How RetireChain Batching Keeps Fees Tiny

The idea in one line

Instead of writing every single contribution to the blockchain, we write one compact proof that mathematically represents hundreds or thousands of contributions at once.

How it works (step-by-step)

1. Collect records

   Take a chunk of verified rows (e.g., all contributions from a payroll run or a 5-minute window).

2. Hash each record

   Turn each row into a short “fingerprint” (a cryptographic hash).

   - Only the hash is ever used on-chain; raw data stays private.

3. Build a Merkle tree

   Combine those hashes pairwise until you get one top hash: the Merkle root.

   - That single root uniquely represents every record in the batch.

4. Anchor the root on-chain

   Write just the Merkle root (plus tiny metadata like batch_id, timestamp, schema version) to the blockchain.

   - One on-chain write now covers the entire batch.

5. Store the batch off-chain (secure)

   The full batch (encrypted) lives in your database or secure object storage.

  - Auditors/partners can verify any record by recomputing its hash and checking it matches the anchored root.

6. Prove any single record later

   When someone asks “was Jane Doe’s $237.50 contribution recorded?” you return a tiny Merkle proof (a few hashes).

   - They combine it with the root on-chain → cryptographic yes/no, no trust required.

Why this is so cheap

- On-chain cost ≈ 1 write per batch, not per record.
- On fast, low-fee chains (like Solana), each anchor typically costs fractions of a cent, so:
  - 10,000 records → 1 write → ~<$0.001 total (order-of-magnitude), not $10–$100+ in old-world ops time.

Batching strategies (simple playbook)

- By time: anchor every N minutes (e.g., 1, 5, or 15).
- By size: anchor when a batch hits N records (e.g., 5k–20k).
- Hybrid: whichever comes first (keeps latency low but fees tiny).

What happens if a mistake is found?

- You never edit the old proof.

- You add a correction record in the next batch that points to the original and shows the fix.
- The chain becomes a clean, append-only audit trail.

Privacy & compliance

- On-chain: only hashes + minimal metadata (no PII).
- Off-chain: encrypted data in your controlled environment.
- Add salts to hashes so they can’t be guessed from public info.

Operational polish to mention to partners/investors

- SLAs: e.g., “proof anchored within 5 minutes of receipt.”
- Monitoring: alert if no anchor happened in expected window.
- Versioning: store a schema hash so everyone knows which data format a batch used.
- Chain-agnostic: same design works on Solana, Ethereum L2s, etc.
- Disaster-safe: redundant storage for off-chain batches; roots are immutable on-chain.
