HIPAA Compliance & Security Strategy

Document Purpose

This document delineates the security architecture and compliance strategy for the AI-Powered FHIR Query System. The framework here in is designed to ensure the confidentiality, integrity, and availability of all Protected Health Information (PHI), adhering rigorously to the technical safeguards mandated by the Health Insurance Portability and Accountability Act (HIPAA).

1. Authentication & Authorization (SMART on FHIR)

The system's security is anchored by the **SMART on FHIR** framework, utilizing **OAuth 2.0** to delegate authentication to a trusted Identity Provider (IdP) and eliminate local credential management.

- **Federated Identity:** Users authenticate directly with the IdP. The application never handles or stores user credentials.
- **Scoped Access:** Upon user consent, the IdP issues short-lived JWT access tokens with fine-grained permission scopes (e.g., `patient/Patient.read`), enforcing the Principle of Least Privilege on every API request.

2. Data Privacy & Auditing

A multi-layered strategy ensures data is protected and all access is tracked.

- **End-to-End Encryption:** All data is secured in transit using TLS 1.2+ and at rest using AES-256. The application is designed to be stateless.
- **Data Minimization:** The NLP service is engineered to construct queries that retrieve only the essential data elements required to fulfill a user's request, strictly adhering to the HIPAA "Minimum Necessary" rule.
- **Immutable Audit Trail:** A secure, tamper-proof log records all PHI access events, capturing the user's identity (Who), the specific action taken (What), the event timestamp (When), and the source IP address (Where).

3. Role-Based Access Control (RBAC)

Access control is managed externally by the IdP, ensuring a secure and maintainable separation of duties.

- **Externally Managed Roles:** User roles (e.g., Clinician, Researcher) and their associated permissions are defined and managed centrally by the authoritative IdP, not the application.
- **Dynamic UI Enforcement:** The application inspects the scopes within a user's JWT in real-time to dynamically enable or disable features, architecturally preventing any attempts to perform unauthorized actions.