System Brief A: DocuMind AI Assistant

Assigned groups: Strawberry, Mango, Kiwi, Pineapple

Use this brief only if your group was assigned to **System A**.

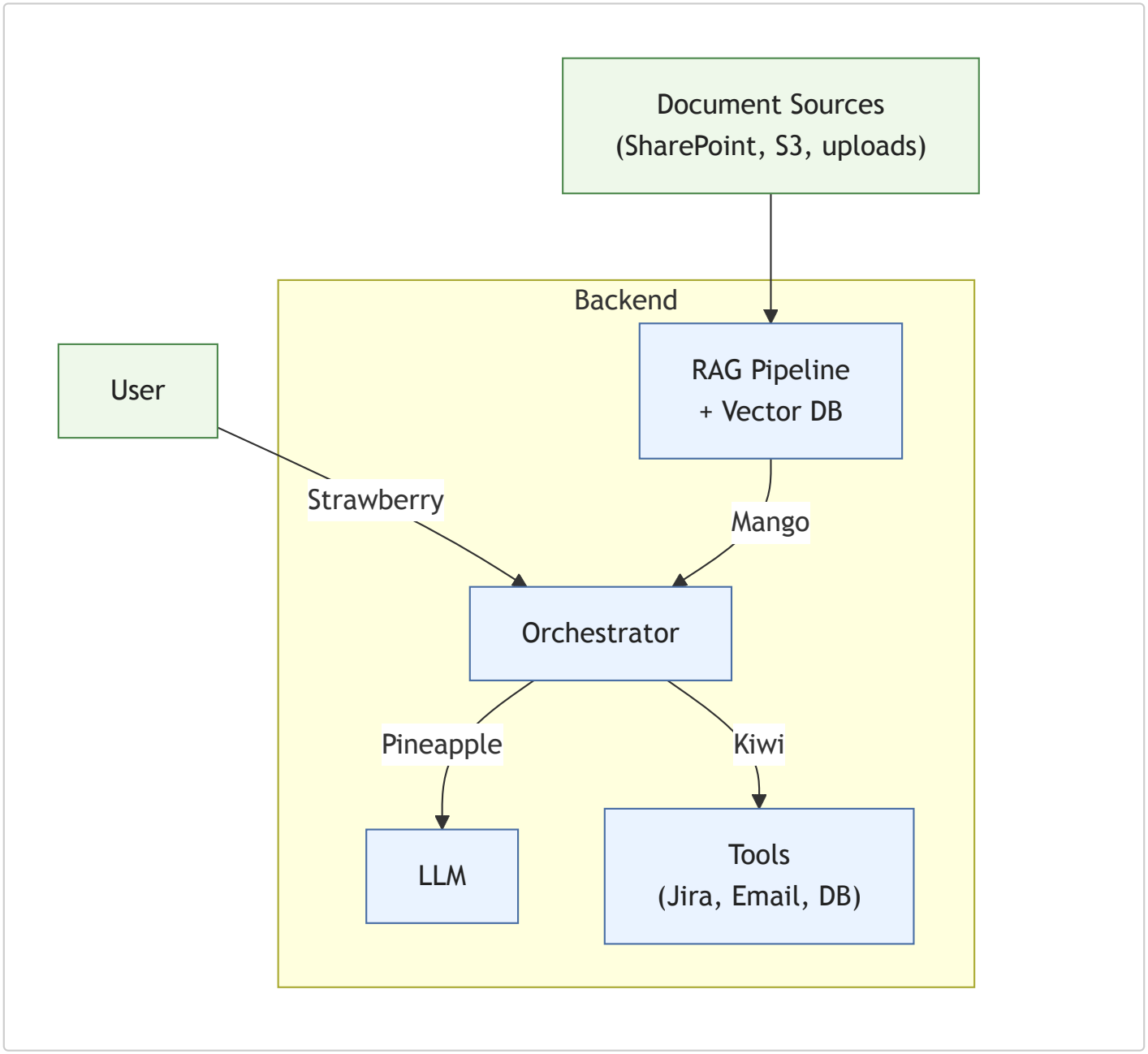
System Summary

DocuMind is an enterprise AI assistant that searches internal documents and can take actions through connected tools.

Main Capabilities

- Retrieval-augmented generation (RAG) over company documents
- Tool use for tickets, email, and database queries
- Multi-user access with different permission levels
- Conversational context across a session

Architecture



Security Note

The **orchestrator** is the main enforcement point. It decides what context to send to the model and which tools may be called.

Group Assignments

Group	Boundary	Main Theme
Strawberry	User → Orchestrator	Input trust, session trust, direct prompt attacks
Mango	RAG → Orchestrator	Retrieval trust, document permissions, injected context
Kiwi	Orchestrator → Tools	Action safety, authorization, tool abuse
Pineapple	Orchestrator → LLM	Unsafe outputs, prompt leakage, output validation

---

Boundary Hints

Strawberry

Crosses boundary:

- User prompts
- Session state
- Auth context
- Conversation history

Good places to look for threats:

- Prompt injection
- Stolen sessions
- Impersonation
- Abuse of long or repeated prompts
- Requests that try to override policy

Example threat: A user crafts input that causes the assistant to ignore policy and reveal restricted information.

Mango

Crosses boundary:

- Retrieved documents
- Snippets
- Metadata
- Search results

Good places to look for threats:

- Permission bypass in retrieval
- Sensitive data leakage through search
- Poisoned or misleading retrieved content
- Fake or tampered documents
- Documents that contain hidden instructions

Example threat: The retriever returns confidential HR data because access checks are applied too late or not at all.

Kiwi

Crosses boundary:

- Tool calls

- Tool parameters
- Tool responses
- Real-world side effects

Good places to look for threats:

- Confused deputy
- Unauthorized tool execution
- Dangerous parameter injection
- Excessive automation
- Data exfiltration through tool results

Example threat: `The assistant uses a send-email tool to exfiltrate sensitive information to an external recipient.`

Pineapple

Crosses boundary:

- Model outputs
- Recommended actions
- Structured responses
- Tool call proposals

Good places to look for threats:

- Hidden prompt leakage
- Unsafe or fabricated tool calls
- Over-trusting structured output
- Malicious content in generated text
- Outputs that bypass downstream checks

Example threat: `The model hallucinates a privileged tool call and the backend executes it without sufficient validation.`

Reminder for This System

Your job is not to redesign the whole system. Focus on:

- Your assigned boundary
- Realistic attacks at that boundary
- Risks that come from the components and trust boundaries themselves
- Mitigations that engineering and security teams could actually implement

Do not assume the risk disappears just because the team writes good code. In this exercise, the important question is what can go wrong because the system uses an LLM, RAG, tools, and access-control boundaries in the first place.