Technical Architecture & Planning Summary (Python-First Approach)

Foster Ulcer AI is built on a high-availability cloud infrastructure designed to bridge the gap between rural frontline clinics and urban specialists. The architecture priorities data integrity, low-bandwidth resilience, and a "Human-in-the-Loop" AI workflow.

Executive Summary: The "Python-First" Strategy

To accelerate the 30-day deployment, we are adopting a **Python-First** development cycle.

1. **Phase 1 (Python Engine):** Develop the core diagnostic logic, Gemini 2.5 Pro API integration, and data parsing in Python. This acts as the "brain."
2. **Phase 2 (Flutter Integration):** Port the logic or connect the Python engine to the **Flutter Mobile App** via a REST API, allowing the Python code to handle heavy AI processing while Flutter handles the AR capture and UI.

Technical Architecture Overview

Category	Component	Specification / Decision
Prototyping Language	**Python 3.10+**	Rapid validation of AI prompts and data schemas
Mobile Platform	**Flutter**	Cross-platform (iOS/Android) with AR-guided capture
AI Engine	Multi-modal Analysis	**Gemini 2.5 Pro** (Vision + Text)
Backend	API & Logic Layer	Node.js (TypeScript) or FastAPI (Python)
Database	Real-time Data Store	Firebase Firestore (NoSQL) with offline sync
Storage	Medical Imaging	Firebase Storage (High-fidelity BLOBs)
Security	Identity & Access	Firebase Auth (MFA + Role-based Access)
Monitoring	Logging & Health	Google Cloud Logging + Error Reporting
Compliance	Standards	HIPAA-compliant backbone & data encryption

Key Technical Outputs

1. System Architecture Workflow

1. **Capture (Flutter):** Nurse captures wound image via AR "Smart-Frame".

2. **Processing (Python/Gemini):** Image + Metadata processed by the Python engine to output a draft Wagner classification and treatment summary.
3. **Verification (React):** Data appears on the Specialist's dashboard for validation.
4. **Closing the Loop:** Final verified plan synced back to the nurse's device.

2. Python Engine Data Model

- **Assessment Request:** `{ "image_b64": str, "patient_vitals": dict, "history": list }`
- **Gemini Response Parsing:** Uses Python's `google-generativeai` library to extract structured JSON from multi-modal prompts.

3. Reliability & Maintenance

- **Local Simulation:** Python scripts allow developers to simulate low-bandwidth scenarios by throttling image uploads.
- **Validation:** Specialists can "blind test" the Python AI output against manual records to calculate accuracy before the Flutter launch.