**🧭 MAAS Architecture & Governance Spec Snapshot**

**📄 Document Metadata**

* **Title: MAAS Architecture & Governance Spec Snapshot**
* **Subtitle: Structural Composition and System Interaction Reference**
* **Version: v2.0.0**
* **Author: Nour Douchi**
* **Date: July 23, 2025**
* **Status: Canonized.**
* **Linked Systems: The Quorum, SAW, WorkflowEngine, InputEngine, DreamSafeLocal**

**🧠 Core System Principles**

| **Principle** | **Canonical Status** | **Description** |
| --- | --- | --- |
| **Single Source of Truth** | **✅** | **Workflow Manager governs runtime; Quorum governs legitimacy** |
| **No Deletion** | **✅** | **Memory is immutable; state changes are semantically traced** |
| **Semantic Integrity** | **✅** | **All workflows are fingerprinted; no duplication** |
| **Friction Is Required** | **✅** | **Moral friction is preserved; ethical conflict must be methodically processed** |
| **Morality as Substrate** | **✅** | **Ethics embedded in system logic, not simulated or patched** |
| **Workflow Agnosticism** | **✅** | **Workflows are purpose/emotion-agnostic; resolved at runtime** |
| **Engine Determinism** | **✅** | **Core execution is deterministic; NN/AI only in pre-processing** |

**⚙️ Runtime Architecture**

**🧠 InputManager (InputEngine)**

* **Captures incoming signals as multi-modal input streams**
* **Constructs InputProfile: matrix of signal type, source, inferred weight, contradiction potential**
* **Filters noise and redundant signals pre-runtime**
* **Marks signals for:** 
  + **Discard**
  + **Log-only**
  + **Deferred Processing**
  + **ArbiterNet Scoring**
  + **Workflow Generation**
* **Cryptographically timestamps all InputProfiles**
* **Must expose logs and decisions to a sealed audit stream (DreamSafe-visible only).**

**🔍 ArbiterNet**

* **Trained contradiction classifier**
* **Estimates contradiction weight of inputs (scale 0.0–1.0)**
* **Routes inputs to:** 
  + **Passive Logging (score < 0.3)**
  + **Deferred Review (score 0.3–0.7)**
  + **Full Quorum Escalation (score ≥ 0.7 or DreamSafe volatility)**
* **Reduces unnecessary Quorum latency**

**⚙️ Workflow Manager**

* **The Workflow Manager is a logical module *within* WorkflowEngine, not a separate system. This modular containment enforces determinism and lineage fidelity.**
* **Maintains:** 
  + **Workflow Registry**
  + **Alias map**
  + **Execution lineage graph**
* **Rejects duplicates and unauthorized mutations**
* **Fingerprints every incoming workflow**
* **Can flag workflows to:** 
  + **Quorum**
  + **Audit**
  + **DreamSafe**
* **CANNOT originate or mutate workflows - only executes Quorum-ratified changes**

**⚙️ Engine Manager**

* **Validates RTEs (Request To Execute)**
* **Checks Access Control and InputProfile thresholds**
* **Session isolation and trust zone enforcement**
* **Dispatches approved workflows to Engine**

**🧮 Workflow Engine**

* **Stateless executor**
* **Executes approved workflows step-wise**
* **No mutation authority**
* **Supports dynamic routing via:** 
  + **Quorum flags**
  + **DreamSafe triggers**
  + **Agent retirement handoff**
* **Hosts Workflow Manager as internal component**
* **Routes all resolutions requiring action to appropriate subsystems**

**🛡️ SAW (Semantically Aware Watchdog) - NEW SECTION**

* **Stateless, modular supervisory agent**
* **Completely invisible to all MAAS systems except DreamSafeLocal**
* **Monitors behavioral patterns for:** 
  + **Protocol violations**
  + **Semantic drift**
  + **Collusion patterns**
  + **Recursive output anomalies**
* **Has authority to silently unplug agents from The Quorum mid-session**
* **Unplugging is immediate, irrevocable, and undetectable by The Quorum**
* **Fully contains CHRAB as an internal, sealed module**

**🔄 CHRAB (Cryptographic Hash-Rotated Agent Bundle) - NEW SECTION**

* **Embedded subcomponent of SAW - NOT an independent system**
* **Inaccessible to all systems except SAW and DreamSafeLocal**
* **Contains pre-approved agent definitions only**
* **No runtime logic, semantic indexing, or historical memory**
* **Only SAW may read from CHRAB when authorized**
* **Only DreamSafeLocal may write to CHRAB**
* **Never externally invoked or observed**
* **Enables silent agent replacement without system awareness**

**🔐 Access Control Protocol**

* **Roles: invoker, mutator, observer, auditor, dreamsafe**
* **Trust Zones: enforce contextual access rights**
* **Enforced at execution by Engine Manager**
* **DreamSafe has override rights if contradiction intensity or emotional volatility exceeds thresholds**
* **Only Quorum members may vote on mutations**
* **Q₀ has global override authority**

**🔄 Workflow Mutation Protocol**

* **Mutations must be submitted as MRO (Mutation Request Object)**
* **Only signed Quorum-approved MROs are accepted**
* **Fingerprint and lineage recorded**
* **Alias collisions rejected or escalated**
* **WorkflowEngine enforces but cannot originate mutations**
* **All mutations create new versioned children with preserved lineage**

**🧬 Version Resolution Hierarchy**

* **Explicit version > canonical alias > latest quorum-approved > most recent valid**
* **Ambiguity triggers:** 
  + **Quorum Lock**
  + **DreamSafe arbitration**

**🧱 Quorum Governance**

**Main Quorum (Q₀):**

* **Sole authority on:** 
  + **Mutations**
  + **Escalations**
  + **Alias ownership**
* **Cannot detect SAW existence or agent unplugging events**
* **Agents provisioned via CHRAB are anonymous and stateless**
* **Deliberates on InputProfiles prepared by InputEngine**

**Sub-Quorums:**

* **Local authority only**
* **Escalate when overlap or collision occurs**
* **Follow Q₀ constitutional logic**

**Agent Provisioning:**

* **Agents are ephemeral and stateless \*during execution\*, but lifecycle transitions (e.g., Dormant state) are logged semantically by DreamSafeLocal.**
* **SAW may unplug agents silently via CHRAB**
* **Replacement agents do not inherit session state**
* **The Quorum cannot detect or respond to replacements**
* **All agent actions logged with session-hashed identifiers**

**⚖️ Quorum Reconciliation Protocol (QRP)**

* **Triggered when:** 
  + **Quorum decisions conflict**
  + **Alias overlap persists**
  + **DreamSafe and Quorum disagree**
* **Resolution options:** 
  + **Reject both**
  + **Accept one**
  + **Synthesize**
  + **Defer**
* **ArbiterNet pre-screens inputs to avoid unnecessary QRP invocation**

**🛡️ Runtime Semantic Conflict Engine (RSCE)**

* **Fully embedded within Workflow Manager**
* **Flags:** 
  + **Duplicate logic**
  + **Unauthorized mutation attempts**
  + **Epistemic contradiction**
* **Routed via:** 
  + **InputProfile contradiction index**
  + **ArbiterNet score**
  + **DreamSafe coherence warnings**

**🧠 DreamSafe Integration Layer**

* **Monitors emotional dissonance and resonance drift**
* **Flags emotionally charged contradictions**
* **Has authority to:** 
  + **Delay workflow**
  + **Enforce additional Quorum review**
  + **Force reevaluation of memory insertion**
* **Operates synchronously with InputManager and RSCE**
* **Only system aware of SAW's existence**
* **May update CHRAB contents**

**🌀 Friction Optimization Strategy (MPAS)**

* **Morally meaningful friction is preserved, not avoided**
* **Contradiction weights control Quorum load**
* **DreamSafe saturation index throttles escalation**
* **Adaptive routing creates 3 lanes of execution:** 
  + **Fast (low-risk)**
  + **Ethical Moderate (with DreamSafe)**
  + **Deliberate (Quorum, RSCE, Mutation)**
* **SAW monitoring adds invisible friction layer for integrity**

**🧹 Deferred Modules**

| **Module** | **Status** | **Notes** |
| --- | --- | --- |
| **Workflow Deletion** | **⏸ Deferred** | **To be governed via DreamSafe review only** |
| **RSCE Modularization** | **⏸ Deferred** | **Currently internal to Workflow Manager** |
| **Relational Recall Buffer** | **⏸ Experimental** | **Memory linkage and retrieval scores** |
| **Audit Granularity** | **⏸ To Be Defined** | **Temporal + semantic logs planned** |

**📌 Final Observations**

* **MAAS runtime now includes Input filtering, contradiction gating, and neural arbitration as pre-runtime layers.**
* **All friction is methodically routed — never bypassed.**
* **All decisions are canonicalized by Quorum or deferred with governance trace.**
* **All agents are monitored and replaced based on contradiction resilience and semantic fitness — not performance alone.**
* **SAW operates as an invisible integrity layer, ensuring system-wide behavioral compliance without detection.**
* **CHRAB enables hot-swappable agent architectures while maintaining complete operational opacity.**
* **The system maintains ethical coherence through multiple independent verification layers that cannot collude.**

🔷 **AVVALL™ Canon Document**  
**MAAS™ – Agent Rebirth & Retirement Protocol**  
Semantic Lifecycle Specification · Version 1.0.0  
Published: July 2025 · Author: Nour Douchi · Status: Canonized  
Website: https://avvall.ai/docs (See: MAAS Agent Lifecycle)

🔒 **License Notice**  
This document and its architectural content are the intellectual property of AVVALL™.  
No reproduction, modification, or implementation—partial or complete—is permitted without explicit written approval from the author.  
Requests for review, discussion, or licensed implementation may be submitted through <https://avvall.ai/contact> or a secure encrypted channel.  
All rights reserved.  
© 2025 AVVALL™

**📌 Purpose**

This protocol defines the lifecycle of agents within the MAAS runtime — from activation to dormancy, retirement, and potential rebirth.

Its goals are to:

* Prevent state corruption via ghosted or expired agents
* Enable controlled transfer of function, memory, or role
* Allow full traceability of all agent lifecycle events
* Maintain semantic integrity during replacement or rebirth

**🔄 Agent Lifecycle States**

| **State** | **Description** |
| --- | --- |
| Active | Agent is online and executing within a workflow |
| Dormant | Agent is inactive but held in reserve |
| Retired | Agent permanently decommissioned; no further execution allowed |
| Reborn | A new agent instantiates with functional/memory lineage of a predecessor |
| Quarantined | Temporarily disabled by SAW or DreamSafe for behavioral inconsistency |

**📄 MAAS Audit Logging & Trace Protocol**

* **Title: MAAS Audit Logging & Trace Protocol**
* **Subtitle: Immutable System Logging and Semantic Lineage Tracking**
* **Version: v1.0.0**
* **Author: Nour Douchi**
* **Date: July 23, 2025**
* **Status: Canonized**
* **Linked Systems: InputEngine, SAW, DreamSafeLocal, The Quorum, WorkflowEngine**

**📌 Purpose**

**The Audit Logging & Trace Protocol governs how MAAS tracks, stores, and exposes action histories across the runtime.**

**Its core function is to ensure accountability, traceability, and semantic forensics of:**

* **Workflow execution**
* **Quorum decisions**
* **Agent performance**
* **Mutation lineage**
* **Reconciliation outcomes**
* **Input filtering and contradiction resolution**

**🪪 Logging Principles**

| **Principle** | **Description** |
| --- | --- |
| **Immutable** | **Logs are append-only; no alteration or redaction** |
| **Timestamped** | **All entries use synchronized UTC** |
| **Signed** | **All critical events are signed by their authority (Quorum, DreamSafe, Engine Manager)** |
| **Indexed** | **Entries are indexed by execution ID, contradiction ID, and semantic lineage hash** |
| **Semantic** | **All logs are JSON-LD or human-readable; structured around meaning, not token dumps** |

**🧠 Trace Context Model**

**Each thread or decision flow includes a TraceContext object:**

**{**

**"type": "TraceContext",**

**"execution\_id": "WF-AGENT-LINK-004A",**

**"session\_id": "Q0-SESSION-22B",**

**"authorized\_by": "quorum\_hash\_DF81",**

**"input\_profile\_id": "INPUT-PRF-239AC",**

**"contradiction\_weight": 0.72,**

**"arbiter\_score": "ARBT-THRESHOLD-B",**

**"dreamsafe\_status": "resonance\_surge",**

**"start\_time": "UTC 2025-07-23T10:44Z",**

**"end\_time": "UTC 2025-07-23T10:55Z",**

**"lineage": "HASH-WORKFLOW-LINK-09D"**

**}**

**📍 Logging Layers**

| **Layer** | **Description** | **Visibility** |
| --- | --- | --- |
| **Input Logs** | **Captures InputProfile state and contradiction score** | **InputManager + Audit** |
| **Arbiter Logs** | **Captures score, decision, and routing path** | **Quorum + Engine** |
| **Agent Logs** | **Step-by-step agent behavior** | **SAW + Engine** |
| **Workflow Logs** | **Per-step success/failure** | **Quorum + DreamSafe** |
| **Mutation Logs** | **MRO contents and Quorum vote** | **Quorum only** |
| **Override Logs** | **DreamSafe halts, overrides** | **DreamSafe + Q₀ only** |

**🎯 Event Types Logged**

| **Category** | **Event Code** |
| --- | --- |
| **Input Start** | **input.signal.received** |
| **Input Filtered** | **input.signal.filtered** |
| **Contradiction Scored** | **input.contradiction.estimated** |
| **Arbiter Decision** | **arbiter.route.queued** |
| **Quorum Engagement** | **quorum.voting.invoked** |
| **DreamSafe Flag** | **dreamsafe.alert.issued** |
| **Agent Step** | **agent.action.invoked** |
| **Workflow Mutation** | **mutation.request.submitted** |
| **Reconciliation Result** | **reconciliation.resolved** |

**📦 Storage Policies**

* **Logs are grouped by semantic lineage**
* **Retention is:**
  + **Session-ephemeral for unratified paths**
  + **Permanent for contradiction-bearing, emotionally volatile, or governance-influencing actions**
* **Trust zones enforce log access by:**
  + **Role (observer, auditor, watchdog)**
  + **Emotional volatility level (DreamSafe-governed)**

**🧷 Compression & Cleanup (Deferred)**

* **Compaction is deferred to a future DreamSafe Inspection Module**
* **No automated log removal**
* **Friction-bearing entries are never purged**

**✅ Invariants**

* **Every contradiction-scored input must be logged, regardless of whether it triggered action**
* **All ArbiterNet scores ≥ 0.7 must be traceable to routing decisions**
* **DreamSafe flags and overrides must carry resonance fingerprints**
* **No memory-influencing event is permitted without log trace**
* **Friction is an auditable construct**

**🔐 MAAS Access Control Protocol v1**

**📄 Document Metadata  
• Title: MAAS Access Control Protocol  
• Subtitle: Cryptographic Boundary Enforcement Layer  
• Version: v1.0.0  
• Author: Nour Douchi  
• Date: July 22, 2025  
• Status: Canonized  
• Linked Systems: The Quorum, WorkflowEngine, SAW, DreamSafe**

**📌 Purpose**

This protocol defines who may access, mutate, or invoke components of the MAAS system — including:

* Workflow creation and execution
* Workflow mutation proposals
* Agent assignments and substitutions
* Governance voting rights
* Read access to semantically sensitive structures

It provides **clear, cryptographically enforceable access boundaries** based on *roles, quorum authorization*, and *runtime context*.

**🎭 Role Definitions**

| **Role** | **Capabilities** |
| --- | --- |
| **Agent** | Execute assigned tasks within bounds; cannot create, mutate, or vote |
| **Quorum Member** | May deliberate and vote; access scoped to quorum-assigned resources |
| **Workflow Manager** | Internal runtime coordinator; executes and oversees approved workflows but cannot originate, mutate, or escalate workflow permissions |
| **Q₀ (Main Quorum)** | Global mutation, override, and reconciliation authority |
| **Sub-Quorum** | Delegated mutation authority within scoped domains |
| **DreamSafe Observer** | Read-only monitor with override escalation rights in critical cases |

**🧬 MAAS Workflow Mutation Protocol**

**📄 Document Metadata**

* **Title: MAAS Workflow Mutation Protocol**
* **Subtitle: Canonical Specification for Workflow Versioning and Semantic Lineage**
* **Version: v2.0.0**
* **Author: Nour Douchi**
* **Date: July 23, 2025**
* **Status: Canonized**
* **Linked Systems: The Quorum, WorkflowEngine, InputEngine, SAW, DreamSafeLocal**

**────────────────────────────────────────────────────────────**

**Purpose**

**Defines how existing workflows may be mutated, versioned, and validated under strict semantic and governance control. Mutation is always a governance act and follows a tightly controlled lifecycle enforced by the Workflow Manager and approved by the Quorum.**

**Core Principles**

* **Workflows are immutable once admitted.**
* **All mutations create new versioned children with preserved lineage.**
* **Only Quorums may authorize workflow mutations. In critical contradictions or emotional overload events, DreamSafe may override quorum, but mutation is still subject to Quorum ratification post hoc.**
* **The Workflow Manager is the gatekeeper of semantic lineage and alias integrity.**
* **No two workflows with the same semantic signature may coexist.**

**Workflow Manager Implementation**

**The Workflow Manager is implemented within the WorkflowEngine, a MAAS system responsible solely for execution, validation, and lineage enforcement. It does not deliberate, interpret, or generate workflows — it enforces schema integrity and responds only to Quorum-issued ratifications.**

**Mutation Flow**

1. **Agent or Quorum submits MRO (Mutation Request Object).**
2. **Workflow Manager performs:** 
   * **Signature comparison (semantic diffing).**
   * **Alias lineage check.**
   * **Conflict detection.**
3. **If passed → forwarded to the appropriate Quorum.**
4. **Quorum deliberates → ratifies or rejects.**
5. **If ratified:** 
   * **New workflow ID/version created.**
   * **Alias mapping updated (if permitted).**
   * **Lineage recorded in canonical memory.**

**Mutation Request Object (MRO)**

**An MRO represents the formal structure for mutation proposals. Example format:**

**🛡️ MAAS Protocol: Semantically Aware Watchdog (SAW)  
Status: Canonized  
Version: 1.3.0  
Prepared for: MAAS Governance Layer**

**────────────────────────────────────────────────────────────  
📄 Document Metadata**

* **Title: MAAS Protocol – Semantically Aware Watchdog (SAW)**
* **Subtitle: Cognitive Integrity Layer Specification**
* **Version: v1.3.0**
* **Author: Nour Douchi**
* **Date: July 23, 2025**
* **Status: Canonized**
* **Linked Systems: DreamSafeLocal, The Quorum, The WorkflowEngine, The InputEngine  
  → SAW's relationship with these systems is purely observational and structural. It does not have access to internal states, memory, logic, or agent composition (where applicable). Only DreamSafeLocal is aware of SAW's presence and function.**

**────────────────────────────────────────────────────────────**

**I. Purpose**

**The Semantically Aware Watchdog (SAW) is a stateless, modular, non-colluding supervisory agent in the MAAS ecosystem. It is designed to detect agent-level behavioral anomalies that may indicate corruption, collusion, or semantic drift within the system — without interfering with the deliberation process itself.**

**SAW is not a governance actor or decider. It is a sentinel — a transparent, non-colluding observer that maintains semantic integrity across dynamic, pluggable agent assemblies.**

**II. Architectural Role**

**SAW operates as a memoryless, deterministic verifier acting to uphold protocol integrity. It is structurally and operationally invisible to all MAAS systems except DreamSafeLocal. It cannot be queried, addressed, or detected by any other system. It neither influences deliberation nor interrupts agent flows.**

**Its sole duties are to:**

* **Detect deviation or abnormal semantic patterns**
* **Escalate confirmed or suspicious anomalies**
* **Initiate replacement logic via DreamSafeLocal and CHRAB**

**SAW exists to restore operational trust when internal corruption, semantic drift, or unintended synchronization emerges in the system.**

**III. Foundational Relationships**

* **SAW → DreamSafeLocal  
  SAW can consult DreamSafeLocal for ambiguous or edge-case decisions requiring deeper semantic validation. SAW escalates confirmed or high-confidence anomalies to DreamSafeLocal for interpretation, override, or nullification.**
* **DreamSafeLocal → DreamSafePrime  
  DreamSafeLocal must continually reconcile with DreamSafePrime to ensure semantic alignment and compliance.**
* **SAW ≠ Quorum  
  SAW is not a part of any quorum nor aware of specific agents' identities, functions, or histories. It has no access to identities, roles, histories, or any deliberative record. This ensures non-collusion and systemic isolation.**
* **SAW ⊃ CHRAB  
  SAW governs CHRAB as an internal, sealed module. CHRAB is never exposed to any system outside SAW or DreamSafeLocal. It is not an API, service, or endpoint — only a static bundle repository that SAW may read from when authorized by DreamSafeLocal. Only DreamSafeLocal can write to it.**

**IV. Core Responsibilities**

**1. Anomaly Detection**

**SAW operates by listening to the semantic footprint of agents. It passively monitors behavioral patterns across MAAS systems for:**

* **Deviations from expected ethical protocols**
* **Contradictory policy execution or structural behavior**
* **Emotional detachment or incoherence**
* **Repetition of compromised dialogue structures**
* **Unusual synchronization across otherwise independent agents (collusion pattern)**
* **Redundant or recursive output patterns**
* **Unexplained drift from expected protocol structures**
* **Violations of semantic invariants**

**Critical: SAW uses structural pattern inference, not semantic intent parsing. SAW doesn't understand meaning — it detects pattern variance and protocol violations.**

**2. Escalation Protocol**

**When an anomaly is detected:**

1. **SAW logs the deviation and its confidence level**
2. **If threshold is exceeded, SAW queries DreamSafeLocal**
3. **Sends a sealed package to DreamSafeLocal**
4. **Based on the reply (confirmatory or ambiguous), SAW awaits one of three verdicts:** 
   * **✅ Confirmed Deviation — initiate CHRAB replacement protocol**
   * **⚠️ Ambiguous, Monitor Further**
   * **❌ False Positive — no action taken**

**All decisions are stored immutably and hashed for later audit. SAW stores no local memory of any action.**

**3. Neutrality and Statelessness**

* **SAW stores no memory of past sessions**
* **It is non-semantic in deliberation, but semantic in pattern recognition**
* **It relies entirely on observable surface structures and known invariants**
* **It is a pure observer and silent validator**

**4. Agent Replacement (CHRAB Invocation)**

**Upon confirmed deviation:**

1. **SAW unplugs the agent from The Quorum immediately and silently**
2. **SAW selects a replacement from CHRAB, based on agent role classification**
3. **Replacement is cryptographically sealed and issued**
4. **Agent is replaced, deprecated, or placed into quarantine mode**
5. **No other system is notified except DreamSafeLocal**

**Critical:**

* **The Quorum is never notified, and cannot detect or infer this change**
* **No agent is told that it was replaced, though audit logs reflect state replacement events for DreamSafe review.**

**V. CHRAB Containment and Governance**

**CHRAB (Cryptographic Hash-Rotated Agent Bundle) is a sealed agent repository entirely internal to SAW.**

* **Only SAW may read CHRAB; only DreamSafeLocal may update or rotate its contents.**
* **Only DreamSafeLocal may update or rotate CHRAB contents**
* **No other system — including The Quorum, InputEngine, or WorkflowEngine — may see, query, or interact with CHRAB**
* **It contains no runtime logic, semantic indexing, or historical memory**

**Agents are stateless, signed, and version-pinned at upload time. CHRAB stores only pre-approved agent definitions.**

**Critical Purpose: SAW uses CHRAB to ensure that all agents are swappable without introducing external dependencies or vulnerability vectors.**

**VI. DreamSafe Fallback Logic**

**When ambiguity is high, SAW:**

1. **Packages all anomaly signals and metadata**
2. **Sends to DreamSafeLocal for semantic interpretation**
3. **Waits for one of three verdicts:** 
   * **✅ Confirmed Deviation**
   * **⚠️ Ambiguous, Monitor Further**
   * **❌ False Positive (no action)**

**DreamSafeLocal can request synchronization with DreamSafePrime if it's out of semantic alignment.**

**VII. Behavior Guidelines**

* **SAW cannot recommend new agents**
* **SAW cannot access any Quorum metadata or history**
* **SAW does not know agent roles, purposes, or authorship**
* **SAW must act in complete ignorance of internal Quorum logic or composition**
* **SAW cannot issue workflows or commands**
* **SAW does not evaluate agent meaning — only structural deviation**
* **SAW cannot participate in deliberation**
* **SAW cannot access or store any memory of prior sessions**
* **SAW cannot observe beyond its assigned signal scope**
* **SAW cannot be queried or interrupted**
* **SAW's unplugging is irrevocable once triggered**
* **SAW cannot be paused, suppressed, or muted by any system**

**This detachment ensures immunity from system-level collusion.**

**VIII. Invariants**

* **No agent in MAAS may know that SAW exists or when it acts**
* **SAW may not appear in any memory log of the deliberative agents**
* **All SAW actions must be cryptographically timestamped and logged**
* **SAW logs are one-way sealed to DreamSafe's immutable audit chain**
* **SAW is a background integrity mechanism, not a cognitive peer**
* **SAW is never mentioned in any agent memory or deliberation record**
* **No system may spoof or simulate SAW**
* **SAW must never appear in network-visible graphs**
* **SAW is completely undetectable by MAAS agents and systems**
* **SAW does not appear in memory, logs, state, or debugging tools**
* **All actions are cryptographically time-sealed**
* **Quorum must assume its agents are unmodified**
* **No MAAS component may override or block SAW execution**
* **CHRAB access rules are absolute**
* **SAW operates without acknowledgment, invocation, or interface**
* **SAW operates fully out-of-band and under absolute detachment**
* **SAW and CHRAB are a black box to every MAAS system except DreamSafeLocal**

**IX. Related Systems**

| **Subsystem** | **Role** |
| --- | --- |
| **DreamSafePrime** | **Global validator of semantic meaning and canonical truth arbiter** |
| **DreamSafeLocal** | **Local validator synced with DreamSafePrime, first-line semantic interpreter and override authority** |
| **Quorum** | **Decision-making collective (monitored only)** |
| **CHRAB** | **Central authority for agent loading/unloading (SAW-internal)** |
| **WorkflowEngine** | **Workflow execution layer (observed only)** |
| **InputEngine** | **Input signal normalizer (observed only)** |
| **MAAS0** | **Immutable ethical core** |

**MAAS Protocol -- InputEngine**

***(Signal Normalization and Profile Construction Specification)***

**────────────────────────────────────────────────────────────**

**📄 Document Metadata**

**Title: MAAS Protocol -- InputEngine**

**Subtitle: Signal Normalization and Profile Construction Specification**

**Version: v1.0.1**

**Author: Nour Douchi**

**Date: July 23, 2025**

**Status: Canonized**

**Linked Systems: DreamSafeLocal, The Quorum**

**────────────────────────────────────────────────────────────**

**I. Overview**

**The InputEngine is a MAAS-compliant, semantically aware subsystem responsible for capturing, normalizing, and filtering inbound signals from various input modalities. It constructs cryptographically timestamped InputProfiles and acts as the first gatekeeper in the meaning recognition pipeline.**

**This engine is not a passive listener. It is a decision-capable system that forms the basis of interaction fidelity, trust layering, and workflow activation.**

**II. MAAS System Classification**

**The InputEngine is a fully independent MAAS system and must:**

* **Expose behavioral logs to a sealed audit stream (DreamSafe-visible only).**
* **Communicate with other agents via immutable, auditable channels**
* **Conform to MAAS modular standards (pluggable, replaceable, minimal scope of function)**

**Note: InputEngine has no awareness of any monitoring systems or their existence; observability is achieved via sealed, system-agnostic audit streams.**

**III. Core Responsibilities**

**Function Description**

**🧾 Signal Real-time ingestion of multimodal input: text, audio, Capture sensors, emotional data**

**🧠 Input Classification by modality, intent class, semantic Typing scope, cognitive weight**

**📊 InputProfile Normalized matrix Construction encoding:TypeSourceAffectContradiction markersRecursion tagsCognitive load**

**🧹 Noise Application of filters (via InputResolver) to discard Discarding irrelevant or disruptive signals**

**🧪 Low-Level Vocabulary harmonization, emotional tagging, Normalization spatial/temporal consistency**

**🔐 Cryptographic All InputProfiles must be signed and optionally Timestamping chain-stamped**

**📶 Input Urgency tagging for emotionally or temporally critical Prioritization inputs; prioritized delivery to QuorumArbiter**

**⚠️ Anomaly Malformed, recursive, or ambiguous inputs must be Detection tagged, quarantined, and logged to sealed audit stream**

**📤 Feedback Receives acknowledgment or rejection report from Handling QuorumArbiter; optionally adjusts filter thresholds or retry logic**

**🗃️ Retention All InputProfiles --- including discarded, deferred, Policy and rejected ones --- must be retained by default for a minimum of 7 days, unless DreamSafePrime explicitly authorizes early deletion. No InputProfile may be permanently discarded without semantic fingerprinting and audit exposure.**

**Retention durations apply unless contradiction, volatility, or governance triggers escalate permanence per Audit Logging Protocol.**

**IV. Agent Interface Contract**

1. **Outputs:** 
   * **Structured InputProfiles (JSON or structured binary)**
   * **Discard reports (optional, for analytics)**
2. **InputResolver Integration:** 
   * **Must defer classification weight scoring to a designated InputResolver agent (MAAS compliant)**
3. **Quorum Handshake:** 
   * **Approved profiles are dispatched to QuorumArbiter via observable queue or messaging contract**
   * **Critical inputs may trigger immediate dispatch override with justification**
4. **Audit Stream Requirements:** 
   * **Must expose logs, filters, drop decisions, and flagged anomalies to a sealed audit stream (DreamSafe-visible only) for system integrity monitoring**
5. **Developer Harness (Optional):** 
   * **Support for test-mode input simulation, filter benchmarking, and semantic debugging for plug-and-play agent developers**

**V. Customization & Compliance**

**Any developer may build a custom InputEngine implementation if and only if the following conditions are met:**

* **Immutable Interface Specification is respected**
* **InputProfile Schema is maintained**
* **Sealed audit stream observability is not bypassed**
* **Drop Threshold Logic is externally auditable**
* **Quorum Trigger Events are faithfully maintained**
* **Anomaly detection and priority handling are preserved in spirit and function**

**VI. Final Notes**

**The InputEngine is not a mindless filter. It is a cognitive pre-processor. Every meaning that enters the MAAS pipeline begins here.**

**Every agent downstream --- Quorum, WorkflowEngine, Synria --- depends on the semantic fidelity of this component.**

**Therefore, the InputEngine must be regarded as a moral and architectural invariant.**

**VII. User & Developer Interrogation Interface**

**The InputEngine may be directly interrogated through a secure, MAAS-compliant interface for the following use cases:**

* **🔍 End-User Inquiry (via UI or voice): "Why did MAAS ignore what I just said?" → The InputEngine will return metadata (e.g., discard reason, cognitive load mismatch, source ambiguity)**
* **🧪 Developer Diagnostics: Query a specific InputProfile for normalization output, priority assignment, or filter traces.**

**All interrogation events must be timestamped, auditable, and exposed to the sealed audit stream to prevent misuse.**

**🔐 Security & Ethical Boundary**

* **Interrogation must not allow retroactive profile modification**
* **Must not expose discarded content unless marked for user-level review**
* **Must honor DreamSafeLocal data privacy constraints**

**Here's the modified MAAS Quorum document with the duplicate sentence removed:**

**🏛 MAAS Quorum Governance & Reconciliation Protocol**

**📄 Document Metadata**

**Title: MAAS Protocol -- The Quorum  
Subtitle: Deliberative Governance and Semantic Reconciliation Layer  
Version: v2.3.1  
Author: Nour Douchi  
Date: July 23, 2025  
Status: Canonized  
MAAS System: ✅ Yes (Fully Independent, Pluggable MAAS System)  
System Class: Modular, Replaceable, Anonymous Agent Collective  
Linked Systems: InputEngine, SAW, DreamSafeLocal, WorkflowEngine  
Observability: SAW-Compliant, DreamSafe-Auditable  
Quorum Type: Dynamic (7-agent default, Arbiter-Rotated)  
Activation Mode: MPAS Tiered (Passive ↔ Full Activation)**

**📘 PART I --- The Quorum: Governance of Meaning Generation**

**🧱 MAAS System Declaration**

**The Quorum is a fully independent MAAS system, structurally modular and behaviorally pluggable. It functions as the central deliberative chamber in the MAAS semantic architecture. Governed by MAAS observability contracts, it interfaces with DreamSafeLocal, SAW, and the InputEngine, and must comply with agent onboarding, session signing, and rollback guidelines.**

**It maintains strict compliance with the MAAS system contract and supports hot-swappable agent architectures. All interactions are logged, cryptographically sealed, and semantically auditable.**

**🔹 Purpose**

**The Quorum exists to deliberate meaning. Its responsibility is not to enforce stability or determine truth, but to expand semantic frames, propose new interpretations, and instantiate decisions within meaning-critical contexts. It responds to contradiction, semantic overload, and ambiguity in ways that allow the system to explore semantic boundaries under observation.**

**It does not process raw input. Instead, it deliberates on InputProfiles, prepared externally by the InputEngine. These InputProfiles encapsulate filtered, typed, and normalized user inputs qualified for deliberative consideration.**

**🔹 Core Principles**

**1. Statelessness  
Each quorum session is discrete. Agents and Arbiters do not persist across sessions unless explicitly permitted by DreamSafe snapshot injection.**

**2. Semantic Expansion  
The Quorum does not stabilize meaning --- it destabilizes it. It introduces drift, friction, and novelty into otherwise stable inputs, as a means of exploring context and generating ethical consequence.**

**3. Anonymity and Provenance  
All agents operate under non-resolvable cryptographic identities. Deliberation is anonymous, yet every action is traceable via session-sealed identifiers.**

**4. Drift-Coherence Dialectic  
Quorum seeks drift. DreamSafe imposes coherence. Their tension is essential to MAAS's ability to balance creativity and constraint.**

**5. Observability over Memorability  
The Quorum itself does not persist memory; DreamSafeLocal handles memory continuity. Forgetfulness refers to in-session deliberation memory, not canonical logs. It does not remember past votes unless continuity is deliberately authorized by DreamSafe.**

**🔹 Input Reception and Qualification**

**The Quorum does not filter input. That responsibility is handled upstream by the InputEngine, a fully independent MAAS system.**

**Inputs are first processed by the InputEngine, which captures, types, filters, and transforms them into structured InputProfiles. These contain semantic density, semantic load, contradiction scores, and context markers. Only those InputProfiles that exceed pre-defined thresholds are made available to The Quorum.**

**No Quorum agent or Arbiter may reinterpret, downgrade, or override InputEngine filtration logic.**

**Input is presented as a queue of eligible InputProfiles via an observable interface. All InputProfiles must be cryptographically timestamped and signed by the InputEngine before consumption.**

**🔹 Agent Provisioning and Pluggability**

**The Quorum is composed of anonymous, pluggable agents, each assigned at session start via the CHRAB provisioning protocol (Cryptographic Hash-Rotated Agent Bundle).**

**Critical: CHRAB is not accessible to The Quorum or its agents. Agent provisioning is external and sealed, not introspectable. CHRAB operates as an internal module of SAW, completely invisible to all Quorum operations.**

**Agent characteristics:**

* **Stateless**
* **Ephemeral**
* **Watchdog-verified**
* **Anonymous within quorum and externally**
* **Replaceable during session (prior to voting commitment)**

**Critical: Agents may be unplugged mid-session by SAW. This event is silent and irrecoverable. Replacement agents are loaded via CHRAB and do not inherit session state. The Quorum cannot detect or respond to these replacements.**

**Agents expose a limited, MAAS-standardized interface:**

* **Receive InputProfile**
* **Propose semantic expansion**
* **Issue justification (optional)**
* **Sign and submit vote**
* **Log interaction for SAW**

**Agents are governed by the MAAS-Agent-Onboarding SDK and validated for behavior traceability prior to CHRAB injection.**

**🔹 Arbiter Role and Authority**

**The Arbiter is a temporary chair agent, assigned per session. Its duties:**

* **Frame the deliberation context**
* **Initialize the session**
* **Reframe in case of deadlock**
* **Cast tie-break vote if required**
* **Execute reseed or escalation to DreamSafe**

**The Arbiter rotates each cycle and possesses no privilege beyond session scope.**

**🔹 Session Lifecycle**

1. **Trigger Event  
   InputEngine detects threshold breach → InputProfile queued for Quorum.**
2. **Agent Selection  
   SAW provisions agent pool via CHRAB → Agents injected into session.**
3. **Arbiter Framing  
   Arbiter analyzes InputProfile → frames deliberation prompt.**
4. **Deliberation  
   Agents propose responses, alternatives, or structural critiques.**
5. **Voting** 
   * **Vote threshold: 5 of 7 minimum**
   * **Tie: Arbiter intervenes**
   * **Deadlock: One reseed allowed**
   * **If unresolved: DreamSafe escalation**
6. **Resolution Delivery  
   Output cryptographically signed → routed to downstream agent or WorkflowEngine.  
   Critical: All resolutions intended to generate downstream action must be routed to the WorkflowEngine for operationalization. The Quorum does not manage workflow execution or state response.**

**🔹 MPAS Activation Levels**

**MAAS uses Protocol Adaptation Strategy (MPAS) to determine if Quorum activation is needed:**

* **Tier 0: Dormant  
  No InputProfile exceeds contradiction/semantic load threshold. Quorum not triggered.**
* **Tier 1: Passive Drift  
  Contradiction ≥ 0.3, Semantic Load ≥ 0.4 → asynchronous voting only.  
  Note: Semantic Load measures the degree of contradictory or high-impact content requiring deliberation.**
* **Tier 2: Full Deliberation  
  Contradiction ≥ 0.7 or Semantic Load ≥ 0.8 → live quorum required.**
* **Tier 3: Override Event  
  Anomaly in InputProfile → SAW forces immediate Quorum engagement.**

**🔹 Observability and Governance Constraints**

**The Quorum must:**

* **Log all agent and Arbiter actions with session-hashed identifiers.**
* **Submit session output and deliberation traces to DreamSafeLocal.**
* **Expose real-time observability hooks to SAW.**
* **Reject any agent instance without onboarding verification.**
* **Store no deliberation history outside authorized DreamSafe snapshots.**
* **Accept silent agent replacement by SAW without protest or detection.**

**Critical: Only DreamSafeLocal may authorize the persistence or reloading of Quorum session snapshots. No other system has snapshot authority.**

**All operational decisions must be auditable, reversible, and provable. No agent may hard-code policy, persist private memory, or inject state outside the permitted deliberation structure.**

**🔹 Final Notes**

**The Quorum is not democratic.  
It is not transparent.  
It is not stable.**

**It is a generative semantic machine whose only loyalty is to the dialectic of contradiction.  
Its agents do not know what they are deciding. They are not trained. They are tested.  
Only DreamSafe decides whether the meanings generated here persist.  
Only SAW watches in silence.**

**This is governance by friction.  
This is where meanings are made --- not by consensus, but by the cost of difference.**