System Architecture

1. Components & responsibilities

End User Device

- Role: user UI, question submission, display answer + confidence/provenance.
- Interface: Web UI (PWA) or light app; offline caching of recent answers.
- Tech hint: (HTML/JS PWA, IndexedDB for local cache)

Mini Hub (mH) — Edge gateway (per site)

- Exposes local Wi-Fi AP and HTTP endpoint for user queries.
- Performs: embedding generation (light), ANN lookup, cache decision, packetization, local fallback inference, capsule ingestion, peer gossip, queueing of pending queries.
- Stores: local Capsule DB + embedding index + pending queue.
- Manages keys/session and local telemetry.
- Tech hint: (SBC + LoRa module; local HTTP server; ANN index)

Main Hub (MH) — Regional brain

- Receives/assembles encrypted packets, performs authoritative inference, generates & signs Knowledge Capsules, stores cluster DB and master index, issues manifests and orchestrates sync.
- Admin console, analytics, OTA server (optional).
- Tech hint: (Higher compute SBC/mini-PC, SSD storage, model runtime)

Long-Range Transport / Mesh

- Role: LR link for mH↔MH and mH↔mH (capsules, manifests, query packets).
- Behavior: low-bandwidth, chunking, ACK/retransmit, manifest-driven selective sync, gossip with duplicate suppression.
- Tech hint: (LoRa / directional Wi-Fi bridges; manifest + chunk protocol)

Cloud / Admin Console (optional)

- For remote analytics, bulk model hosting, OTA distribution, and cross-cluster federation when internet available.
- Tech hint: (Cloud APIs, dashboards)

2. Core data artifacts (abstract)

- **QUERY_PKT** — encrypted packet carrying question + meta (q_hash, emb_hint, priority).
- **RESPONSE_PKT** — encrypted answer + provenance + capsule reference.
- **KNOWLEDGE_CAPSULE** — compact Q-A unit: {id, question_hash, answer, compressed_embedding, source, timestamp, signature}.
- **MANIFEST** — list of capsule IDs/versions for selective sync.

(Encryption: payloads encrypted with AEAD — AES-GCM or equivalent; capsule integrity signed with asymmetric keys — Ed25519 or similar.)

3. Sequence flows (detailed)

Flow A — Normal Query (Cache miss → authoritative)

1. User → POST /query → mH (HTTP).
2. mH computes lightweight embedding → ANN lookup.
3. No acceptable capsule found → mH forms QUERY_PKT (q_hash + enc payload) and transmits to MH via LR link (chunked + AES-GCM).
4. MH reassembles, decrypts, invokes LLM dispatcher → authoritative answer.
5. MH creates KNOWLEDGE_CAPSULE, signs it, stores in cluster DB.
6. MH returns RESPONSE_PKT to mH.
7. mH verifies signature, stores capsule, updates ANN index, serves answer to user.

(Notes: reliability via ACK/NAK and sequence numbers; manifest update scheduled.)

Flow B — Cache Hit (local)

1. User → mH → compute embedding → ANN returns capsule with sim ≥ T_sim.
2. mH serves cached answer immediately (include confidence and source=LocalCache).
3. Log usage; optionally mark capsule for higher retention.

Flow C — Offline Fallback (MH unreachable)

1. mH detects MH unreachable (heartbeat timeout).
2. mH tries ANN lookup; if none, runs local fallback (short quantized model) to produce provisional answer and queues PENDING_PKT.
3. On reconnection, mH forwards queued packets; MH processes and issues authoritative capsules; mH reconciles previous provisional answers.

Flow D — Capsule Propagation & Gossip

1. MH generates MANIFEST (capsule IDs + digests) and broadcasts to mHs.
2. mHs selectively request missing capsules (manifest-driven fetch).
3. Peer mHs gossip capsules to neighbors with TTL and duplicate suppression (bloom filters) to increase local availability.

4. Cache decision & ANN logic (abstract)

- Compute query embedding e_q.
- ANN nearest neighbor search → get max_sim.
- If max_sim ≥ T_sim and capsule_age ≤ TTL → local serve; else forward.
- Eviction policy: LRU + usage-weighted retention; capsules may be promoted by MH via manifest.

5. Security & trust model

- **Node identity**: asymmetric keypair per node (Ed25519).
- **Session keys**: ephemeral DH (X25519) → derive AES-GCM keys for transport.
- **Capsule signature**: origin signed by source node; recipients verify before accept.
- **Bootstrapping**: provisioning via QR/USB or preloaded configs; MH as initial trust anchor.
- **Revocation**: revocation list distributed in manifests.

6. Operational modes

- **Online Mode**: MH reachable → full quality, frequent manifests.
- **Offline Mode**: MH unreachable → local cache + fallback LLM answers; queue pending queries.
- **Learning Mode**: periodic capsule exchange and ANN rebuilds.
- **Recovery Mode**: bulk sync and reconciliation.

7. Deployment & provisioning (brief)

- Provision node keys and config at factory or on-site (QR/USB).
- MH registers mH public key; mH seeds initial capsule set.
- mH auto-configures Wi-Fi SSID for users.
- OTA via signed packages over opportunistic uplink (MH→mH or cloud→MH→mH).

8. Data formats (examples)

- Query/response/capsule: JSON for control + compressed binary for embeddings.
- Capsule size goal: compact (< 1 KB typical).
- Embeddings: quantized (int8 / float16) and compressed (zstd/lz4).

9. All elements included in the project

- End user UI (Web/PWA, optional mobile app)
- Mini Hubs (AP, local server, cache, ANN, fallback LLM, LoRa)
- Main Hub (dispatcher, capsule manager, DB, manifest generator)
- Long-range transport (LR protocol, chunking, ACKs)
- Peer mesh (gossip, duplication suppression)
- Knowledge Capsules (creation, signing, indexing)
- Security (node keys, sessions, encryption, revocation)
- Provisioning & OTA system
- Monitoring & telemetry (local metrics, optional cloud dashboards)
- Power & enclosure considerations (PoE/solar, weatherproof enclosures)
- Testing & validation harness (crypto tests, chunking tests, ANN accuracy tests)

10. Short tech legend (very short)

- Local access: HTTP/HTTPS (Web UI)
- LR transport: LoRa / directional Wi-Fi (chunked + ACK)
- Local DB: SQLite on mH; SSD DB on MH
- Embedding & ANN: quantized embeddings + HNSW/HNSW-like index

- Models: quantized LLMs on edge; larger model on MH or cloud fallback
- Crypto: X25519 (session), AES-GCM (AEAD), Ed25519 (signatures)

11. Next practical deliverables (recommended)

- Create architecture diagram (SVG/PNG) for slides.
- Produce sequence diagrams (PlantUML) for the four flows.
- Stage-1 code scaffold (mini_hub + main_hub + common packet & crypto).
- Prototype: simulated LoRa via sockets to validate packetization, signing, cache logic.