# C:\Users\Admin\AppData\Local\Microsoft\Windows\INetCache\Content.Word\System Context.pngAssignment 3 – API Service Designs & Architecture (MemoriAI)

Course: INFO8665 – Projects in Machine Learning

Team: Group 3 (MemoriAI) – Tessa Nejla Ayvazoglu, Krishna Reddy Bovilla, Adhitya Kondeti, Bhupender Sejwal, Kumari Nikitha

Date: October 2025

# Table of Contents

1. Executive Summary
2. System Context
3. Use Case 1 – Cognitive & Identity Assist
4. Use Case 2 – Daily Reminders & Safety
5. Use Case 3 – Caregiver Dashboard
6. Data Layer & Security
7. Deployment & Operations
8. Key Flows
9. Appendix – OpenAPI Snippets, Risk Matrix & References

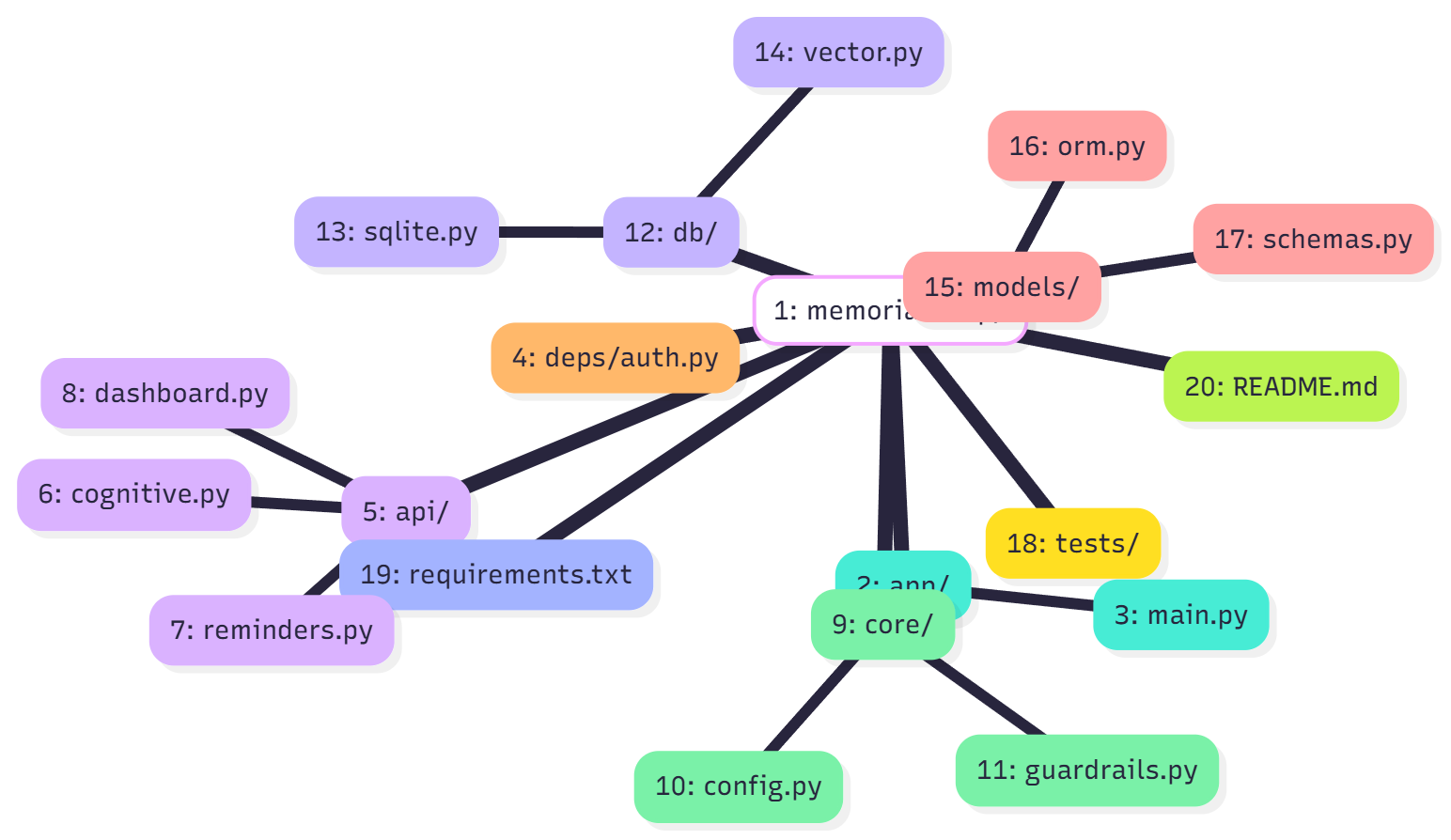
# 1. Executive Summary

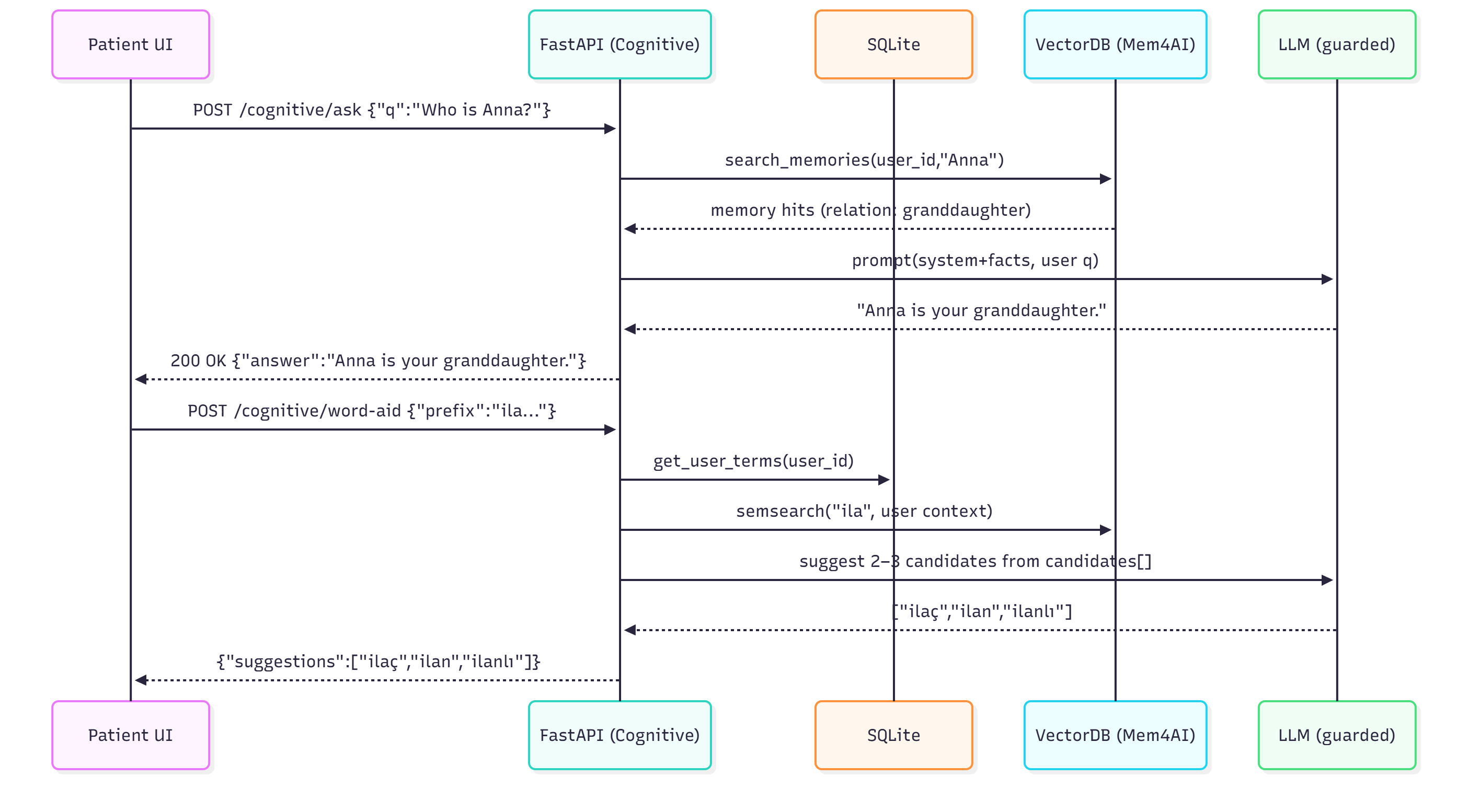
MemoriAI is an AI-powered assistive system supporting Alzheimer’s care through three core pillars: Cognitive & Identity Assist, Daily Reminders & Safety, and Caregiver Dashboard. Each use case includes three alternative API service designs (REST, Microservice, Event-Driven) to evaluate scalability, modularity, and safety.

# 2. System Context

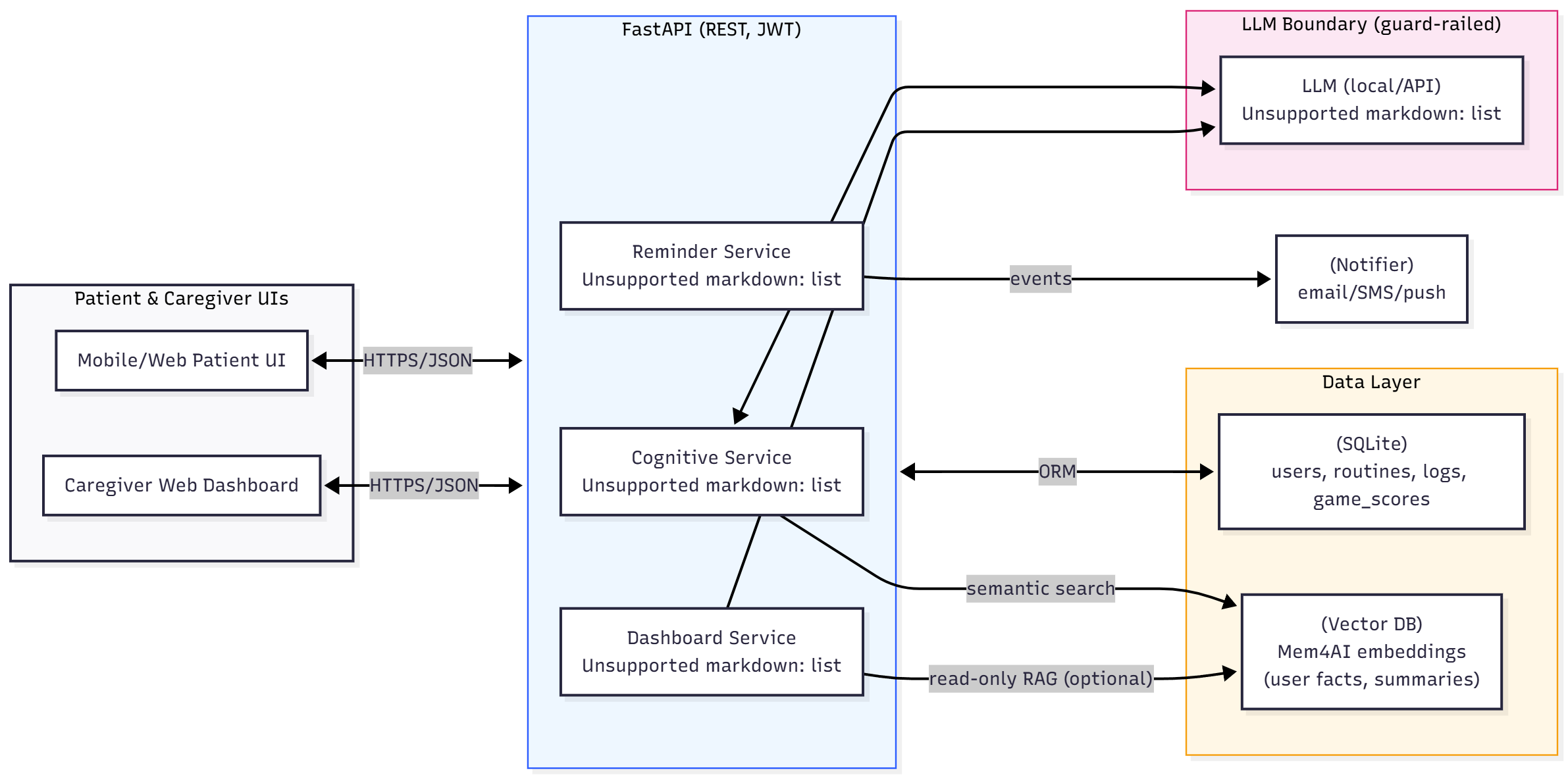
Figure 1 – System Context Diagram

Caption: System context showing external actors and internal services within the secure cloud environment.



Figure 2 – MVP Architecture

Caption: MVP architecture with modular REST services connected via API Gateway in Azure.

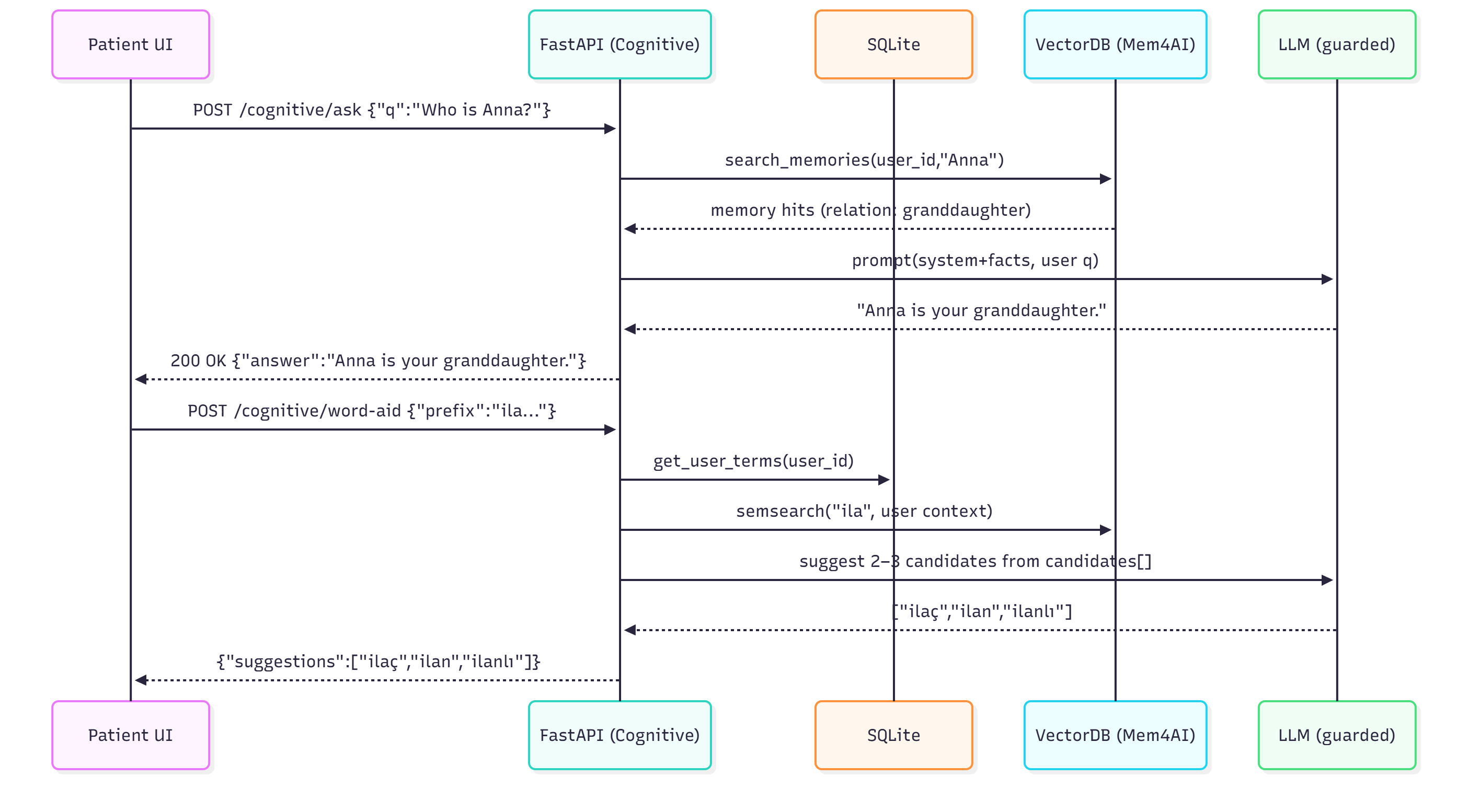
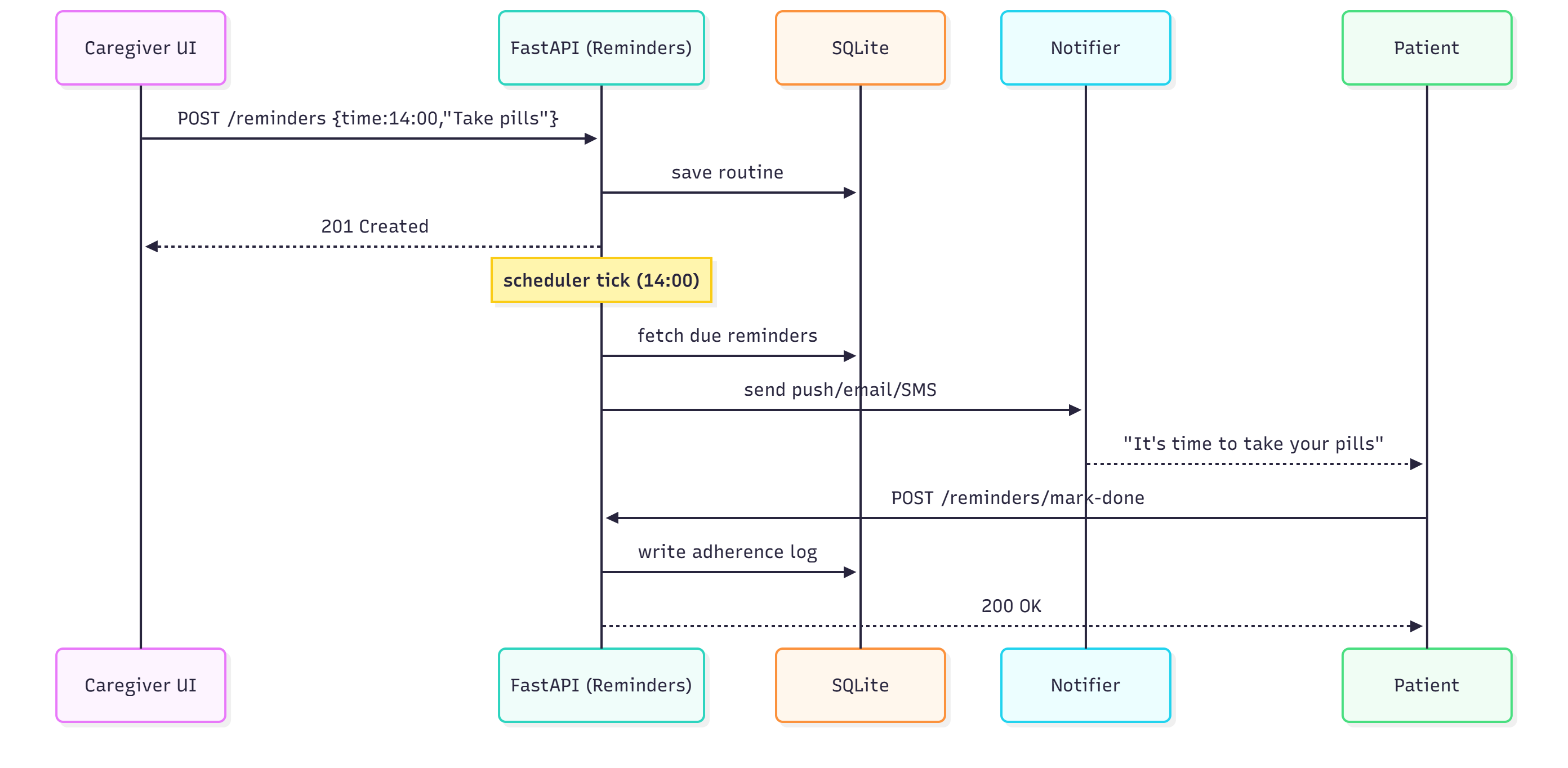


# 3. Use Case 1 – Cognitive & Identity Assist

Goal: Support memory recall and word association safely through explainable AI.  
Primary Actor: Patient  
Secondary Actor: Caregiver

Figure 3 – Cognitive & Identity Flow

Caption: Cognitive & Identity Assist flow representing fact recall and word-aid suggestion through RESTful APIs.



## 3.1 API Service Designs (Use Case 1)

### Design #1 – Synchronous REST (MVP Implementation)

Endpoints: /cognitive/ask, /cognitive/word-aid  
Stack: FastAPI, LangChain, SQLite, Vector DB  
Security: OAuth2 + JWT, RBAC roles (patient/caregiver)  
Low latency (<1.5s p95)

### Design #2 – Split Microservices (Cognitive Core + LLM Adapter)

Public core API delegates phrasing to a private LLM adapter. Strong data boundary (no PII exposure). Scalable and safer.

### Design #3 – Event-Driven Async Design

POST /cognitive/ask (async) returns job\_id. GET /cognitive/jobs/{job\_id} fetches results later. Queue or worker ensures retry and audit logging.

Figure 4 – Design Comparison Table

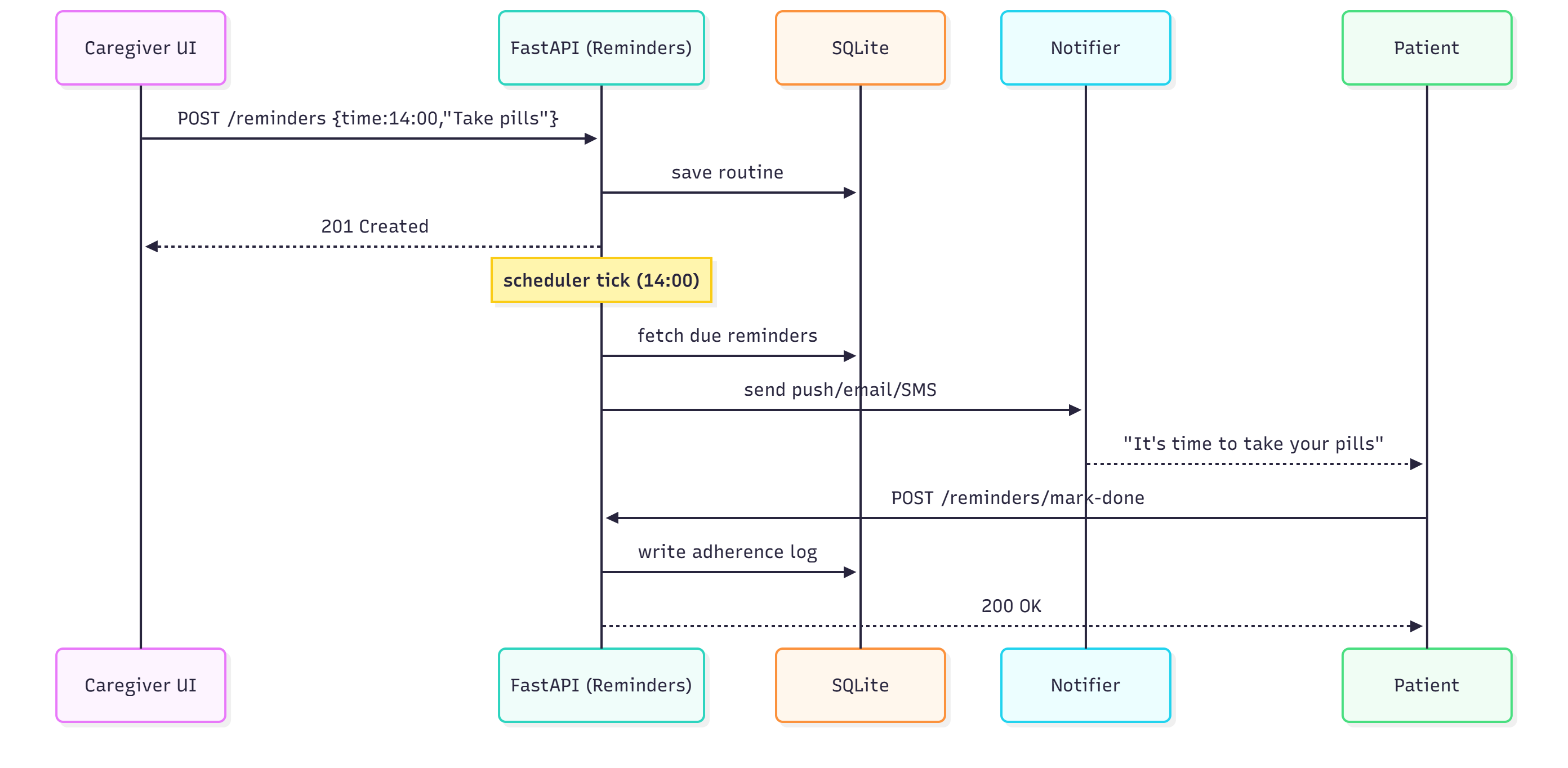
Caption: Comparison of 3 API service designs for Cognitive & Identity Assist.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Design #** | **Architecture Type** | **Core Routes & Functions** | **Key Components / Tech** | **Security & Data Controls** | **Pros / Cons / When to Use** |
| **1. Synchronous REST** | Classic RESTful FastAPI service | /cognitive/ask, /cognitive/word-aid | FastAPI, LangChain, SQLite, Vector DB | OAuth2 + JWT, RBAC, TLS, PII minimization | ✅ Simple, fast for MVP❌ Tightly coupled |
| **2. Split Microservices** | Cognitive Core + LLM Adapter | Same routes; adapter handles phrasing | FastAPI, Docker, internal REST bus | RBAC per service, short-lived tokens | ✅ Safe boundaries, scalable❌ More setup |
| **3. Event-Driven Async** | REST front + worker queue | POST /cognitive/ask (async) → job\_idGET /cognitive/jobs/{id} | FastAPI, Queue/Worker, SQLite | JWT scopes, audit trails, retries | ✅ Reliable & observable❌ More complexity |

# 4. Use Case 2 – Daily Reminders & Safety

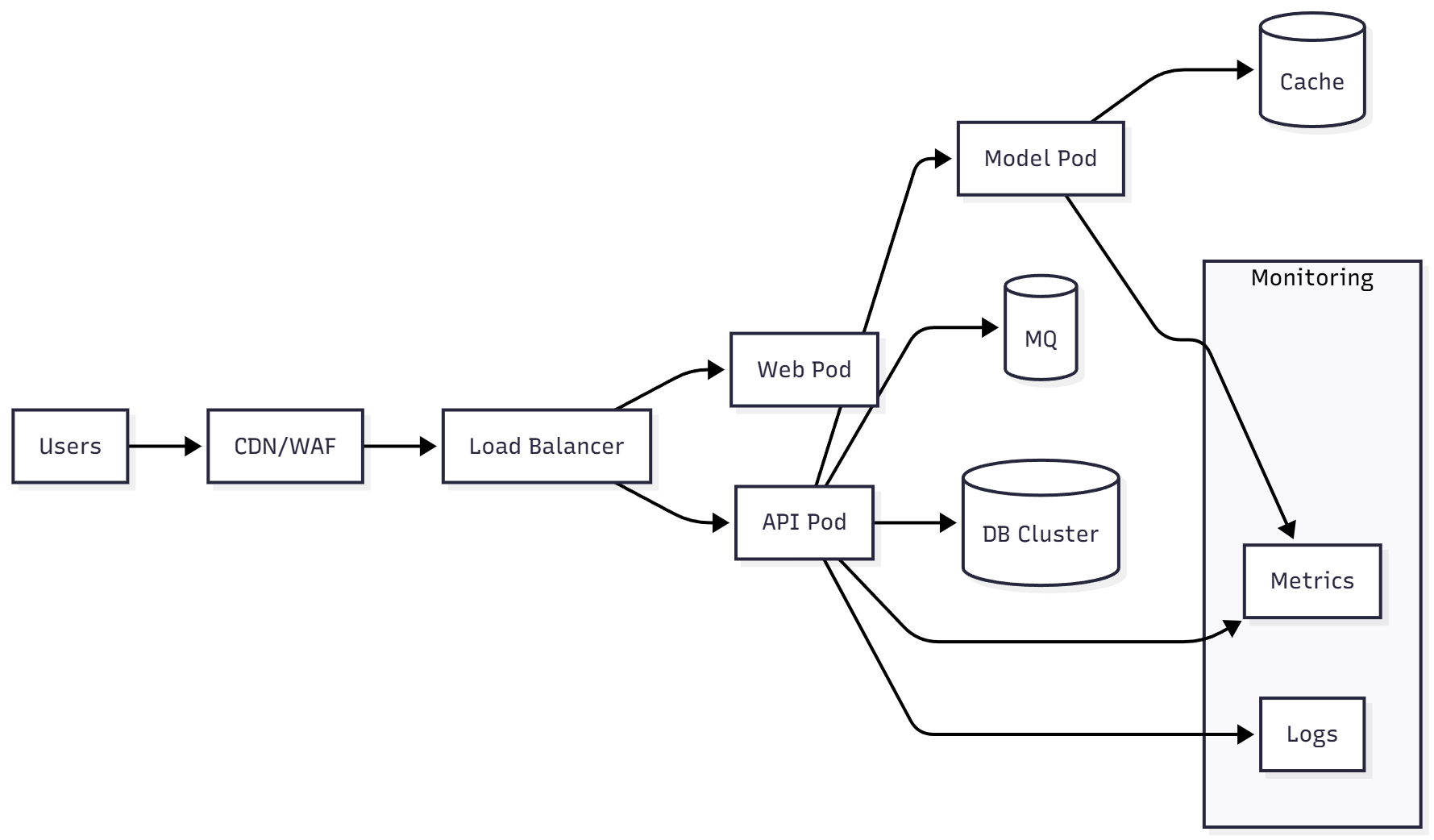
Figure 5 – Daily Reminder Flow

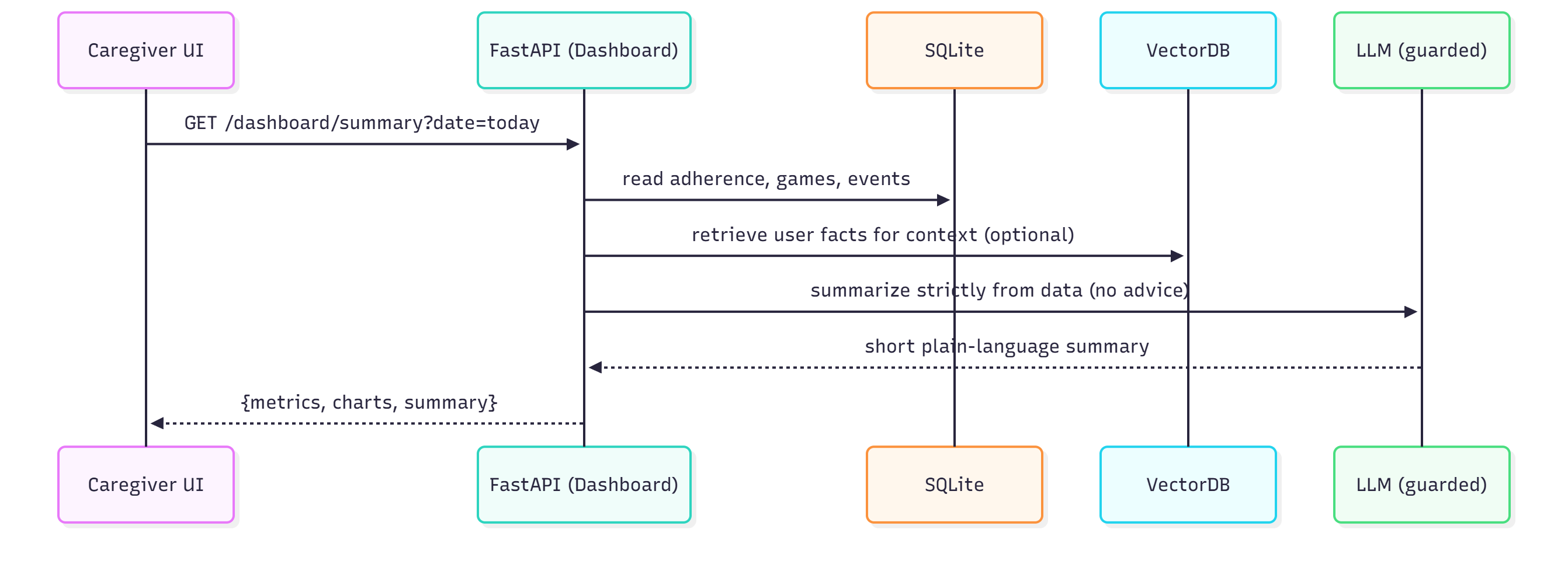
Caption: Flow of reminder scheduling, adherence logging, and caregiver notification.



# 5. Use Case 3 – Caregiver Dashboard

Figure 6 – Caregiver Dashboard Daily Summary

Caption: Caregiver Dashboard summarizing patient adherence, cognitive trends, and notifications.



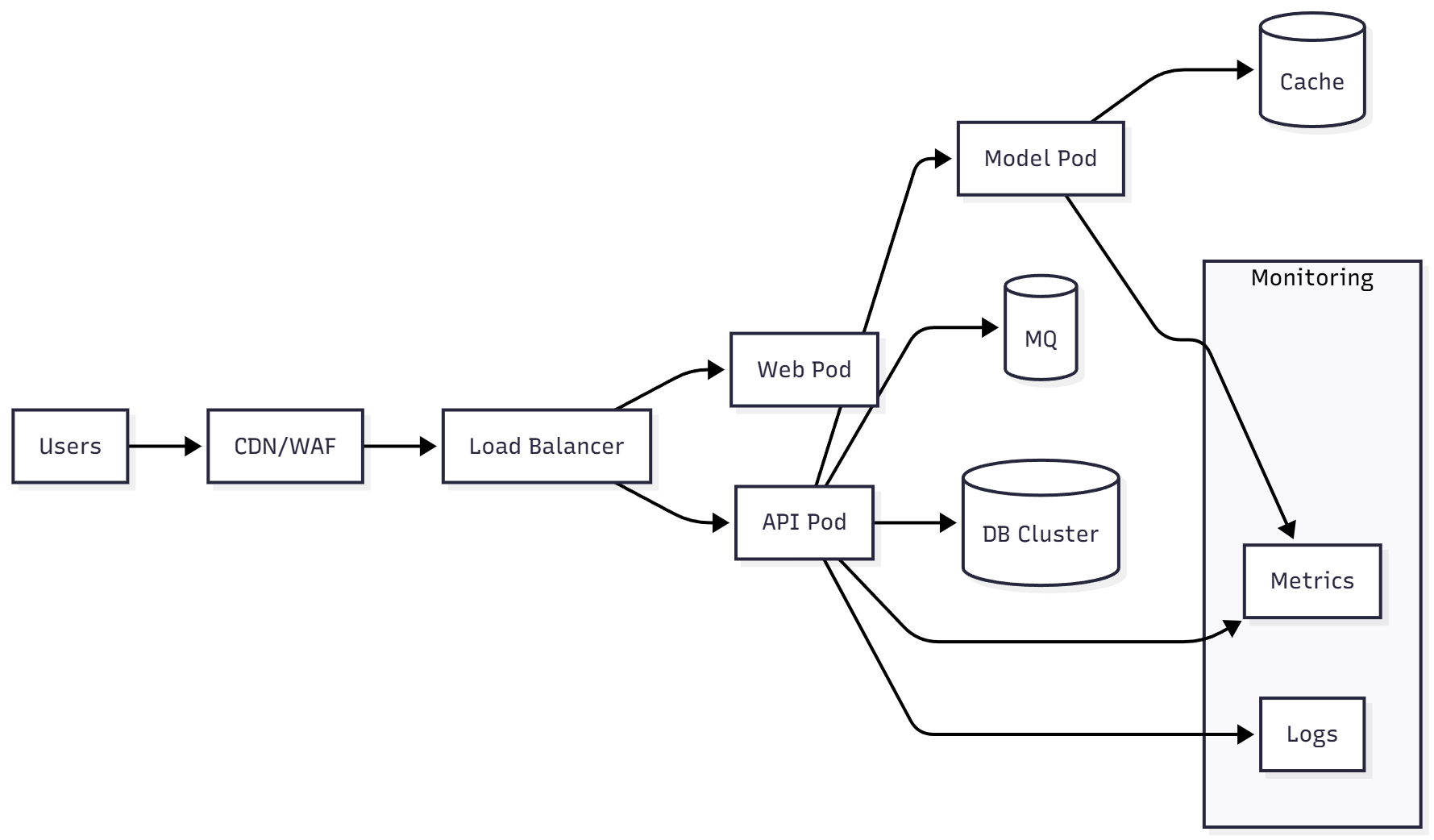
# 6. Data Layer & Security

Core Entities: users, routines, adherence\_logs, game\_scores, mem4ai\_embeddings  
Security Controls:  
- OAuth2 + JWT  
- RBAC (patient/caregiver/admin)  
- TLS 1.3, SQLite encryption  
- Audit logs & guardrails

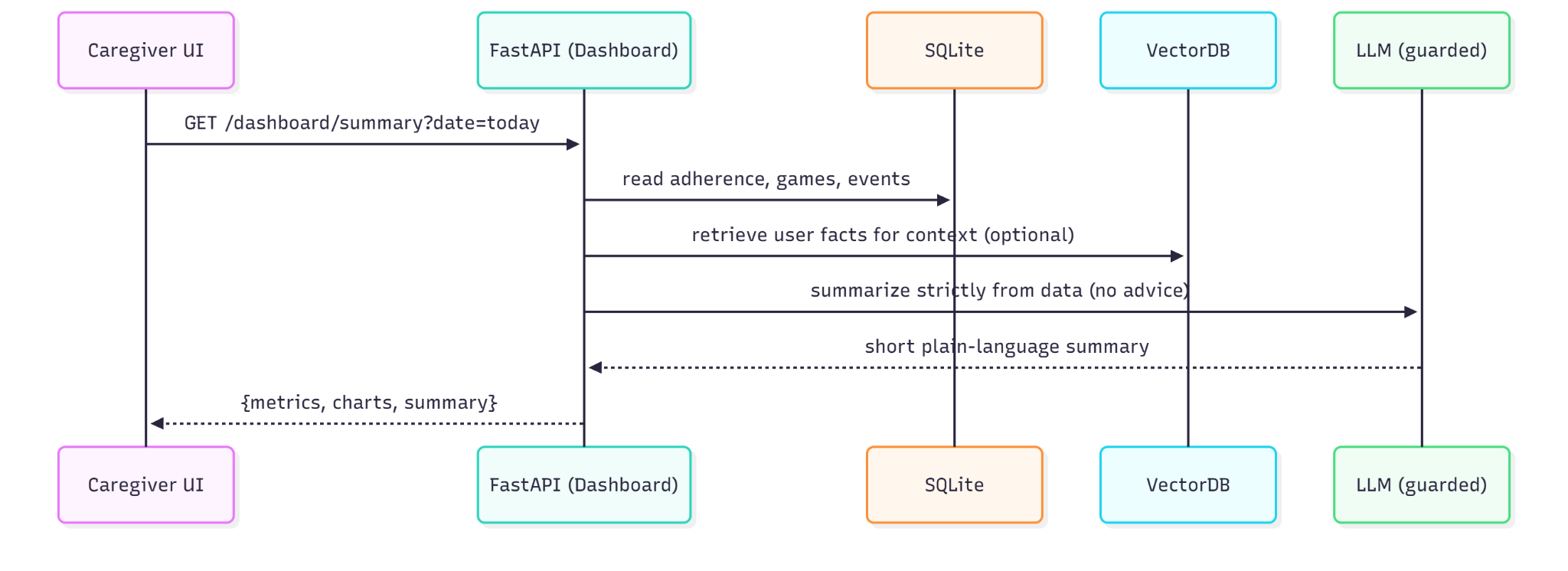
# 7. Deployment & Operations

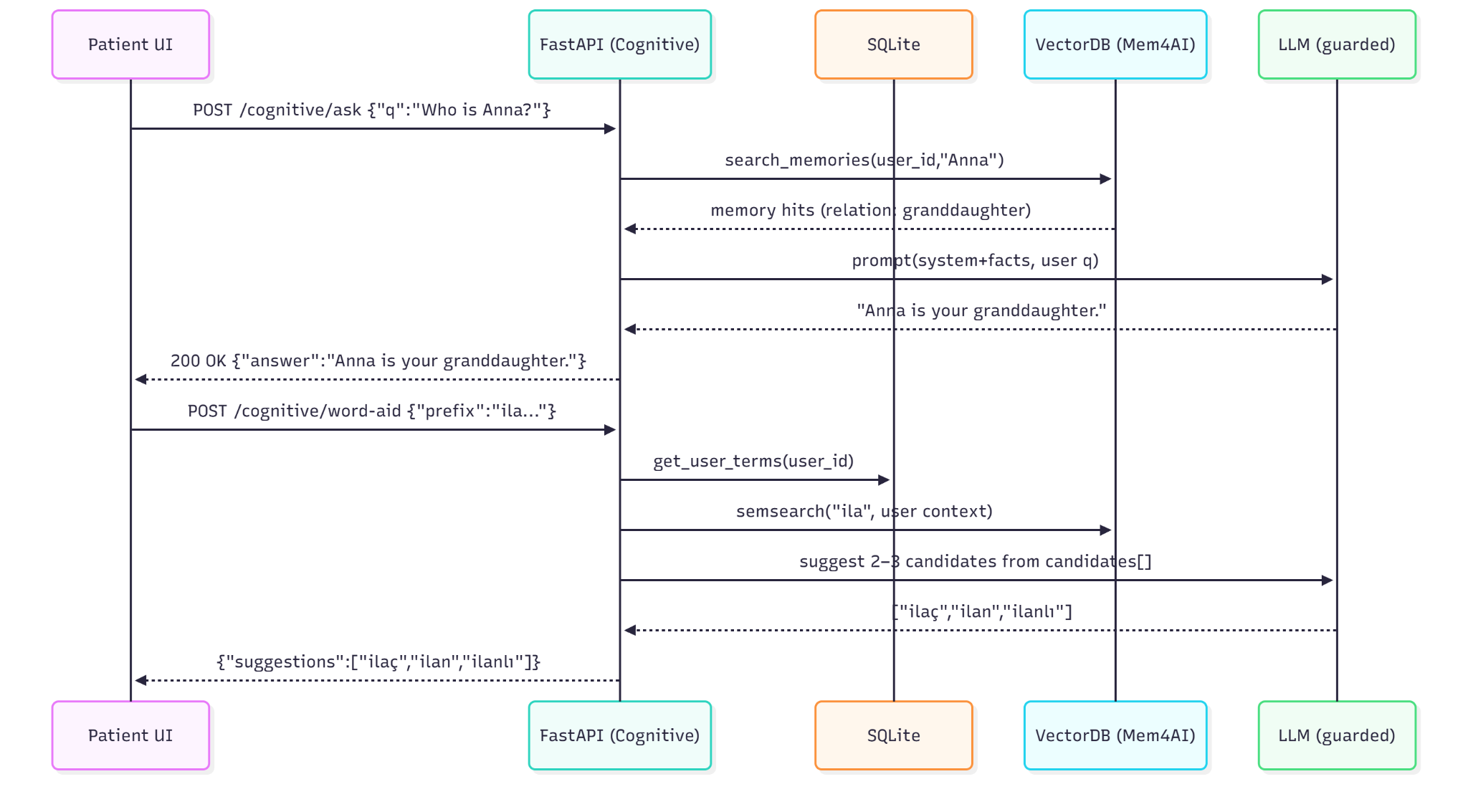
Figure 7 – Deployment Architecture

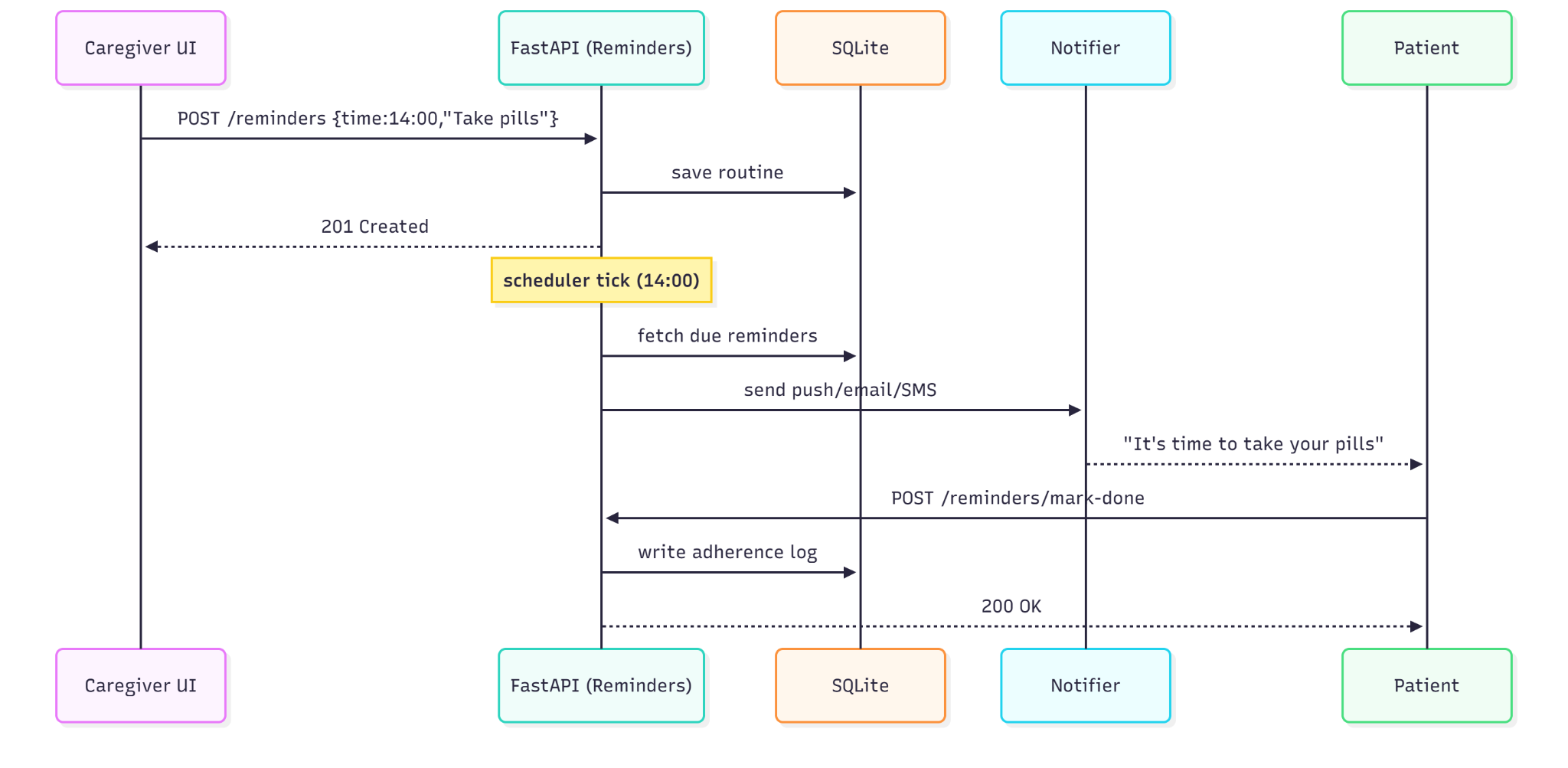
Caption: Azure deployment with containerized FastAPI microservices and CI/CD pipeline.

CI/CD via GitHub Actions → Docker → Azure App Service. Health checks, metrics, logging, and environment isolation (Dev/Test/Prod).

# 8. Key Flows







# 9. Appendix – OpenAPI Snippets, Risk Matrix & References

## 1. OpenAPI Snippets

Below are example API contract snippets representing MemoriAI’s main service endpoints. These follow the OpenAPI-style format to illustrate request and response structures for the MVP design.

* Example – Cognitive Ask API

POST /cognitive/ask  
Summary: Retrieve factual identity information  
Request Example:  
{  
 "q": "Who is Anna?"  
}  
Response Example:  
{  
 "answer": "Anna is your granddaughter."  
}

* Example – Daily Reminder API

POST /reminders  
Summary: Create a new reminder for a patient  
Request Example:  
{  
 "time": "14:00",  
 "text": "Take your medicine"  
}  
Response:  
201 Created

* Example – Async Job API

POST /cognitive/ask (async)  
Request Example:  
{  
 "q": "Who is Anna?",  
 "mode": "async"  
}  
Response:  
202 Accepted  
{  
 "job\_id": "cog\_7f9a1234"  
}  
  
GET /cognitive/jobs/{job\_id}  
Response Example:  
{  
 "status": "done",  
 "answer": "Anna is your granddaughter."  
}

## 2. Risk Matrix

The following table outlines key technical and operational risks for MemoriAI’s MVP phase, with their likelihood, impact, and mitigation strategies.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Risk | Likelihood (1–5) | Impact (1–5) | Priority (L×I) | Mitigation | Owner |
| LLM hallucination | 2 | 5 | 10 | Facts-only recall, guardrails, manual testing | AI Lead |
| Reminder delivery failure | 3 | 4 | 12 | Retries, acknowledgements, synthetic monitoring | Backend Dev |
| PII leakage | 2 | 5 | 10 | Data minimization, encryption, RBAC, audits | Security |
| Scope creep | 4 | 3 | 12 | MVP gating, sprint control | Project Manager |
| Single-node failure | 2 | 4 | 8 | Health checks, restart policies, backups | DevOps |

## 3. References

The following documents and external sources were referenced during the development of the MemoriAI architecture and API design:

* MemoriAI – High-Level Design (HLD) v1, 2025.
* MemoriAI – High-Level Design (HLD) v2 (Advanced Actors), 2025.
* MemoriAI – Architecture Overview, 2025.
* Alzheimer’s Association (2024). Dementia Facts & Figures.
* Brookmeyer et al. (2018). Forecasting Global Burden of Alzheimer’s Disease.
* Etters et al. (2008). Caregiver stress in Alzheimer’s disease.
* Nature (2024). Empowering caregivers with AI.