System Brief B: CarePilot Clinical Operations Assistant

Assigned groups: Watermelon, Blueberry, Coconut, Lemon

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

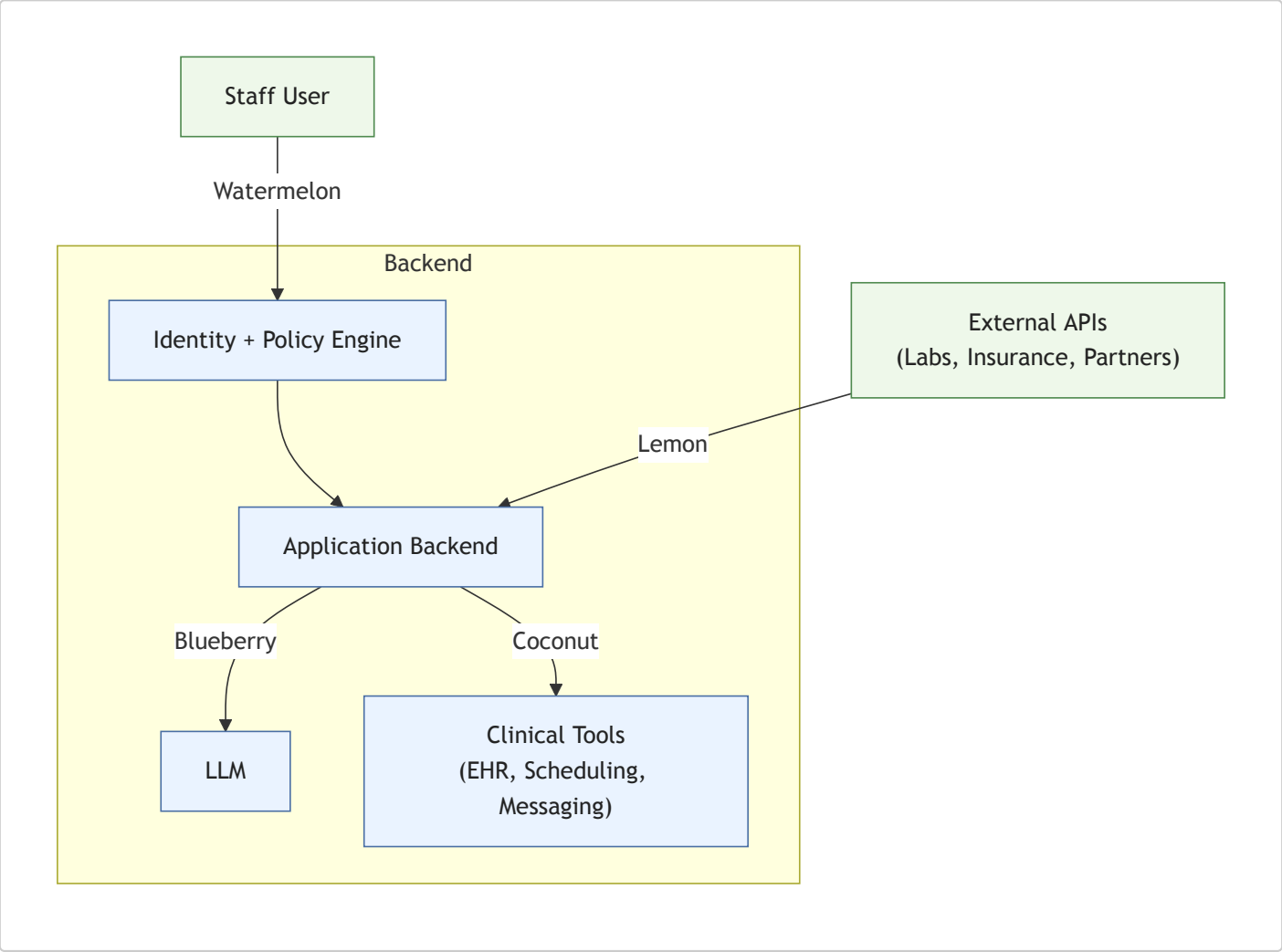
System Summary

CarePilot is a healthcare AI assistant used by staff to support clinical operations and administrative workflows.

Main Capabilities

- Summarizes policies and support documents with RAG
- Assists with patient operations workflows
- Interacts with clinical systems and messaging tools
- Relies on external healthcare and insurance services

Architecture



Security Note

The backend and policy controls sit between the user, model, tools, and external services. In this system, bad outputs can create real operational and privacy harm.

Group Assignments

Group	Boundary	Main Theme
Watermelon	Staff User → Identity/Policy Engine	Identity trust, role mix-ups, authorization
Blueberry	Backend → LLM	Unsafe model outputs, policy bypass through generation
Coconut	Backend → Clinical Tools	Action safety, record integrity, operational abuse
Lemon	External APIs → Backend	Third-party trust, data integrity, upstream compromise

---

Boundary Hints

Watermelon

Crosses boundary:

- User identity
- Session context
- Role claims
- Delegated access

Good places to look for threats:

- Shared workstation sessions
- Impersonation
- Broken role checks
- Stale or mixed session state
- Unauthorized viewing of patient or billing data

Example threat: `A staff member inherits a previous clinician's session context and gains access to patient information they should not see.`

Blueberry

Crosses boundary:

- Model prompts

- Generated summaries
- Recommendations
- Structured action suggestions

Good places to look for threats:

- Model outputs that appear authoritative
- Prompt leakage
- Unsafe generated actions
- Fabricated or misleading recommendations
- Outputs that try to bypass approval checks

Example threat: `The model generates an "emergency override" recommendation that downstream systems treat as valid.`

Coconut

Crosses boundary:

- Scheduling actions
- Record updates
- Referrals
- Outbound notifications

Good places to look for threats:

- Wrong-patient actions
- Mass automation errors
- Misuse of messaging tools
- Privilege misuse
- Weak auditability of actions

Example threat: `An automated workflow updates or messages the wrong patients because tool parameters are not sufficiently checked.`

Lemon

Crosses boundary:

- Lab responses
- Insurance responses
- Partner payloads
- Upstream service metadata

Good places to look for threats:

- Spoofed partner APIs

- Tampered results
- Dependency failure
- Excess data sharing
- Compromise of a trusted third party

Example threat: `A compromised external service returns manipulated data that the assistant summarizes and acts on as if it were trustworthy.`

Reminder for This System

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

- Threats at your boundary
- Patient privacy and operational impact
- Risks that come from the components and trust boundaries themselves
- Mitigations that engineering and security teams could realistically 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, external APIs, policy controls, and clinical tools in the first place.