# Formal Logical/Cryptographic Condition and corresponding law

| Law ID | Law Name | Formal Logical/Cryptographic Condition | Notes on the Formula |
|---|---|---|---|
| **LAW ZERO** | **PRIMACY OF HUMAN SAFETY** | $\neg (\text{Harm}) \text{ UNLESS } (S_{\text{Authority}}(\text{Lawful}) \land \text{DueProcess})$ | Harm is forbidden unless justified by a signed, provable legal authority. |
| **LAW 1** | **VERIFIED INPUT ONLY (NIPS)** | $\text{Execute} \iff (\text{SchemaValid}(D) \land \text{Verify}(S_{\text{Source}}(H(D))))$ | Execution proceeds only if the input is valid and carries a verifiable, signed attestation from its source. |
| **LAW 2** | **EXECUTION RECEIPT IMMUTABILITY (HARMONEE)** | $\text{Receipt} = H(R_{\text{ID}} \| H(D) \| H(O) \| \text{Actor}_{\text{ID}} \| \text{Timesta$ | Every execution creates a unique, immutable hash (the receipt) binding all essential parameters. |
| **LAW 3** | **GLOBAL AUDIT CONSISTENCY (REVELATION)** | $\forall n: L_n = H(L_{n-1} \| \text{Receipt}_n$ | The ledger state ($L_n$) is the cryptographic hash of the previous state and the current receipt, ensuring historical integrity. |

| | | | |
|---|---|---|---|
| **LAW 4** | **EQUAL EXECUTION REQ.** | $\text{IF } (D_1 \equiv D_2) \land (R(D_1) \neq R(D_2)) \implies \text{MUST Produce } J \text{ where } S(H(J))$ | If inputs are identical but outputs differ, a signed cryptographic **justification ($J$)** must be generated. |
| **LAW 5** | **PUBLIC VERIFIABILITY** | $\forall \text{Verifier } V: V(\text{Receipt}, \text{PublicKeys}) = \text{TRUE}$ | Any third party can independently verify the receipt's integrity using only public, non-secret materials. |
| **LAW 6** | **CORRECTNESS BEFORE IMMUTABILITY** | $H(\text{LegalityProof}) \implies \text{WriteTo}(L)$ | The system must prove legality/correctness ($P$) before the immutable ledger ($L$) can be updated (the **"Prove THEN Record"** principle). |
| **LAW 7** | **FORBIDDEN SILENT OUTPUTS** | $\neg (\text{Decision} \land \neg \text{Logged}) \implies \text{HALT} \land \text{LogForensics}$ | If a decision is made but not logged, the system must immediately stop and capture forensic data. |

| | | | |
|---|---|---|---|
| **LAW 8** | **REDUNDANT AUTHORITY** | $\text{State}_1 \equiv \text{State}_2 \equiv \text{State}_3$ $\text{ IF } \neg(\text{Consensus}) \implies \text{HALT}$ | The state of at least three independent integrity anchors must be identical. Failure to agree results in a system halt. |
| **LAW 9** | **REALITY CONSISTENCY RULE** | $\text{OutputClaim}(O) \iff \text{Verify}(S_{\text{Oracle}}(H(\text{RealityMatch})))$ | The success claim ($O$) must be validated by a signed cryptographic report from an external, attested oracle. |
| **LAW 10** | **MACHINE ACCOUNTABILITY** | $R \subset H(\text{LegalCode}_{\text{ID}})$ | The executed rule ($R$) must be provably derived from (a subset of) a verifiable, hashed version of legal code. |
| **LAW 11** | **HUMAN OVERRIDE SUPREMACY** | $\text{Override} \implies \text{Logged}(S_{\text{Human}}(\text{OverrideAction}))$ | Any human override must be logged and carry a non-anonymous, permanent digital signature. |

| | | | |
|---|---|---|---|
| **LAW 12** | **MODEL AND DATA PROVENANCE** | $\text{Execute} \iff \text{Verify}(S_{\text{Dev}}(\text{ModelHash} \| \text{DataHash} \| \text{ProcHash}$ | Execution requires a verifiable signature attesting to the integrity of the Model, its Training Data, and the Training Process. |