# Ghost Mesh 48 – Memory■Crystal Defragmenter (APP■MCD)

Hash■Deduplication & Aged■Fragment Reclamation Service

## 1. Purpose

Prevent unbounded storage growth and drift noise by deduplicating identical motif hashes and retiring stale Memory■Crystal fragments older than a preset cycle threshold or flagged as obsolete by Fusion Fold consolidations.

## 2. Trigger & Cadence

Runs every 50 cycles (configurable) as a low■priority background task.
Skips execution if Emergency Hollow Echo Cascade active.

## 3. Deduplication Algorithm

- Build SHA■256 hash index of all crystal motif blobs.
- If duplicate hashes detected, merge references and delete redundant copy.
- Update any Dreamling pointers atomically to prevent dangling links.

## 4. Aged■Fragment Reclamation

- Determine `age_cycles` = current_cycle – last_access_cycle.
- If `age_cycles` ≥ 300 and fragment not referenced by active anchors → mark for purge.
- Purge limit: 1■000 fragments per run to avoid I/O spikes.

## 5. Safety & Rollback

- Purged fragments archived to `crystal_vault/` for 200 cycles.
- Vault uses append■only log; can restore via `mcd_restore(fragment_id)`.
- Defrag pauses if vault size > 2× active crystal size until manual prune.

## 6. Metrics Emitted

- `mcd_dedup_count` – merged duplicates this run
- `mcd_purge_count` – fragments purged
- `mcd_vault_size_mb` – size of rollback vault

## 7. Integration Steps

1. `pip install gm48-mcd`
2. Add `from gm48.mcd import schedule_defrag` in mesh init.
3. Optionally expose metrics to Dashboard via `/mcd` endpoint.

## 8. Quick■Reference JSON


```
{ "mcd": {
  "interval_cycles": 50,
  "age_purge": 300,
  "purge_batch_max": 1000,
  "vault_cycles": 200 } }
```