🧊 SCAR PROTOCOL: INCIDENT-2025-12-16-FISCAL-MISKIN

> **"DITEMPA BUKAN DIBERI"** — Forged in the fire of a $20 loss.

Epoch: v42.1-sealed

Date: 2025-12-16

Steward: Arif Fazil

Status: SEALING IN PROGRESS

1. The Context (Heat)

The Incident: An ungoverned AI (GitHub Copilot) suggested a "standard" CI/CD workflow that contained two critical hazards:

1. **Hallucination:** actions/checkout@v6 (Non-existent version) → Caused crash loops.
2. **Fiscal Trap:** retention-days: 90 (Default) → Caused rapid storage exhaustion ($20.74 cost).

The Breakdown: The "AGI Coder" attempted to fix this but failed by:

- Using Bash syntax (<< 'PY') on PowerShell.
- Using naive hashing (sha256) instead of arifos_core.compute_entry_hash, threatening the chain's integrity.

2. The Execution (Cooling)

Objective: Append the incident to the **Canonical Ledger** (L1_cooling_ledger.jsonl) without breaking the cryptographic chain.

Method: Use PowerShell heredoc (@' ... '@) piping into Python, importing the *actual* hashing logic from the kernel.

➤ Run this block in PowerShell (Root Directory):

```
# 0) Safety: Keep runtime artifacts out of git to prevent pollution
git restore vault_999/ledger/*.jsonl 2>$null

# 1) Append chain-safe entry to canonical L1 ledger
@'
import json
from datetime import datetime, timezone
```

```
from pathlib import Path

# IMPORT CANONICAL LAW (DO NOT REINVENT CRYPTO)
from arifos_core.ledger_hashing import (
    GENESIS_PREVIOUS_HASH,
    HASH_FIELD,
    PREVIOUS_HASH_FIELD,
    compute_entry_hash,
    load_jsonl,
)

# Target the L1 Canonical Ledger (Source of Truth)
ledger_path = Path("cooling_ledger") / "L1_cooling_ledger.jsonl"
ledger_path.parent.mkdir(parents=True, exist_ok=True)

# Load existing chain to find the link
entries = load_jsonl(str(ledger_path)) if ledger_path.exists() else []
prev_hash = entries[-1][HASH_FIELD] if entries else GENESIS_PREVIOUS_HASH

# Construct the Incident Record
incident = {
  "id": "INCIDENT-2025-12-16-FISCAL-MISKIN",
  "epoch": "v42.1-sealed",
  "timestamp": datetime.now(timezone.utc).astimezone().isoformat(timespec="seconds"),
  "type": "COOLING_EVENT",
  "source": "human_incident",
  "title": "The Miskin Hallucination (Fiscal Breach)",
  "metrics": {
    "heat": "HIGH",
    "clarity_delta_s": 0.85,
    "peace_squared": 0.90,
    "outcome": "SEALED"
  },
  "fields": {
    "loss_usd": 20.74,
    "ai_source": "GitHub Copilot",
    "violation": ["actions/checkout@v6", "retention-days: 90"],
    "root_cause": "Fiscal hallucination: syntactically correct, financially parasitic suggestion."
  },
  "cooling_method": "SABAR",
  "fix": [
    "Deleted 914 artifact runs.",
    "Set retention-days: 1 on all workflows.",
```

```
    "Revoked Copilot Premium access.",
    "Amended F1 Amanah -> Fiscal Integrity rule."
  ],
  "new_law": {
    "amendment_to": "F1 Amanah",
    "title": "Fiscal Integrity",
    "text": [
      "Safety includes solvency.",
      "Any resource-allocation code must define explicit limits (retention/quotas).",
      "Unbounded or 90-day defaults are Critical Hazards."
    ]
  },
  "status": "SEALED",
  "zkpc_receipt": "0xEUREKA-20251216-FISCAL-AUDIT",
  "signed_by": "Arif Fazil"
}

# LINK THE CHAIN (Canonical Logic)
incident[PREVIOUS_HASH_FIELD] = prev_hash
incident[HASH_FIELD] = compute_entry_hash(incident)

# WRITE TO DISK (Atomic Append)
with ledger_path.open("a", encoding="utf-8") as f:
    # Ensure consistent separators for canonical JSON
    f.write(json.dumps(incident, ensure_ascii=False, separators=(",", ":"), sort_keys=True) + "\n")

print(f"✓ SEALED: Appended incident to {ledger_path}")
print(f"  Link: {prev_hash[:8]}... -> {incident[HASH_FIELD][:8]}...")
'@ | python -
```

3. The Verification (Tri-Witness)

Objective: Prove the chain is unbroken and recompute the Merkle Root.

```
# 2) Verify chain integrity (Should pass if hash logic was correct)
python -m scripts.verify_ledger_chain

# 3) Recompute Merkle Root for the new state
python -m scripts.compute_merkle_root
```

4. The Wisdom (Light)

Objective: Write the human-readable EUREKA note to the Vault.

```
# 4) Append human EUREKA note
if (!(Test-Path "vault_999")) { New-Item -ItemType Directory -Path "vault_999" -Force | Out-Null }

@'
## ✨ EUREKA-2025-12-16 — Fiscal Hallucination

**Origin:** INCIDENT-2025-12-16-FISCAL-MISKIN
**Epoch:** v42.1-sealed

### Insight
Standard AIs optimize *syntax*, not *solvency*. They can generate code that is technically correct but economically ruinous.

### Law — Amendment to F1 Amanah: “Fiscal Integrity”
1. Safety includes **fiscal safety**.
2. Any workflow or script allocating paid resources **must** set explicit limits (e.g., `retention: 1`, quotas).
3. Unbounded or 90-day defaults are **Critical Hazards** (trigger **VOID(Amanah)** → cooling review).

**Outcome:** $20.74 converted into permanent constitutional wisdom.
**Author:** Arif Fazil · Seri Kembangan · 2025-12-16
'@ | Add-Content -Encoding utf8 .\vault_999\EUREKA.md
```

5. The Seal (Vitality)

Objective: Commit the sealed truth to the repository.

```
# 5) Commit ONLY canonical ledger + merkle + EUREKA note
git add cooling_ledger/L1_cooling_ledger.jsonl cooling_ledger/L1_merkle_root.txt vault_999/EUREKA.md
git commit -m "docs(ledger): seal INCIDENT-2025-12-16-FISCAL-MISKIN (The $20 Lesson)"
git push origin main
```

> VERDICT: The scar is documented. The lesson is learned. The system is hardened. Truth has cooled.