TECHNICAL BRIEF

Secure AI-Assisted Clinical Note System (Demo Prototype)

1. SYSTEM OVERVIEW & OBJECTIVE

This project presents a secure, AI-assisted clinical note system designed for a single-patient workflow. The system focuses on:

- Reducing clinician documentation burden
- Preserving traceability and data provenance
- Enforcing strict access control (RBAC)
- Preventing data leakage during AI-assisted writing

This is a demonstration prototype intended to show system design, governance thinking, and technical trade-offs rather than a production-scale deployment.

2. ARCHITECTURE OVERVIEW

The system follows a modular architecture with clear separation between user interaction, data governance, and AI-assisted logic.

High-level flow:

[Clinician UI (Single-Patient Page)]
|
v
[Application Layer]
- Entry management
- Commenting & highlights
- Version control
|
v
[Governance Layer]
- Redaction engine
- RBAC enforcement
- Provenance tracking
|
v
[AI Scribing Module]
- Receives redacted context only
- Generates draft notes
|
v
[Encrypted Storage Layer]
- TLS in transit
- Encryption at rest

Key design principle:
AI never directly accesses raw sensitive data without redaction.

3. DATA MODEL & RELATIONSHIPS
All data entities are linked to a single patient page.

Core entities:
• **Entry**
- Represents a clinician-authored note or update. Primary anchor entity.

• **Comment**
- Attached to a specific Entry. Used for clarification or peer review.

• **Version**
- Snapshot of an Entry at a given time. Enables auditability and rollback.

• **Highlight**
- Marks specific text spans within an Entry. Used for attention, tagging, or AI context selection.

• **Provenance**
- Metadata layer. Records author, timestamp, role, and source of changes.

• **AI_Scribed_Notes**
- AI-generated draft outputs. Always linked to a specific Entry and Version.

Relationship structure (simplified):

```
Entry
 ├── Versions
 │    ├── Highlights
 │    └── Provenance
 ├── Comments
 └── AI_Scribed_Notes
```

4. SECURITY, REDACTION & RBAC
Security is enforced by design rather than by policy alone.

Encryption:
- • TLS for all data in transit
- • Encrypted storage for all persisted data

Redaction:
- • Sensitive fields are stripped or masked before AI access
- • Redaction occurs at the governance layer
- • Raw data never leaves the secure boundary

RBAC (Role-Based Access Control):
- • Roles: clinician, reviewer, admin

- • Permissions enforced at API and data-access level
- • AI operates under the lowest-privilege role

5. AI INTEGRATION & LEARNING SCOPE

The AI component functions as an assistive drafting tool.

Key constraints:

- • No autonomous decision-making
- • No direct write access to authoritative records
- • Outputs require human review and approval

Learning mechanism:

- • No online learning from live patient data
- • Demo assumes stateless inference only
- • Prevents unintended data memorization

6. ASSUMPTIONS & FIRST-PRINCIPLES THINKING

Key assumptions:

- • Single-patient page simplifies navigation and access control
- • Clinicians require transparency over AI outputs
- • Auditability is more important than automation speed

First-principles approach:

- • Every AI action must be explainable
- • Every text change must be traceable
- • Governance is part of system architecture, not an add-on

7. TRADE-OFFS & SCOPE DECISIONS

Deliberate trade-offs:

- • Reduced automation in exchange for safety
- • Limited learning capability to prevent data leakage
- • Narrow clinical scope to ensure clarity of design

Out of scope:

- • Multi-patient dashboards
- • Cross-institution data sharing
- • Real-time AI retraining

8. CONCLUSION

This system demonstrates how AI-assisted clinical documentation can be designed with security, governance, and human oversight as first-class concerns.

The prototype prioritizes trust, traceability, and compliance over raw AI capability, reflecting real-world healthcare needs.