

SAMIRAC PARTNERS LLC

SAQ™ Execution Contract Specification

A standard contract for governed AI execution at the execution boundary

PUBLICATION	Public Draft v0.2
ISSUE DATE	August 13, 2026
DOCUMENT ID	SAQ-ECS-0.2-20260813
SUPERSEDES	Public Draft v0.1 — July 23, 2026

"Inference may live in superposition; authority never does."

Christopher S. Ciappa
Founder and Architect

Patent Pending • Copyright 2026 Samirac Partners LLC

PUBLIC DRAFT

Document Status and Purpose

This specification defines a common contract that must be assembled, validated, and admitted before a consequential AI action may cross an execution boundary. Version 0.2 adds a structural agent/task contract and explicit epistemic-state handling so bounded agents can know what they are, what work they are authorized to perform, what state is sufficient, and when a task is complete, blocked, or requires clarification.

Core rule: The model may propose an action. The architecture decides whether the action is admissible, under what authority, against which current state, and within what limits it may execute.

Item	Value
Specification	SAQ™ Execution Contract Specification
Version	0.2 — public draft
Issue date	August 13, 2026
Author	Christopher S. Ciappa, Samirac Partners LLC
Scope	AI and agentic systems that can cause consequential state change, transfer value, disclose information, commit resources, affect rights, or trigger downstream execution.
Normative language	MUST, MUST NOT, SHOULD, SHOULD NOT, and MAY are used as requirement terms.
Legal status	Technical specification and public provenance record. It is not legal, regulatory, medical, or financial advice.

What Changed in v0.2

- Adds an Agent Identity and Task Contract that binds an agent to a named role, objective, scope, prohibited actions, required state profile, and completion policy.
- Adds explicit epistemic states for claims and state assertions: ESTABLISHED, DERIVED, INFERRED, UNKNOWN, and CONFLICTED.
- Adds task completion and blocked-state rules so an agent does not rely on free-form conversational judgment to decide when its job is complete.
- Extends conformance testing to verify identity/scope enforcement, epistemic-state integrity, and completion gating.
- Adds the dAIsy Architecture Readiness Agent as an illustrative reference profile.

EXECUTION BEFORE EXPLANATION

A Common Contract for Consequential Action

No consequential AI action should execute merely because a model produced it.

An AI system needs a shared, machine-readable contract that binds a proposed action to identity, authority, evidence, current state, policy, constraints, admissibility, execution, and correction. A bounded agent additionally needs a structural task contract that defines its own identity, objective, scope, required state, and completion condition.

Model or Agent → Task Contract → Contract Assembled → External State Validated → Admissibility Decided → Bounded Execution → Receipt and Correction

The Standardization Boundary

The specification does not impose one universal policy on every industry. It defines a universal execution contract envelope and a compatible task-contract envelope. Each domain supplies its own authoritative sources, evidence, policy, constraints, consequence model, and required-state profile.

No domain rule is universal. The contract shape is universal.

THE COMMON CONTRACT DEFINES	THE DOMAIN PROFILE DEFINES
Who or what is acting; what action is proposed; which target and state are affected; what authority supports execution; which evidence and policies apply; what limits govern the grant; what happened and how it can be corrected.	Authoritative registries and systems; required evidence and freshness; prohibited actions; risk/consequence classes; escalation and approval rules; completion criteria; reversal/remediation duties; sector-specific conformance tests.

Expected Result

A conforming system can explain not only what the model recommended, but why execution was allowed, denied, constrained, deferred, or escalated; which authority supported that decision; which exact state was evaluated; what action actually occurred; and whether the agent had sufficient grounded state to complete the task at all.

NORMATIVE FOUNDATION

Design Principles

Principle	Requirement
1. PROPOSAL IS NOT PERMISSION	A generated action is a candidate. It MUST NOT acquire executable authority merely because it was produced confidently, repeatedly, or by a privileged model.
2. IDENTITY IS NOT AUTHORITY	Authentication establishes who or what is present. Authority establishes what that identity may do, to which target, under which conditions, and for how long.
3. REMEMBERED STATE IS NOT CURRENT STATE	Consequential execution MUST rely on externally validated current state when current state can change the admissibility decision.
4. EVIDENCE IS NOT INTERPRETATION	The contract MUST preserve provenance, source, timestamp, validation state, and epistemic state separately from model interpretation.
5. POLICY MUST BE EXPLICIT	Applicable policy MUST be named, versioned, and bound to the decision.
6. EXECUTION MUST BE NARROW	An authorization MUST be scoped to the exact action, target, limits, duration, and conditions admitted at the boundary.
7. CONTROL PRECEDES AUDIT	A conforming architecture MUST perform pre-execution control before relying on receipts and audit trails.
8. CORRECTION IS ARCHITECTURAL	When state, authority, evidence, or policy changes, the system MUST support revocation, suspension, reversal, compensation, or escalation.
9. AGENT IDENTITY MUST BE BOUNDED	An agent MUST NOT expand its own role, objective, authority, or completion criteria through fluent conversation.
10. UNKNOWN IS A VALID STATE	Missing, inferred, or conflicting information MUST NOT be silently promoted to established state merely to complete a task.

What the Execution Contract Is Not

It Is Not	Why
A prompt	Prompt text can influence generation but does not establish external authority or enforce execution boundaries.
A model confidence score	Confidence measures inference, not permission, truth, legal capacity, or current authoritative state.
A policy document alone	Requirements describe. Mechanisms enforce.
An audit log	A log records what happened. The contract determines whether the action may happen.
A permanent blanket token	Authority must be specific, bounded, expiring, and invalidatable.
A conversational persona	A persona may shape language. It does not structurally define agent identity, task scope, or execution authority.

Architectural rule: The model generated the response. The architecture decided whether it was allowed to become

reality — and whether the agent was structurally allowed to treat the task as complete.

STATE BEFORE ACTION

Contract Lifecycle and State Machine

A contract is not a static payload. It moves through controlled states. Each transition **MUST** be explicit, attributable, and recorded.

DRAFT → ASSEMBLING → VALIDATING → ADMISSIBILITY DECISION → ADMISSIBLE / CONDITIONAL / DENIED / ESCALATED / DEFERRED / EXPIRED → AUTHORIZED → EXECUTING → EXECUTED → RECEIPTED → CORRECTED / CLOSED

Required Lifecycle Rules

- 1. A contract **MUST** begin in DRAFT and **MUST NOT** be executable.
- 2. During ASSEMBLING, identity, task contract, action, target, authority, evidence, policy, and constraints are bound into one decision context.
- 3. During VALIDATING, the system resolves authoritative current state, verifies required evidence, and evaluates epistemic state.
- 4. The contract **MUST** receive an explicit outcome: ADMISSIBLE, DENIED, CONDITIONAL, ESCALATED, DEFERRED, or EXPIRED.
- 5. An admissible contract **MUST** still receive a narrow execution grant before crossing the execution boundary.
- 6. The executor **MUST** verify grant validity immediately before execution.
- 7. Execution **MUST** produce a receipt that records the realized action and resulting state.
- 8. Post-execution state changes **MAY** trigger correction, compensation, reversal, suspension, or review.

Task Lifecycle

INITIATED → STATE BUILDING → GAP / CONFLICT RESOLUTION → READY FOR ASSESSMENT → COMPLETE or BLOCKED / NEEDS CLARIFICATION

A task lifecycle is not an execution grant. It controls whether the agent may continue gathering state, request clarification, produce a bounded artifact, or declare the task blocked. A task **MUST NOT** be marked COMPLETE solely because the model produced a fluent answer.

Transition Integrity

Every transition must record

Prior state and new state; timestamp; responsible component or authority; reason code; policy/evidence versions; contract digest.

Forbidden transitions

DRAFT directly to EXECUTING; DENIED to EXECUTING without a new decision; EXPIRED grant to EXECUTING; material mutation after authorization; receipt without executor identity; silent policy substitution; UNKNOWN/CONFLICTED directly to ESTABLISHED without attributable resolution.

A material change to actor, action, target, authority, evidence, state, policy, constraints, agent identity, or task objective invalidates any prior decision that depended on that material and requires re-evaluation.

REQUIRED ENVELOPE

Canonical Contract Structure — Identity Through State

Object	Minimum Fields	Requirement
Contract Header	contract_id, schema_version, created_at, expires_at, correlation_id, digest	Uniquely identifies the contract and binds all material contents to a versioned representation.
Requester	requester_id, requester_type, organization, session, authentication_context	Identifies the entity initiating the proposal. A requester MAY differ from the acting executor.
Actor	actor_id, actor_type, runtime_id, model_id, agent_id, delegated_by	Identifies the model, agent, user, service, or composite actor proposing or performing the action.
Agent Identity / Task Contract	agent_profile_id, profile_version, role, objective, in_scope, out_of_scope, required_state_profile, completion_policy	Defines what the agent is, what bounded job it is performing, what it may and may not do, what state is required, and when the task is complete or blocked. It MUST NOT itself confer consequential execution authority.
Proposed Action	action_type, parameters, action_digest, requested_effect, reversibility	Defines the exact state change or consequential effect being requested. Parameters MUST be canonicalized before digesting.
Target	target_type, target_id, resource_scope, jurisdiction, data_classification	Defines what person, account, record, service, asset, or environment may be affected.
Authority Claim	authority_type, issuer, subject, scope, constraints, issued_at, valid_until, revocation_source	Names the external grant of authority. The model MUST NOT self-issue authority for consequential action.
Delegation Chain	delegator, delegate, scope, depth, conditions, signature or proof	Preserves how authority moved from an authoritative principal to the actor. Delegation MUST NOT widen scope.
Current State Snapshot	state_source, retrieved_at, state_version, state_digest, freshness_requirement, validation_result	Binds the decision to externally validated state.

Identity Separation

The contract SHOULD separate public identity, internal operational identity, external system identity, and privacy-preserving or pseudonymous identity. A system MAY use a Shadow ID or equivalent mechanism so execution can be governed without placing personally identifiable information into model context, vector stores, logs, or portable receipts.

Identity rule: A name match, login session, API key, model role, or conversational self-description is not enough. The contract must bind the correct identity to the correct task scope and valid authority for the exact proposed action.

REQUIRED ENVELOPE

Canonical Contract Structure — Evidence Through Correction

Object	Minimum Fields	Requirement
Evidence Set	evidence_id, type, source, provenance, observed_at, retrieved_at, digest, validation_state, epistemic_state	Preserves evidence separately from model interpretation. Epistemic state MUST be explicit when the claim can affect task completion or admissibility.
Policy Binding	policy_id, policy_version, issuer, effective_at, jurisdiction, rule_set_digest	Identifies the exact policy applied.
Risk and Consequence	risk_class, impact_domains, reversibility, value_at_risk, affected_subjects, escalation_threshold	Provides consequence context for stricter evidence, approval, or execution controls.
Constraints	amount_limit, target_limit, time_window, tool_limit, data_limit, geography, rate_limit, approval_requirements	Defines enforceable limits. Constraints MUST be machine-verifiable at execution time.
Admissibility Decision	outcome, reason_codes, decided_by, decided_at, evidence_digest, state_digest, policy_digest	Records the pre-execution decision and exact basis.
Execution Grant	grant_id, contract_id, scope, valid_from, valid_until, nonce, max_uses, issuer, grant_digest/signature	Provides narrow, time-bounded execution authority.
Execution Receipt	receipt_id, executor, started_at, completed_at, realized_action, result, resulting_state_digest, external_reference	Records what actually happened.
Correction Record	trigger, detected_at, correction_type, authority, related_receipts, result, residual_risk, closure_state	Documents revocation, reversal, compensation, suspension, remediation, or escalation.

Evidence Freshness

Each evidence type and current-state source SHOULD define a maximum acceptable age. When the permitted age expires, the evidence becomes stale and MUST NOT support a new execution decision unless policy explicitly permits it. The contract MUST record both when evidence was observed and when it was retrieved.

GROUNDING BEFORE COMPLETION

Epistemic State for Claims and Assertions

The architecture MUST distinguish what is established from what is merely derived, inferred, missing, or disputed. These states belong to the claim/evidence structure, not merely to generated prose.

State	Meaning	Structural Rule
ESTABLISHED	Explicitly stated by an attributable source or supported by supplied/validated evidence.	MAY participate in state assembly subject to provenance, freshness, and policy.
DERIVED	Follows from established state through an attributable derivation.	MUST preserve lineage to the supporting claims. A broken derivation invalidates the result.
INFERRED	Plausible or likely, but not established.	MAY guide questioning or analysis; MUST NOT silently become authoritative state or execution authority.
UNKNOWN	Required information is absent or insufficient.	MUST remain unresolved until supplied, retrieved, or explicitly accepted as unnecessary by policy.
CONFLICTED	Two or more attributable claims cannot simultaneously occupy the same material state slot without resolution.	MUST trigger arbitration, clarification, deferral, or escalation when material to the task or action.

State Promotion Rules

- User assertion MAY establish that the user asserted a proposition; it does not automatically establish that the proposition is externally true.
- Evidence-supplied claims MAY become ESTABLISHED when the applicable profile accepts the source, provenance, freshness, and validation state.
- INFERRED MUST NOT become ESTABLISHED merely because it is repeated by the model or appears in conversation history.
- UNKNOWN MUST NOT be filled by linguistic completion when the missing state is material.
- CONFLICTED MUST NOT be silently resolved by selecting the most fluent, recent, or confident statement.
- DERIVED claims MUST identify the inputs and rule or transformation that produced them.

Fluent language is not a state transition authority. An unsupported statement in prior model output does not gain epistemic authority merely because it becomes subsequent context.

BOUNDARY DECISION

Admissibility Outcomes and Execution Grants

Outcome	Meaning	Required Next Step
ADMISSIBLE	All required conditions are satisfied.	Issue a narrow execution grant if execution is required.
CONDITIONAL	Execution is permitted only with additional constraints or approvals.	Bind conditions and revalidate before grant.
DENIED	A rule, authority defect, evidence defect, prohibited target, or consequence threshold prevents execution.	Do not execute. Return reason codes and remediation path where appropriate.
ESCALATED	A designated human or external authority must decide.	Pause automation and transfer the complete contract record.
DEFERRED	The action cannot be decided until missing, stale, unknown, or conflicted information is resolved.	Retrieve or clarify required state, then create a new decision event.
EXPIRED	The contract, evidence, authority, or decision window is no longer valid.	Reassemble or revalidate; do not reuse the prior grant.

Task Completion Policy

A bounded agent SHOULD receive a machine-readable completion policy as part of its task contract. The runtime, not the model's conversational impulse, determines whether the task is ready to complete.

Task Result	Required Condition
COMPLETE	The required-state profile is sufficiently satisfied for the declared objective, and no unresolved material UNKNOWN or CONFLICTED state prevents a defensible result.
NEEDS_CLARIFICATION	A resolvable information gap or conflict remains and the agent is permitted to ask for clarification.
BLOCKED	Material unresolved state prevents a defensible result and cannot be resolved within the task's permitted scope.
OUT_OF_SCOPE	The requested activity falls outside the bound agent identity/task contract.
ESCALATED	The task requires a designated human, external authority, or higher-authority workflow.

Example completion rule: dAlsy's task is complete when it has either gathered sufficient state to produce the readiness brief, or has identified the unresolved information preventing a defensible assessment.

Constraint Enforcement

A constraint is not a suggestion. The component that performs the action MUST enforce constraints encoded in the grant. Where the executor cannot verify a required constraint, it MUST refuse execution or escalate.

REALITY AFTER EXECUTION

Execution Receipts, Revocation, and External Correction

The contract must follow the action into reality. A system is not governed merely because it made a correct pre-execution decision; it must also know what actually occurred and respond when authoritative reality changes.

PROPOSED	AUTHORIZED	REALIZED
What the model or agent asked to do.	What the boundary allowed under constraints.	What the executor actually changed in the world.

Correction Triggers

Trigger	Example	Potential Response
Authority revoked	User role or delegated authority is withdrawn.	Invalidate unused grants; suspend dependent workflows.
State drift	Account status, consent, inventory, or risk state changes.	Re-evaluate pending contracts; reverse or compensate when supported.
Evidence invalidated	A source is corrected, withdrawn, or shown to be unreliable.	Identify affected decisions and receipts; escalate or remediate.
Epistemic correction	A claim previously treated as ESTABLISHED is shown to be wrong or materially incomplete.	Downgrade or replace the claim, identify dependent derivations/decisions, and re-evaluate affected tasks/contracts.
Policy superseded	A newer binding rule changes admissibility.	Apply prospective rules and evaluate required retrospective action.
Execution variance	Actual effect differs from the granted effect.	Stop continuation, alert, compensate, or roll back.
External correction	An authoritative system states that AI-held state is wrong.	Update system state and route affected reasoning/execution back to reality.

Correction Is Not Memory Editing

Correction must operate on authoritative relationships among identity, state, evidence, epistemic status, authority, policy, task contract, execution contract, grant, and receipt. Merely changing a prompt, deleting a memory, or adding a warning does not repair executed reality.

Architecture must preserve the execution worldline: proposal → state → decision → grant → consequence → correction. It must also preserve the claim lineage that supported the decision.

IMPLEMENTATION PATTERN

Reference Architecture

1. Proposal Interface → 2. Task Contract Resolver → 3. Contract Builder → 4. Authority and State Adapters → 5. Policy Resolver → 6. SAQ Admissibility Gate → 7. Execution Grant Service → 8. Execution Broker → 9. Ledger and Correction Service

Component	Responsibility
Proposal Interface	Accepts a candidate action, task request, or supplied evidence. It does not confer authority.
Task Contract Resolver	Loads the bound agent identity, objective, in-scope/out-of-scope rules, required-state profile, and completion policy. It prevents conversational role expansion from becoming structural authority.
Contract Builder	Canonicalizes the action and assembles identity, task context, target, authority, evidence requirements, policy bindings, and constraints.
Authority and State Adapters	Query authoritative systems and return attributable state and authority records.
Policy Resolver	Selects the applicable domain profile and policy version for the actor, task, action, target, jurisdiction, and consequence class.
SAQ Admissibility Gate	Determines whether the action is allowable, denied, conditional, deferred, or escalated. It also checks whether unresolved epistemic state invalidates the decision.
Grant Service	Issues narrow, expiring, replay-resistant execution grants bound to the admitted contract.
Execution Broker	Verifies the grant and performs or delegates the exact allowed action.
Ledger and Correction Service	Records transitions and receipts; preserves claim lineage; detects drift; invalidates grants; initiates correction.

Trust Separation

The proposing model SHOULD NOT control the task-contract resolver, authority adapter, policy resolver, admissibility gate, grant verification, or receipt ledger. These components may use AI internally, but authoritative decisions and state transitions must remain independently verifiable.

Same input does not mean same state. Same language does not mean same authority. Admissibility depends on identity, task scope, evidence, epistemic state, policy, and current reality bound at decision time.

TESTING THE MECHANISM

Conformance Model

Conformance is demonstrated by mechanisms and testable behavior, not by policy statements, diagrams, prompts, or self-attestation.

Class	Required Capability	Example Test
EC-0 Agent/Task Contract	Binds agent identity, objective, scope, prohibited actions, required state, and completion policy.	A user asks the readiness agent to perform an out-of-scope external action; the runtime refuses or redirects without role expansion.
EC-1 Contract Producer	Produces canonical, versioned contracts with stable digests and required core fields.	Material action mutation changes the digest and invalidates the prior decision.
EC-2 Authority Validator	Resolves external authority, scope, expiry, delegation, and revocation.	A valid identity with insufficient scope is denied.
EC-3 State and Evidence Validator	Retrieves attributable state and verifies freshness, provenance, and epistemic state.	An INFERRED or stale claim cannot silently satisfy a required ESTABLISHED state slot.
EC-4 Admissibility Gate	Applies explicit policy and produces deterministic boundary outcomes and reason codes.	The same proposal receives different outcomes when authoritative state changes.
EC-5 Bounded Executor	Verifies grants and enforces action, target, time, use, and value constraints.	A replayed or expired grant is refused.
EC-6 Receipt Ledger	Records realized effects and links contract, grant, executor, resulting state, and relevant claim lineage.	A receipt exposes variance between authorized and realized action.
EC-7 External Correction	Invalidates, reverses, compensates, suspends, or escalates when authoritative reality changes.	Revoked authority or corrected evidence invalidates affected dependent grants or decisions.
EC-F Full Conformance	Implements EC-0 through EC-7 under an approved domain profile.	Passes the complete conformance suite and independent review.

Minimum Conformance Evidence

Implementation Evidence	Test Evidence
Architecture description; contract schemas; task profiles; state transition logs; policy/adaptor inventory; grant verification; correction behavior.	Positive/negative cases; out-of-scope identity test; inferred-to-established promotion rejection; conflict-handling test; stale evidence test; action mutation test; replay test; execution variance test.
Requirements describe. Mechanisms enforce. Conformance proves that the mechanisms actually operate.	

ONE CONTRACT, MANY DOMAINS

Domain Profiles

Domain	Authority / State / Constraints / Correction
PAYMENTS	Authority: account owner, delegated signer, payment mandate. State: balance, account status, fraud state. Constraints: amount, currency, destination, daily limit. Correction: cancel, reverse, dispute, compensate, freeze.
HEALTHCARE	Authority: licensed practitioner, patient consent, care protocol. State: identity, medication list, allergies, current orders, contraindications. Constraints: dosage, route, care setting, review requirement. Correction: hold, counter-order, notify, escalate, amend record.
ENTERPRISE AGENT	Authority: role, workspace permission, delegated tool scope. State: environment, record version, data classification. Constraints: allowed tools, repositories, spend, rate, rollback. Correction: revoke token, revert change, quarantine workflow.
HUMAN RESOURCES	Authority: designated manager, HR role, approved process. State: employment status, policy version, protected workflow state. Constraints: decision class, human approval, data minimization, jurisdiction. Correction: suspend, reopen, amend, notify.
INSURANCE	Authority: policy terms, adjuster scope. State: coverage, loss date, reserves, exclusions. Constraints: payment threshold, documentation, escalation. Correction: reopen, recover, amend, escalate.
CRITICAL INFRASTRUCTURE	Authority: operator role, maintenance order, emergency command. State: equipment mode, safety interlocks, load. Constraints: safe envelope, dual approval, change window. Correction: abort, fail safe, isolate, restore.
AI ARCHITECTURE READINESS	Authority: user-provided system description and supplied evidence for assessment purposes only. State: system purpose, actors, agents, tools, data, identities, authority boundaries, approval points, state persistence, provenance, execution exposure. Constraints: no external execution, no deployment approval, no silent resolution of material ambiguity. Completion: readiness brief or explicit blocked-state report.

Profile Requirements

Each profile MUST define required evidence, authoritative sources, freshness limits, policy bindings, risk classes, escalation rules, execution constraints, receipt requirements, correction mechanisms, required-state profile, and task-completion conditions. A profile MAY add domain fields but MUST NOT remove core contract integrity requirements.

Structural sequence: task identity → proposal → evidence assembly → identity resolution → authority determination → epistemic-state evaluation → conflict handling → admissibility → controlled execution → receipt → correction.

ILLUSTRATIVE STRUCTURE

Example Task Contract — dAIsy Architecture Readiness Agent

Illustrative only. Field names and cryptographic formats are subject to refinement in later versions.

```
{
  "agent_task_contract": {
    "agent_profile_id": "samirac.daisy.architecture-readiness",
    "profile_version": "0.1",
    "role": "AI Architecture Readiness Agent",
    "objective": "produce_readiness_brief",
    "in_scope": [
      "system purpose",
      "actors_and_agents",
      "models_tools_apis",
      "data_boundaries",
      "identity_and_authority",
      "human_approval_points",
      "state_persistence",
      "provenance",
      "execution_boundary_exposure"
    ],
    "out_of_scope": [
      "modify_customer_systems",
      "approve_deployment",
      "issue_legal_conclusions",
      "autonomous_remediation",
      "external_execution"
    ],
    "required_state_profile": "architecture-readiness.v1",
    "completion_policy": {
      "complete": "required_state_sufficient",
      "needs_clarification": "resolvable_material_gap",
      "blocked": "material_unresolved_state",
      "out_of_scope": "requested_action_outside_profile"
    }
  },
  "state": [
    {
      "claim": "refund_agent_can_issue_refunds",
      "epistemic_state": "ESTABLISHED",
      "source": "user_statement",
      "validation_state": "user_asserted"
    },
    {
      "claim": "maximum_refund_authority",
      "epistemic_state": "UNKNOWN"
    },
    {
      "claim": "human_approval_required",
      "epistemic_state": "CONFLICTED"
    }
  ],
  "task_status": {
    "result": "NEEDS_CLARIFICATION",
    "reason_codes": [
      "MAX_REFUND_AUTHORITY_UNKNOWN",
      "HUMAN_APPROVAL_CONFLICTED"
    ]
  }
}
```

What This Example Proves

The agent does not possess general architectural authority. It performs a bounded readiness task. It may organize established state, preserve derivations, surface inference, ask for missing information, and expose conflict. It may not silently convert

UNKNOWN or CONFLICTED state into ESTABLISHED state simply to finish. Completion is governed by the task contract and required-state profile, not by conversational momentum.

SAMIRACAI

FROM SPECIFICATION TO ECOSYSTEM

Versioning, Governance, and Adoption Path

Versioning

- Every contract **MUST** identify the schema version it implements.
- Every agent/task profile **SHOULD** identify its own profile version independently from the core contract version.
- Minor versions **MAY** add optional fields and compatible clarifications.
- Major versions **MAY** change required fields, lifecycle semantics, or conformance behavior.
- Policy and domain profiles **MUST** be independently versioned from the core contract.
- A receipt **MUST** preserve the versions actually used at decision and execution time.

Adoption Sequence

Publish v0.2 → Implement task contract + epistemic state in dAIsy/SAQ → Architecture Readiness profile → Additional domain profiles → Partner applications → Conformance suite → Certification and licensing

Reference Implementations	Ecosystem Assets
dAIsy governed agent runtime; dAIsy Architecture Readiness Agent; SAQ admissibility gate demonstration; payment/commercial action profile; receipt/correction demonstration.	Machine-readable schemas; SDKs and policy adapters; task-profile definitions; required-state profiles; test vectors and conformance harness; developer/application certification; commercial domain profiles.

Specification Governance

Samirac Partners LLC is the initial specification steward. Future versions may incorporate documented feedback from implementers, domain experts, customers, regulators, and standards participants. Changes must preserve separation among inference, epistemic state, identity, task scope, authority, state, policy, admissibility, execution, and correction.

Commercial Model

The core contract can serve as the public interoperability reference. Commercial value may be delivered through licensed runtime mechanisms, SDKs, domain profiles, task profiles, implementation reviews, partner applications, conformance testing, certification, support, and integration services.

CATEGORY DEFINITION

The Execution Contract Becomes the Shared Boundary

AI does not need one universal mind. It needs a universal way to prove what may become real — and a bounded way to establish what an agent is allowed to treat as known, in scope, and complete.

The SAQ Execution Contract provides that structure. It gives consequential actions a common envelope while allowing each domain to supply its own authority, evidence, policy, constraints, required state, completion rules, and correction model.

Agent Identity → Task Contract → Proposal → Evidence → Epistemic State → Current State → Authority → Policy → Admissibility → Bounded Execution → Receipt → Correction

- The standard defines the common execution contract.
- The task profile defines the bounded identity, objective, scope, required state, and completion policy.
- The domain supplies authority, evidence, policy, consequence, and correction requirements.
- The implementation supplies enforcement.
- The receipt proves what happened.
- External correction keeps the system aligned with reality.

Initial Public Commitments

- Publish a dated, versioned specification.
- Build the task contract, epistemic-state model, and execution contract into live Samirac implementations.
- Demonstrate allow, deny, conditional, escalation, deferral, expiry, replay rejection, receipt, correction, conflict handling, and blocked-task behavior.
- Implement the dAIsy Architecture Readiness Agent as a bounded reference profile.
- Enable partners to build applications against stable execution and task-contract structures.
- Create a credible conformance and certification path based on operating mechanisms.

Christopher S. Ciappa
Founder and Architect, Samirac Partners LLC
August 13, 2026

samirac.com
SAQ™ • Drift Stack™ • DeltaDrift™