M6 系統 M3 邏輯與工程規格書 (Engineering Specification)

版本：v1.0
適用對象：AGI 架構師、DLT 工程師、系統安全專家、經濟模型設計師
密級：Core Infrastructure / Open Standard

1. 核心公理定義 (Axiomatic Definitions)

本節旨在消除對 $\mathbf{V}_{\text{Work}}$ 的誤解，明確定義其與 PoW (Proof of Work) 及 GDP 的本質區別。

1.1 $\mathbf{V}_{\text{Work}}$ (結構化工作單位) 的物理定義

$\mathbf{V}_{\text{Work}}$ 並非單純的「算力消耗」或「法幣產值」，它是一個**「淨熵減向量」(Net Entropy-Reduction Vector)**。

- 數學定義：
  $$\mathbf{V}_{\text{Work}} = \int_{t_0}^{t_1} (\vec{E}_{in} \cdot \eta_{sys}) - \vec{X}_{ext} \, dt$$
  - $\vec{E}_{in}$：投入的有效能量或勞動 (Input Energy/Labor)。
  - η_{sys}：系統轉換效率係數 (System Efficiency, λ Base)。
  - $\vec{X}_{ext}$：負外部性懲罰向量 (Negative Externalities: 污染、社會危害)。
- 工程實作區別：
  - **≠ Bitcoin PoW**：PoW 是無意義的哈希碰撞 (Hash Collision)；$\mathbf{V}_{\text{Work}}$ 是有意義的物理/資訊結構優化 (如：蛋白質折疊計算、高效能代碼撰寫、房屋建設)。
  - **≠ GDP**：GDP 包含災難重建和污染產值；$\mathbf{V}_{\text{Work}}$ 透過 $\vec{X}_{ext}$ 強制扣除破壞性價值 (VNNV)。

1.2 M3 結構確定性 (Structural Certainty)

系統中的「確定性」並非指「預測未來」，而是指**「邏輯執行的不可篡改性」**。

- M3 邏輯鎖：所有的價值轉移 (Transaction) 必須滿足：
  $$\text{Verify}(\text{Input}) \land \text{Verify}(\text{Logic}) \rightarrow \text{Output}$$
  - 若輸入數據 (如 API 數據) 存在衝突，系統進入 **Fail-Safe** 模式 **(Coin A** 鎖定保護**)**，而非錯誤執行。

2. 系統架構 (System Architecture)

採用「邊緣計算 + 鏈上驗證」**(Edge-Compute, Chain-Verify)** 的混合架構，以解決隱私與效率問題。

2.1 硬體層：M6-SE 載體 (Client Side)

- 組件：Secure Element (SE) 晶片，符合 EAL6+ 安全標準。
- 本地邏輯 **(Local Logic)**：
  - **Logic Switcher** 實例化：交易路由 (Route A/B) 在晶片本地完成，而非雲端。
  - 隱私保護：使用 **ZK-SNARKs (**零知識證明**)** 生成交易憑證。
    - 輸出：「此交易符合 P_{bento} 規則，金額 50 Coin A」的加密證明。
    - 隱藏：具體買了什麼 (如：牛奶、雞蛋)。解決「老大哥監控」的隱私恐慌。

2.2 數據層：多模態預言機網絡 (Multi-Modal Oracle Network)

為解決 **"Oracle Problem" (**數據源造假**)**，系統不依賴單一 AI。

- 共識機制：**Proof of Audit (PoA)**
  - 針對同一企業的 VNPV 審計，隨機調用 3-5 個異構 AGI 模型 (如 Claude-class, Gemini-class, Open-Source Llama-class)。
  - 差異計算：若模型輸出差異 $\Delta > 5\%$，觸發 M2 人工介入或更高階審計。
  - 數據源錨定：強制交叉比對 物理數據 (IoT 水電表) 與 財務數據 (稅務 API)。物理守恆定律 **(M3)** 是最終檢驗標準。

3. 核心協議規格 (Core Protocols)

3.1 邏輯切換器協議 (LSP - Logic Switcher Protocol)

這是一組運行在 SE 晶片與 DLT 節點上的智能合約標準。

```
class LogicSwitcher:
  def execute_transaction(self, user_id, items, merchant_lambda):
    # 1. 初始化交易池
    pool_A = 0  # Survival
    pool_B = 0  # Growth

    # 2. 本地分類 (Local Classification within SE)
    for item in items:
      if item.id in self.P_bento_List:
        # M3 確定性：必需品強制 1:1
        pool_A += item.price_fiat
      else:
        # M6 治理：非必需品應用 Lambda 乘數
        # Lambda 越高，Coin B 支付越少 (激勵)
        effective_price = item.price_fiat / merchant_lambda
        pool_B += effective_price

    # 3. 餘額檢查與 ZK 證明生成
    if user.balance_A >= pool_A and user.balance_B >= pool_B:
      proof = generate_zk_proof(user_id, pool_A, pool_B)
      return self.submit_to_chain(proof)
    else:
```

raise InsufficientFundsError()

3.2 P_{bento} 動態錨定算法 (DAA - Dynamic Anchor Algorithm)

如何防止 P_{bento} 被人為操縱？

- 去極端值平均算法 **(Trimmed Mean w/ M3 Weighting)**:
  - 輸入：全國 10,000+ 個銷售點 (POS) 的即時成交數據。
  - 邏輯：剔除最高 5% 和最低 5% 的價格（排除惡意刷單）。
  - **M2** 緩衝：設定 $\Delta_{max} = 10\%$ (每日最大漲跌幅限制)，防止閃崩或惡意攻擊導致生存金劇烈波動。

4. 容錯與爭議解決 (Fault Tolerance & Resolution)

工程師最關心的問題：「如果 **AI** 判錯了怎麼辦？」

4.1 樂觀審計與挑戰期 (Optimistic Audit with Challenge Period)

- 對於 Coin B 的 VNPV 鑄幣，採用 樂觀 **rollup** 機制。
- **T+3** 挑戰期：初步審計通過後，Coin B 進入「鎖定狀態」3 天。
- 挑戰者激勵：任何第三方 (驗證節點、競爭對手) 若能提供數學證明 (如物理數據不守恆) 證明該 VNPV 造假，可獲得被挑戰資產的 **50%** 作為獎金 (剩餘 50% 銷毀)。
- 意義：讓「找 AI 的錯誤」成為有利可圖的生意，建立去中心化的監督網。

4.2 系統性回滾 (Systemic Rollback)

- 若 M2 監測到 P_{bento} 算法出現系統性偏差 (如：黑客攻擊導致米價顯示為 0)，系統觸發 **M3** 熔斷機制。
- 安全模式：Coin A 交易暫時鎖定為 **T-1** 日的快照價格，保障生存交易不中斷，直到補丁上線。

5. 給開發者的實作指引 (Implementation Guide)

1. 不要發明新密碼學：使用標準的 ECC (secp256r1) 和 SHA-256。
2. **API** 隔離：政府 API 數據只能進入 Oracle 節點，不能直接寫入鏈上。鏈上只存儲 Oracle 簽名後的 Hash。
3. 開源要求：Logic Switcher 的分類邏輯和 P_{bento} 算法必須 100% 開源，接受公眾代碼審計。這是建立信任的唯一途徑。

文件結束 **(EOF)**