Solution

1. High-level architecture (conceptual)

- **Main Hub (central node):** Acts as the authoritative processing and coordination point for a regional cluster. Responsible for high-quality inference, capsule generation, capsule verification, and cluster orchestration.
- **Mini Hubs (edge nodes):** Distributed access points that interface with end users, perform packetization, local cache lookup, lightweight inference fallback, and peer exchange of knowledge.
- **End user devices:** Connect to a nearby mini hub through a local access interface to submit queries and receive answers.
- **Transport plane:** Two-tier communications — a local access channel between user and mini hub, and a long-range, low-power channel between mini hubs and the main hub (plus opportunistic higher-throughput links for bulk sync).

2. Core functional flows

1. **Query flow (normal):** User → mini hub → if local cache hit: respond; else forward to main hub → main hub processes and returns answer → mini hub delivers to user and stores resulting capsule.
2. **Fallback/Offline flow:** If main hub unreachable, mini hub serves from local cache or uses lightweight local inference to generate an answer and queues the query for later reconciliation.
3. **Capsule propagation:** Main hub issues compact, signed knowledge capsules after authoritative answers. Capsules are disseminated to mini hubs and peer hubs via controlled broadcast/gossip to improve local hit rates.
4. **Recovery flow:** When connectivity to main hub is restored, queued queries and local logs are synchronized; authoritative capsules and model updates are reconciled.

3. Data & capsule model

- **Query packet:** Unique ID, source/destination identifiers, timestamp, encrypted payload containing question and optional context, and light metadata (hashes, compressed embedding hints, priority).

- **Knowledge capsule:** Compact record containing question fingerprint, concise answer, compressed embedding, source provenance, timestamp, and a cryptographic signature for origin verification.
- **Sync metadata:** Small manifests listing available capsules/IDs and concise digests for deduplication and selective retrieval.

Design goals for data items:

- Capsules should be compact (designed to fit constrained long-range payloads).
- Embeddings must be compressed/quantized to reduce transmission size while preserving similarity semantics.
- Deterministic hashing enables deduplication and conflict resolution.

4. Communication & routing (constraints-aware)

- **Low-bandwidth long-range channel:** Use it for compact query packets, capsule IDs/metadata, and short answers. Larger payloads are chunked, compressed, and delivered opportunistically or via higher-throughput links when available.
- **Local access channel:** Optimized for low-latency user interactions within the mini hub's coverage area.
- **Mesh behavior:** Mini hubs can directly exchange capsules with neighbors to reduce upstream load and provide redundancy. Exchange is governed by gossip with duplicate suppression and rate controls.
- **Reliability mechanisms:** ACK/NAK, sequence numbering, chunk reassembly, and local queuing with exponential backoff are applied to handle intermittent links and high packet loss.

5. Local inference & caching strategy (conceptual)

- **Local inference fallback:** Mini hubs run compact, quantized models with limited context to provide approximate answers when upstream is unavailable.
- **Cache decision policy:** Compute similarity between query embedding and stored capsule embeddings. If maximum similarity ≥ configured threshold and answer age ≤ configured freshness window, serve from cache; otherwise forward to main hub.
- **ANN index:** Maintain a lightweight approximate nearest neighbour index for fast similarity lookup at the edge (memory-constrained representation).
- **Answer confidence & metadata:** Cached answers include confidence score, provenance, and timestamp; UIs display confidence to users and allow escalation.

6. Knowledge lifecycle & learning dynamics

- **Creation:** Main hub generates capsules from authoritative answers and signs them for trust.
- **Distribution:** Capsules are shared to mini hubs on scheduled windows and via opportunistic peer exchange; only compact capsule representations are broadcast routinely.
- **Retention policy:** Each capsule has metadata controlling TTL, retention priority, and privacy tags; storage is bounded and local garbage collection removes stale or low-value capsules.
- **Local adaptation:** Mini hubs may augment capsules with local annotations (e.g., region-specific vocabulary) but such augmentations are flagged and, unless authorized, not promoted as authoritative.
- **Federated growth:** Over time, capsule sharing yields a distributed knowledge graph that increases local hit rates and reduces dependency on central queries.

7. Security, privacy & trust (abstract)

- **Confidentiality:** All payloads are encrypted in transit and at rest on nodes; keys are managed through a secure bootstrapping and ephemeral session process.
- **Integrity & provenance:** Capsules and critical control messages are cryptographically signed to prevent tampering and to enable origin verification.
- **Access control:** Nodes enforce rate limits and content filters; sensitive categories may be redacted or require manual vetting before capsule creation.
- **Privacy safeguards:** Local retention policies, optional PII redaction before capsuleization, and configurable anonymization steps prevent unnecessary exposure of user data.

8. Operational modes & behavior

- **Online mode:** Main hub reachable — full quality responses and frequent capsule distribution.
- **Offline mode:** Main hub unreachable — mini hubs rely on cache and local inference; queries are queued for later reconciliation.
- **Learning mode:** Periodic capsule exchange and index updates across the cluster to converge on shared knowledge.
- **Recovery mode:** Bulk resynchronization of queued queries, capsule manifests, and integrity checks when links recover.