

Emergency Access Health Data System (EA-HDS)

Conceptual Design Architecture

1. Overview

The Emergency Access Health Data System (EA-HDS) is designed to provide fast, secure, and limited access to critical patient health data during emergencies. It enables paramedics and certified first responders to access essential information (e.g., allergies, medications, chronic illnesses) without exposing full Electronic Health Records (EHRs). The design prioritizes selective disclosure, rapid response, strong security, complete auditability, and patient-defined control.

2. System Components

- **Encrypted Health Data Vault (EHDV):** Secure cloud-based repository that stores encrypted EHRs.
 - **Emergency Access Interface (EAI):** Secure API through which certified emergency responders request access.
 - **Patient Policy Dashboard (PPD):** A web/mobile platform that allows patients to set and update emergency data access preferences.
 - **Distributed Ledger System (DLS):** Tamper-proof blockchain that stores access policies and audit logs.
 - **Responder Access Devices (RAD):** Tablets/phones with biometric authentication and TEE support used by certified responders.
 - **Secure Hardware Module (SHM):** Built-in Trusted Execution Environment (TEE) within responder devices for secure decryption and display.
-

3. Protocol Workflow

Step 1: Patient Policy Definition

- Patients log into the PPD and define which medical data is accessible during emergencies (e.g., blood type, allergy list).
 - Access conditions can include responder type, location, and time constraints.
 - Policies are hashed and published to the Distributed Ledger System (DLS).
 - EHRs are encrypted using **AES-256**, and the emergency data subset is segmented.
 - A short-lived emergency decryption key is created and protected using **threshold encryption** (e.g., 2-of-3 trusted authority decryption).
-

Step 2: Emergency Access Request

- Responder authenticates using biometric verification and a valid responder certificate.
 - The RAD captures contextual data (location, timestamp) and sends an access request with the patient's identifier (via QR code, NFC, or biometric match) to the EAI.
 - The request is verified by the device's **TEE** before being transmitted.
-

Step 3: Policy Validation & Key Retrieval

- The EAI checks the DLS for the corresponding patient access policy hash.
 - If the request satisfies the policy, it triggers a **threshold decryption** process involving any 2 of 3 trusted authorities (e.g., EMS server, hospital certifier, regional admin).
 - Once the emergency key is decrypted, it is used to decrypt only the authorized data subset inside the responder's **TEE**.
-

Step 4: Access Logging & Auditing

- Each access event (who, when, where, why) is digitally signed and appended to the **DLS**.
 - These logs are immutable, timestamped, and visible to patients through the PPD.
 - Patients can review and flag unauthorized access, and all logs are revocable or terminable in real-time.
-

Step 5: Patient-Centric Control

- Patients can update their emergency data set or access conditions at any time via the PPD.
 - Emergency access can be temporarily disabled (e.g., during international travel).
 - Revocation of specific responders or institutions is supported.
 - Smart wearables (e.g., wristbands or ID tags) contain QR codes linked to the latest policy pointers.
-

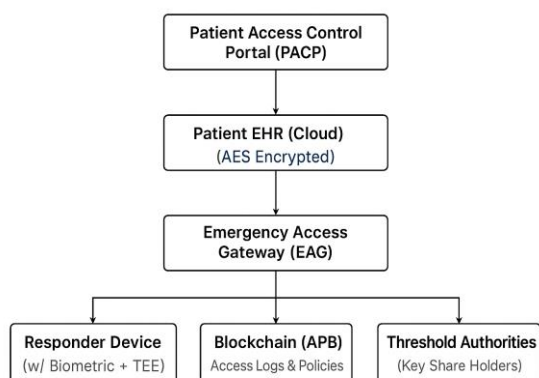
4. Security Design Summary

Requirement	Feature(s) Used	Explanation
Selective Disclosure	Data segmentation + threshold decryption	Only the predefined emergency subset is decrypted; rest of the EHR remains encrypted.
Rapid Access	Local TEE + AES-256 + minimal interaction	Emergency data is decrypted within seconds inside the responder's secure device.
Security Assurance	AES-256 encryption, TEE, blockchain logging	Full EHR is never exposed; decryption only in secure enclave; access attempts are logged.
Accountability	Blockchain-based logging + digital signatures	All access attempts are tamper-proof, traceable, and visible to the patient.
Patient Control	PPD access policy editor + QR/wearable updates + revocation support	Patients retain continuous control over what responders can access and when.

5. Practical Considerations & Trade-Offs

- **Offline Emergency Cache:** Devices may maintain an encrypted offline cache of emergency data that can be unlocked using cached threshold keys and TEE validation.
- **Latency vs. Security:** Use of AES-256 ensures low decryption latency. Threshold key retrieval may add a few seconds but remains acceptable for emergency use.
- **Scalability:** Decentralized ledger allows secure and efficient scaling as the number of users and responders grows.
- **Privacy Protections:** Data access is tightly scoped and revocable, and TEE ensures no unauthorized copying even from within the device.

6. System Diagram



7. Conclusion

This EA-HDS design meets all five mandated requirements by integrating modern cryptographic techniques (AES-256, threshold decryption), secure hardware (TEE), and distributed trust (blockchain). It ensures emergency teams can quickly access only what they need while preserving patient privacy, enabling auditability, and respecting individual control preferences. The design is scalable, secure, and adaptable for diverse emergency settings.