input (Legal Project Description)
    ↓
[ Project Classifier Agent (LLM - GPT-4o) ]
    ↓
[ Planner / Orchestrator Agent (Semantic Kernel) ]
    ↓
┌──────────────┬──────────────────┬──────────────────┐
| Docs Agent   | Regulation Retriever | Compliance Checker   | ← Specialized sub-agents
└──────────────┴──────────────────┴──────────────────┘
    ↓
[ Report Generator Agent ]
    ↓
[ Final Output: Legal Summary + Cited Regulations ]
    ↓
──────────────────────────────────────────────────────
[ Memory Agent (PostgreSQL / Thread Store) ]   ← Multi-turn context + persistence
    ↓
[ Prompt & Agent Configuration Manager ]       ← Enables LLM + prompt variation testing
    ↓
[ Human-in-the-Loop Review (Optional) ]        ← Legal expert validation checkpoint
    ↓
[ Audit Log Agent ]                            ← Logs inputs, outputs, actions
──────────────────────────────────────────────────────
Optional: LangChain Pipeline (used for comparative benchmarking)
```



y

Incremental Development Plan for LexAtlas Prototype

To mitigate risks and ensure steady progress throughout the 3-week hackathon, we propose an incremental development approach with three structured phases:

Phase 1: Technical Feasibility (Minimal End-to-End Flow) [done]

Objective: Validate the feasibility of building a functional end-to-end pipeline.
Scope:

- Develop a minimal web application where users can input legal queries or upload documents.
- Connect the frontend with Azure OpenAI (GPT-4 o) via a backend service.
- Return a basic legal summary or recommendation based on the input.
- Focus on achieving a stable, functional roundtrip interaction.

Goal: Demonstrate core system integration and establish a foundation for further capabilities.

Phase 2: Core Functional Expansion [done]

Objective: Expand the system with essential features that represent the core functionality of LexAtlas.
Scope:

- Project Classifier Agent: Identify project type and jurisdiction (e.g., state-level).
- Regulation Retrieval: Use Azure AI Search and a structured database (Azure SQL or mock data) to fetch relevant legal information.
- Report Generator: Return user-friendly summaries with initial compliance insights.
- Basic Orchestration: Use Azure Semantic Kernel to manage multi-agent coordination.

Goal: Enable LexAtlas to handle real-world legal project queries with contextual understanding.

Phase 3: Advanced Capabilities

Objective: Integrate enhanced features for more intelligent, dynamic user experiences.
Scope:

- Memory & Context Management: Implement Azure Agent Thread Storage to track past queries and projects. [done]
- Compliance Checker Agent: Assess legal risk or gaps based on retrieved regulations.
- Human-in-the-Loop: Add the ability to flag results for human legal review or recommendation routing. [done]
- Multi-turn Interaction: Enable follow-up questions and deeper dialogue using stored context. [done]

Goal: Deliver a more powerful, intelligent prototype demonstrating depth and extensibility.

Resources

Infrastructure Table

Component	Azure Service	Purpose
LLM Backend	Azure OpenAI (GPT-4 Turbo)	Natural language understanding, classification, compliance checking
Semantic Retrieval Engine	Azure AI Search	Retrieve relevant legal content using semantic search (RAG)
Legal Data Store	Azure SQL Database (or Cosmos DB)	Stores structured state-level regulation mappings
Document Storage	Azure Blob Storage	Store user-submitted project files and legal corpora
Context/Memory Store	Azure Agent Thread Storage	Maintains conversation history and project context
Agent Orchestration Layer	Azure AI Agent SDK + Semantic Kernel	Coordinates multi-agent logic and task planning
Backend Logic Hosting	Azure App Service (or Azure Functions)	Hosts APIs and orchestrates interaction with services
Web Application Frontend	Azure Static Web Apps (or App Service)	Provides user interface for project input and results display
Monitoring & Logging	Azure Monitor + App Insights	Logs system behavior and errors for debugging and audit
Security & Secrets Management	Azure Key Vault	Secure storage of API keys and credentials
Authentication (optional)	Azure AD B2C (or MS Entra ID)	User authentication and access control
CI/CD & DevOps	GitHub Actions / Azure DevOps	Deployment automation, source control, and build pipelines

Team Roles

Member Role	Primary Skills	Responsibilities	Member Name
Team Lead / PM / Legal Advisor	Product vision, legal knowledge, coordination	Define scope, manage roadmap, guide legal focus, act as subject-matter expert	
AI/ML Engineer	Prompt engineering, GPT-4 integration, LLM chaining	Implement classification, compliance logic, and summarization agents	
Full-Stack Developer	Backend APIs, Azure integration, frontend dev	Build web UI, connect to OpenAI, Azure AI Search, and orchestrate workflows	
Data Engineer / Infra Specialist	Azure Search, SQL setup, indexing pipelines	Create legal data backend, enable semantic search, structure state law mappings	
UX/UI & Testing Specialist	UI design, usability, QA testing	Design intuitive frontend, test system flows, validate results for demo	

Timeline

Week	Focus Phase	Key Activities	Responsible Roles
Week 1	Phase 1: Technical Feasibility		
Day 1–2	Project kickoff & scoping	Define MVP, assign roles, finalize architecture	Team Lead, All
Day 3–5	Build minimal frontend/backend	User input form, connect to Azure OpenAI, return basic GPT-generated legal summary	Full-Stack Dev, AI Engineer
Day 6–7	Test E2E flow & refine UX	Verify input → output roundtrip, mock basic legal output	UX/UI, Team Lead, AI Engineer
Week 2	Phase 2: Core Functionality		
Day 8–10	Implement Project Classifier Agent	Use GPT-4 to extract project type and state from natural language	AI Engineer
Day 10–12	Set up Azure AI Search + DB	Index mock legal corpora, build regulation retriever logic	Data Engineer
Day 13–14	Generate compliance summary	Add logic for plain-English legal + compliance reporting	AI Engineer, Full-Stack Dev
Week 3	Phase 3: Advanced Capabilities		
Day 15–16	Add memory/context logic (mocked)	Simulate Agent Thread Storage; track input history (or mock it)	AI Engineer, Dev
Day 17–18	Human-in-the-loop (mock flow)	Simulate flagging for review; show placeholder for legal reviewer recommendation	Team Lead, UX/UI
Day 19	Final integration & polish	Refactor code, ensure clean UI/UX, fix bugs	All
Day 20	Internal demo + pitch prep	Prepare slides, run through demo flow, highlight agent orchestration	All
Day 21	Hackathon demo day	Deliver live or recorded demo, answer questions, showcase architecture + use case	All