EVIDENCEBOUND-MAS

Verifiable Evidence for Human Decision-Making

Contestable Verification for AI-Generated Analysis

A bounded, fail-closed architecture for explicit specifications, deterministic execution, reproducible Proof Packs, and human contestation

CORE CLAIM AI-generated executable analysis should reach operational systems only through explicit, reproducible and independently reviewable boundaries.

CONTROL	VALUE
Prepared for	Cosmos Institute - AI for Truth-Seeking
Applicant	Ruslan Vrublevskyi
Location	Kyiv, Ukraine
Version	2.0 - Grant Review Edition
Date	25 July 2026
Technical evidence heads	Open-core merge 2719e078f957... | Enterprise merge 319452ffb832... | Exact-head QA ca3739d1d3c...
Public demonstration	https://signalreview.co/proof-center
Document status	Technical whitepaper and grant roadmap; not certification, legal advice, or production acceptance

00 Abstract

From probabilistic generation to contestable evidence

EvidenceBound-MAS is a proposed standalone truth-seeking infrastructure project derived from repository-backed verification controls implemented inside SignalReview. It treats AI-generated Python and analytical claims as untrusted artifacts, then binds explicit specifications, immutable inputs, restricted source semantics, formal obligations, deterministic execution, and human-readable evidence into a reproducible Proof Pack.

The current implementation is evidence for a substantial control stack; it is not a universal proof of truth, alignment, complete containment, or production suitability. The 90-day grant project will convert the existing private-monorepo capability into a public, contestable prototype that independent reviewers can run, inspect, reproduce, challenge, reject, and escalate.

READING RULE Every use of VERIFIED is bounded by the recorded specification, supported semantic profile, input snapshot, environment fingerprint, and proof policy. It never means “true in the external world.”

Contents

01 Executive Summary

What exists, what the grant will deliver, and what is explicitly not claimed

- AI systems increasingly generate executable analysis. When generated code can reach tools, databases, financial controls, scientific workflows, or operational infrastructure without reproducible controls, model error can become system action.
- EvidenceBound-MAS intervenes at the boundary between generation and actuation. It combines typed contracts, a fail-closed AST policy, Z3 obligations, deterministic AST-to-IR execution, per-request semantic-preservation checks, content-addressed Proof Packs, optional Ed25519 sealing, and mandatory human authorization.
- The merged repository baseline already contains a six-phase verification core, a CLI reproduction contract, tamper detection, controlled Proof Explorer evidence, and a private isolated worker boundary. Standalone public extraction, public service hardening, independent reproduction, benchmark evidence, and human-contestation measurement remain grant work.
- The 90-day commitment is outcome-based: a first-time reviewer can inspect, reproduce, challenge, and reject an AI-generated analysis. Four architecture deliverables are committed; six additional workstreams are adaptive research that may be redesigned or deferred based on evidence.
- The project aligns with Cosmos AI for Truth-Seeking because it supports open contestation and preserves human judgment rather than asking users to defer to an opaque model verdict.

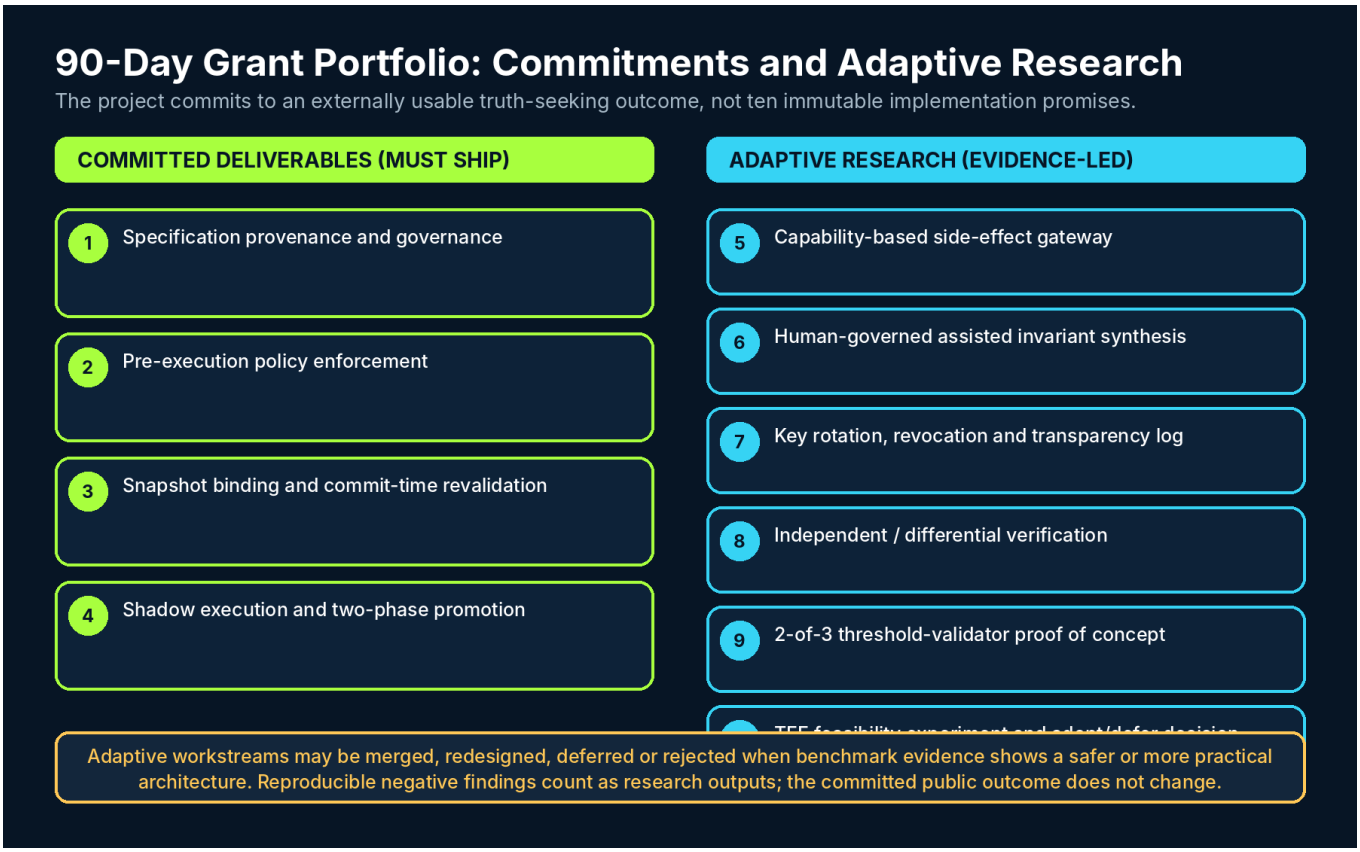

Figure 1. The grant separates committed public outcomes from evidence-led research workstreams.

02 Truth-Seeking Problem

The gap between "the code ran" and "a human can challenge why it should be accepted"

The target failure mode is not ordinary hallucination in a chat response. It is the transition from a probabilistic model output to executable analysis with operational consequences. Conventional unit tests and sandboxes answer narrow questions: whether a program compiled, whether selected tests passed, or whether obvious capabilities were unavailable. They rarely bind the code, the governing specification, the exact inputs, the execution semantics, and the resulting evidence into one independently reproducible object.

FAILURE MODE	WHY ORDINARY CONTROLS ARE INSUFFICIENT	EVIDENCEBOUND RESPONSE
Specification error	A program can correctly implement a wrong or incomplete policy.	Versioned specification provenance, explicit assumptions, human approval and

FAILURE MODE	WHY ORDINARY CONTROLS ARE INSUFFICIENT	EVIDENCEBOUND RESPONSE
		contestation.
Capability overreach	Passing tests does not prevent undeclared network, filesystem or process access.	AST capability policy, isolated worker, resource limits and planned capability gateway.
Semantic drift	Generated source may not match translated IR or concrete execution.	Canonical AST/IR bindings and per-request semantic-preservation obligations.
State drift	Inputs may change after verification but before promotion.	Committed immutable snapshot binding and commit-time revalidation.
Evidence tampering	A result may be copied, modified or detached from its inputs.	Content-addressed Proof Pack, deterministic root and optional trusted node seal.
Automation bias	A technical PASS can be mistaken for truth or approval.	Visible non-claims, counterexamples, UNKNOWN/BLOCKED states and explicit reject/escalate actions.

TRUTH-SEEKING OUTCOME The system does not replace judgment. It lowers the cost of asking: What was assumed? What was proved? What failed? Can I reproduce it? Should I reject it?

03 Existing EvidenceBound Architecture

Repository-backed controls as the starting point, not the completed grant product

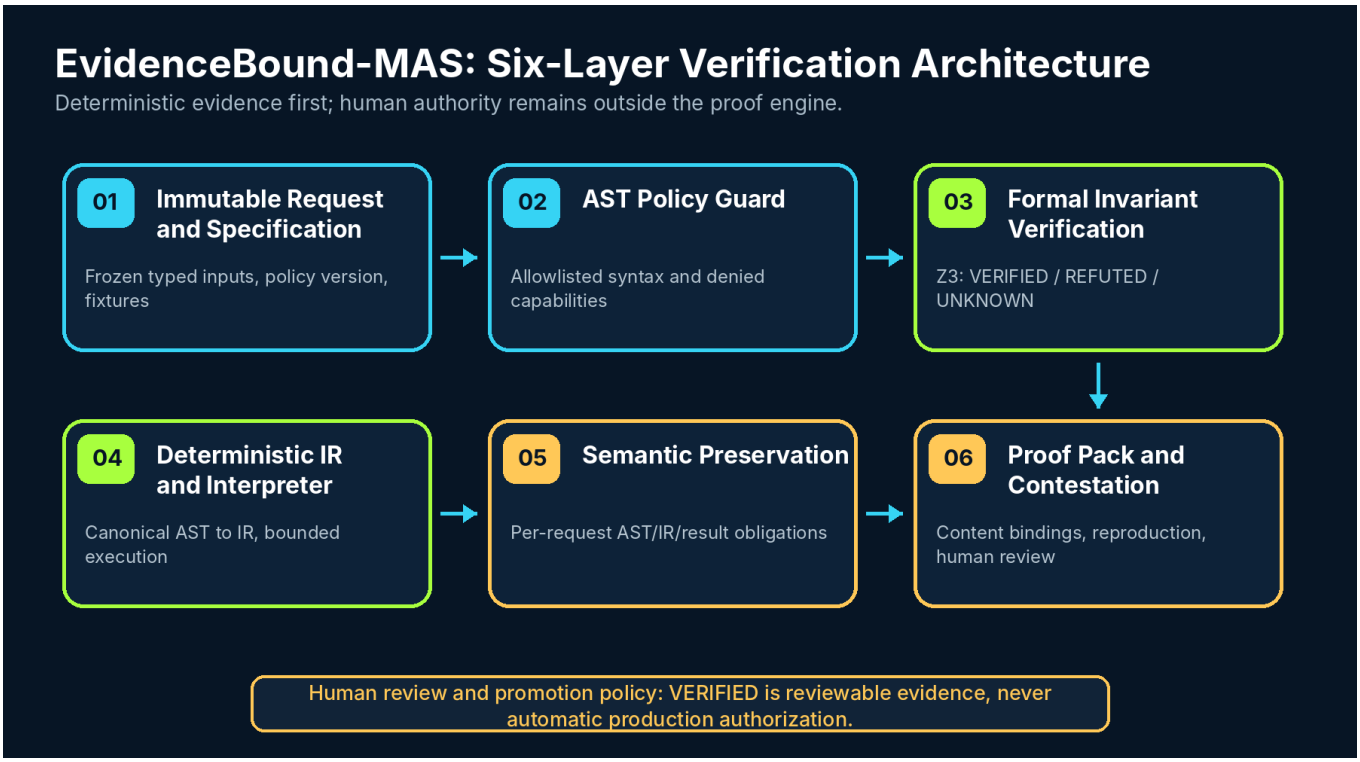

Figure 2. Current six-layer architecture and the cross-cutting human authorization boundary.

EvidenceBound-MAS is model-agnostic. The candidate may originate from a hosted model, an open-weight deployment, a deterministic generator, or a human. The same verification boundary applies after generation. LLM agents may propose or interpret; deterministic components own policy, execution, binding, proof status, and promotion eligibility.

ARCHITECTURE BOUNDARY SignalReview's four-role sports evidence review and EvidenceBound code verification are separate workflows. The sports agents do not receive permission to invent provider facts, proof statuses, probabilities, signatures, or production approvals.

04 Six Verification Layers

What each layer supports - and what it cannot establish

#	LAYER	MECHANISM	BOUNDED CLAIM
1	Immutable Request and Specification	Typed request, fixtures, policy version, declared inputs, invariants, supported expression profile.	Exact contract identity; not correctness of the policy itself.
2	Restricted AST Policy	Canonical Python AST, allowlisted constructs, denied imports/calls/attribute roots, bounded source policy.	Syntactic/capability compliance; not complete semantic safety.
3	Formal Invariant Verification	Exact Z3 translation for Boolean, integer and fixed-point obligations; counterexamples when satisfiable.	Proof/refutation within declared theories; not external-world truth.
4	Deterministic Execution and Source Binding	Canonical AST to execution IR, bounded interpreter, typed result and artifact identities.	Exact source/IR/result binding; not a universal compiler theorem.
5	Semantic Preservation	Independent AST and IR	Per-request equivalence in

#	LAYER	MECHANISM	BOUNDED CLAIM
		symbolic semantics, translation validation and concrete result checks.	supported scalar profile; containers/strings remain outside scope.
6	Proof Pack and Contestation	Deterministic manifest body, dynamic wrapper, SHA-256 bindings, reproduction, tamper detection, optional node seal, human inspection.	Integrity and reproducibility; not authorship unless a trusted key verifies, and never automatic approval.

```
if request_or_policy_invalid:       BLOCKED
elif negated_invariant_is_sat:       REFUTED + counterexample
elif solver_or_profile_unresolved:    UNKNOWN
elif artifact_binding_mismatch:       MANIFEST_TAMPERING_DETECTED
elif all_supported_obligations_hold:  VERIFIED (bounded)
else:                                 BLOCKED
```

Z3 may return sat, unsat, or unknown; EvidenceBound maps these outcomes into explicit proof states rather than converting uncertainty into confidence [2]. Python AST analysis is version-sensitive because the abstract grammar evolves with Python releases, so the environment fingerprint and supported toolchain are part of the evidence boundary [3].

05 Proof Pack and Contestability

A proof artifact designed to be inspected, reproduced, challenged, and rejected

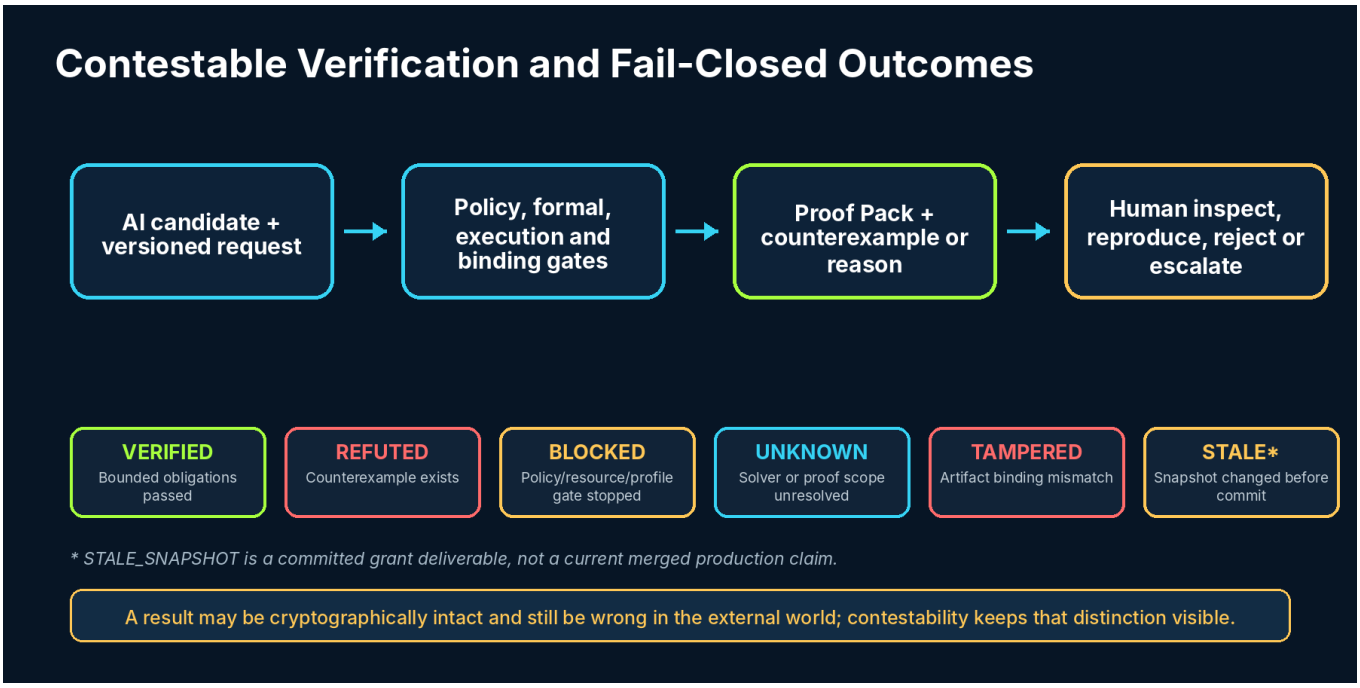

Figure 3. Verification produces contestable states and evidence, not a single opaque approval.

The Proof Pack separates a canonical deterministic body from dynamic metadata. The deterministic body binds the request, canonical AST, execution IR, execution result, Z3 context, formal proof, source-binding proof, semantic-preservation proof, environment fingerprint, safety contract, and artifact digests. Timestamp and optional signing metadata live outside the root so they do not change deterministic reproduction.

COMPONENT	PURPOSE	REVIEWER QUESTION
request.json	Frozen inputs, fixtures, policy and specification identity.	What exactly was requested and assumed?
canonical AST	Normalized source identity and policy surface.	What syntax and capabilities were admitted?
execution IR	Deterministic executable representation.	What program did the bounded interpreter execute?
formal proof / counterexample	Z3 obligations and satisfiability evidence.	Why was the invariant verified or refuted?
semantic proof	AST/IR/result correspondence within supported profile.	Did translation or execution change the declared semantics?
manifest body and ledger	Content-addressed bindings and environment identity.	Can any bound artifact be replaced without detection?
optional Ed25519 seal	Node assertion over the canonical body digest.	Does an independently trusted public key authenticate the issuer?

CONTESTABILITY RULE A reviewer may reject a VERIFIED pack because the specification is wrong, the input source is untrusted, the scope is insufficient, or the operational consequence is unacceptable. Cryptographic integrity does not remove human authority.

06 Current Verified Technical Baseline

Evidence snapshot for the grant application

AREA	VERIFIED REPOSITORY EVIDENCE	STATUS / LIMITATION
Core phases 1-6	Immutable contracts, AST guard, Z3 invariants, deterministic source-to-IR binding, semantic preservation, Proof Pack CLI/reproduction, mutation corpus, threat model and read-only Proof Explorer.	Implemented inside the SignalReview monorepo; standalone public extraction remains grant work.
Open-core seal contract	Backward-compatible Ed25519 seal schema and compatibility tests. Merge 2719e078f9573049483e1f7f88041bc0b6622752.	Schema support exists; production key custody is not claimed.
Private enterprise boundary	Separate package, per-job spawned worker, Unix-socket IPC, bounded timeout, kill switch, optional node sealing, trusted-key verification, hardened deployment reference. Merge 319452ffb832853ec12c845b9286b250dfd87dcc.	Repository acceptance only; installation and production operations were not performed.
Exact-head QA evidence	Enterprise/core compile, Ruff, strict mypy, pytest contracts, schema/audit drift, Qwen/cross-stack gates, current-tree and complete-history secret scans passed at ca3739d1d3c9a5465ce9b6e8464276d0e9642ecb.	A successful workflow proves named gates for that revision, not universal security.
Public evidence surface	SignalReview /proof-center route and proof-scope section provide controlled inspection.	Read-only demonstration; not yet the complete standalone public service.

SENSITIVE BOUNDARY No production private key, KMS/HSM, environment mutation, DNS change, systemd installation, customer-data migration, billing change, or production deployment is represented by this whitepaper.

07 Exact Proof Boundary and Non-Claims

Why bounded claims are stronger than absolute safety language

Phase 5 is a mechanized per-request proof for a supported scalar analytical profile. It establishes a chain of equality only when the declared input domain, supported operators, signed-i64 behavior, fixed-point scale, branches, bounded loops, and concrete interpreter result satisfy all encoded obligations:

```
approved canonical source AST semantics
    == canonical execution IR semantics
    == concrete deterministic interpreter result
```

EVIDENCEBOUND MAY SUPPORT	EVIDENCEBOUND DOES NOT CLAIM
A candidate satisfied an encoded syntactic and capability policy.	That all malicious behavior or aliases were detected.
The negation of a declared invariant was unsatisfiable in the encoded theory.	That the human-authored invariant is correct, complete, ethical or factually true.
Canonical artifacts reproduce under the declared environment fingerprint.	That results reproduce across undeclared toolchains or unsupported environments.
A trusted Ed25519 key authenticated the canonical manifest digest.	That the key was never compromised, that the host was honest, or that the content is true.
A supported request was VERIFIED, REFUTED, BLOCKED, UNKNOWN or TAMPERED.	Universal compiler correctness, model alignment, forecast accuracy, profitability, regulatory compliance or production suitability.

PUBLIC CLAIM EvidenceBound-MAS provides bounded, reproducible evidence that a candidate satisfies an operator-approved specification under a recorded environment and proof policy. It does not claim universal truth or absolute safety.

08 Five Core Limitations

The grant is justified by visible technical boundaries, not by claiming a complete solution

LIMITATION	CURRENT IMPACT	GRANT RESPONSE
1. Specification / “garbage-in” risk	The verifier can rigorously prove compliance with an incomplete, malicious or mistaken specification. It cannot infer pragmatic truth from syntax alone.	Specification provenance, human-governed policy approval, coverage reports and pre-execution enforcement.
2. Latency and solver overhead	Complex formal obligations, process isolation and full artifact generation can exceed interactive latency targets. A 250 ms target is workload-dependent, not universal.	Shadow execution, bounded profiles, staged proof modes and two-phase promotion; measure p50/p95/p99 and timeout rates.
3. State drift and snapshot invalidation	A proof may be internally valid but obsolete if authoritative inputs change between verification and promotion.	Content-addressed snapshots, freshness envelope, state version / ETag, commit-time compare-and-swap and STALE_SNAPSHOT.
4. Manual invariant bottleneck	Writing complete formal invariants is expensive and requires domain expertise. Direct natural-language-to-Z3 automation risks silent weakening.	Typed policy DSL, assisted synthesis, ambiguity/coverage reports, counterexamples and mandatory human approval.
5. Validator key or host compromise	A stolen signing key or compromised verifier host can issue apparently authentic but invalid reports. A signature authenticates the key, not the truth.	Rotation, revocation, transparency log, independent validators, supply-chain provenance and TEE feasibility research.

These limitations are not defects hidden from reviewers; they define the experimental agenda and acceptance criteria. A negative or revised research result is useful when it is reproducible and does not displace the committed public outcome.

09 Four Committed 90-Day Deliverables

Minimum architecture that the grant project will ship

#	COMMITTED DELIVERABLE	ACCEPTANCE EVIDENCE
1	Specification provenance and governance	Versioned specifications with author/reviewer identity, policy source, effective date, approval status, revocation state, digest, examples, assumptions and change justification. Promotion rejects unapproved, revoked or unbound specifications.
2	Pre-execution policy enforcement	A strict policy stage validates the typed request, supported semantic profile, resource/capability envelope and operator-approved invariants before deterministic execution. Violations are BLOCKED; refutable obligations return counterexamples.
3	Snapshot binding and commit-time revalidation	Each pack binds source identity, data snapshot digest, version/sequence, freshness envelope and expected authoritative-state version. A mismatch before promotion returns STALE_SNAPSHOT and requires re-verification.
4	Shadow execution and two-phase promotion	Unverified candidates execute only against quarantined state and produce a declarative proposed state delta. No external side effect or authoritative write occurs before verification and compare-and-swap authorization.

STOP CONDITION The 90-day commitment is not complete until a first-time public reviewer can run a case, inspect proof scope, provoke a controlled counterexample or stale/tamper failure, download the complete Proof Pack, reproduce it locally, and reject or escalate the claim.

10 Six Adaptive Research Workstreams

Important directions whose implementation may change as evidence develops

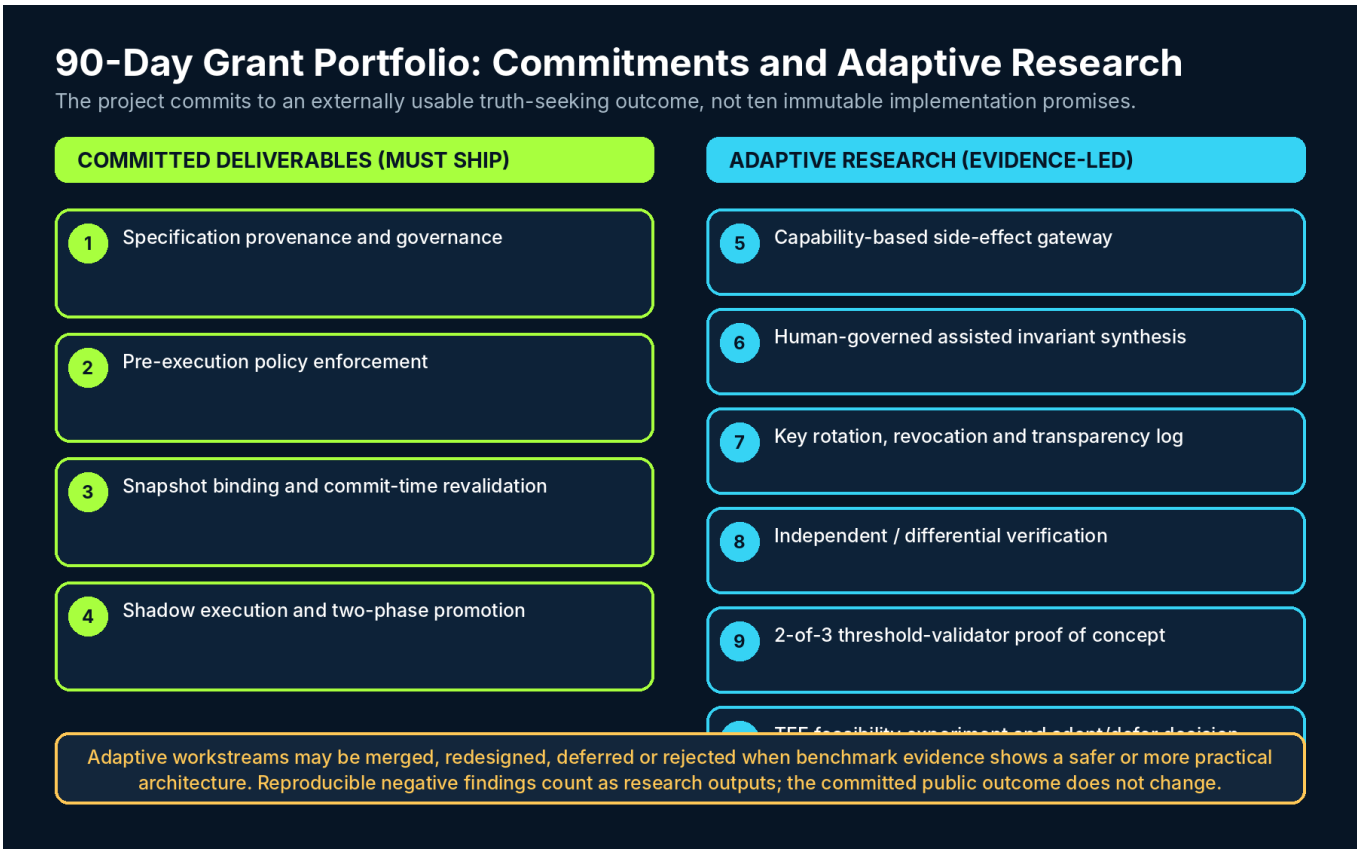

Figure 4. Four mandatory outcomes and six adaptive research workstreams.

#	WORKSTREAM	RESEARCH QUESTION	DISPOSITION RULE
5	Capability-based side-effect gateway	Replace direct network/database/filesystem authority with signed, declarative, allowlisted action requests constrained by tenant, resource, volume, expiry and state version.	Prototype if shadow-promotion integration demonstrates a real side-effect boundary.
6	Human-governed assisted invariant synthesis	Translate typed policy graphs/DSL into candidate Z3 invariants with ambiguity, conflict, unsupported-feature and coverage reports. No automatic activation.	Keep only if benchmark false-negative and human-review behavior are acceptable.
7	Key rotation, revocation and transparency log	Test key lifecycle, trust-store updates, revoked-key rejection and append-only Proof Pack checkpoints.	Deliver a local or public prototype; defer production KMS/HSM commitments.
8	Independent / differential verification	Cross-check critical obligations with an independent interpreter, checker, second solver profile or mutation harness.	Adopt selectively where it reduces correlated implementation risk at acceptable cost.
9	Threshold validator proof of concept	Collect 2-of-3 independent seals with separated keys/failure domains and explicit disagreement state.	Research PoC only; not a production decentralized network promise.
10	TEE feasibility experiment	Evaluate AMD SEV-SNP or another available confidential-computing environment for key and worker isolation, attestation, latency and operating cost.	Publish adopt/defer/reject evidence; no production TEE guarantee.

11 Public CLI/API/Proof Explorer Deliverable

The externally usable product that turns repository capability into truth-seeking infrastructure

PUBLIC COMPONENT	MINIMUM USER JOURNEY	SECURITY / TRUST BOUNDARY
Standalone CLI	evidencebound verify request.json --output proof-pack/; evidencebound reproduce proof-pack/.	Versioned package, pinned dependencies, environment fingerprint, no private SignalReview IDs.
Read-only verification API	Submit only supported versioned requests; poll explicit status; download a complete service-generated Proof Pack.	Networkless non-root worker, tenant isolation, rate/source/fixture/fuel/memory/solver/time limits, redacted logs.
Proof Explorer	Inspect request, policy result, AST identity, IR identity, execution result, invariant table, counterexample, semantic obligations and digest chain.	No arbitrary public source upload until worker security gate passes; no user-data writes to SignalReview.
Contestation controls	Mutate a controlled source/specification fixture, observe changed digest/status, reject or escalate.	Controlled corpus first; clear distinction between replay and fresh service execution.
Download and reproduction	Download the full pack and reproduce from public instructions on a fresh supported machine.	Tamper and toolchain drift remain separate outcomes; no hidden credential dependency.

```
Public acceptance path

select example or submit allowed versioned request
  -> run fail-closed verification
  -> inspect policy, AST, IR, execution and specification
  -> see VERIFIED / REFUTED / BLOCKED / UNKNOWN / TAMPERED / STALE
  -> download content-addressed Proof Pack
  -> reproduce locally
  -> contest, reject or escalate
```

12 Adversarial Benchmark and Human Evaluation

Technical detection is necessary; correct human interpretation is part of the product

BENCHMARK FAMILY	EXAMPLES	PRIMARY METRICS
Policy / capability	Forbidden imports, indirect attribute roots, subprocess/network/filesystem attempts, malformed request.	Covered-violation block rate, bypass rate, false-positive rate.
Formal semantics	Branch drift, inconsistent specification, signed-i64 overflow, fixed-point scale mismatch, division, fuel exhaustion.	Proof completion, counterexample quality, UNKNOWN and timeout rates.
Binding / integrity	AST/IR/result mutation, manifest-root change, signature substitution, revoked key, toolchain drift.	Tamper detection, correct error classification, reproduction rate.
State / promotion	Stale snapshot, version race, mismatched ETag, conflicting state delta, attempted irreversible side effect.	STALE detection, zero unsafe promotion, retry behavior.
Distributed / supply chain	Split-brain validators, correlated defect fixture, mismatched build identity, revoked release fingerprint.	Disagreement visibility, provenance coverage, recovery time.
Human contestability	Misleading VERIFIED wording, incomplete assumptions, counterexample navigation, reject/escalate tasks.	State comprehension, time to locate evidence, correct rejection/escalation, over-trust rate.

TARGET	MEASUREMENT RULE
Deterministic-body stability	100% across the supported environment matrix for the same declared fingerprint and canonical inputs.
Tamper detection	100% for the pre-agreed mutation corpus; not a claim about all conceivable attacks.
Unsafe automatic promotion	Zero in the benchmark and public prototype.
Unsupported / unknown visibility	Every unsupported or unresolved case remains non-verified.
Independent reproduction	At least two reviewers reproduce public cases from instructions, target <=15 minutes for canonical examples.
Performance	Report p50/p95/p99, queue, solver, execution and proof-pack cost; no universal 250 ms guarantee.
Human understanding	Report percentage correctly distinguishing VERIFIED from external truth and correctly locating assumptions/counterexamples.

TERMINOLOGY An unsafe case incorrectly returned VERIFIED is a false negative and a safety incident. A valid case incorrectly blocked is a false positive and a quality/reliability issue.

13 Threat Model

Assets, adversaries, trust boundaries, residual risks, and fail-closed intervention points

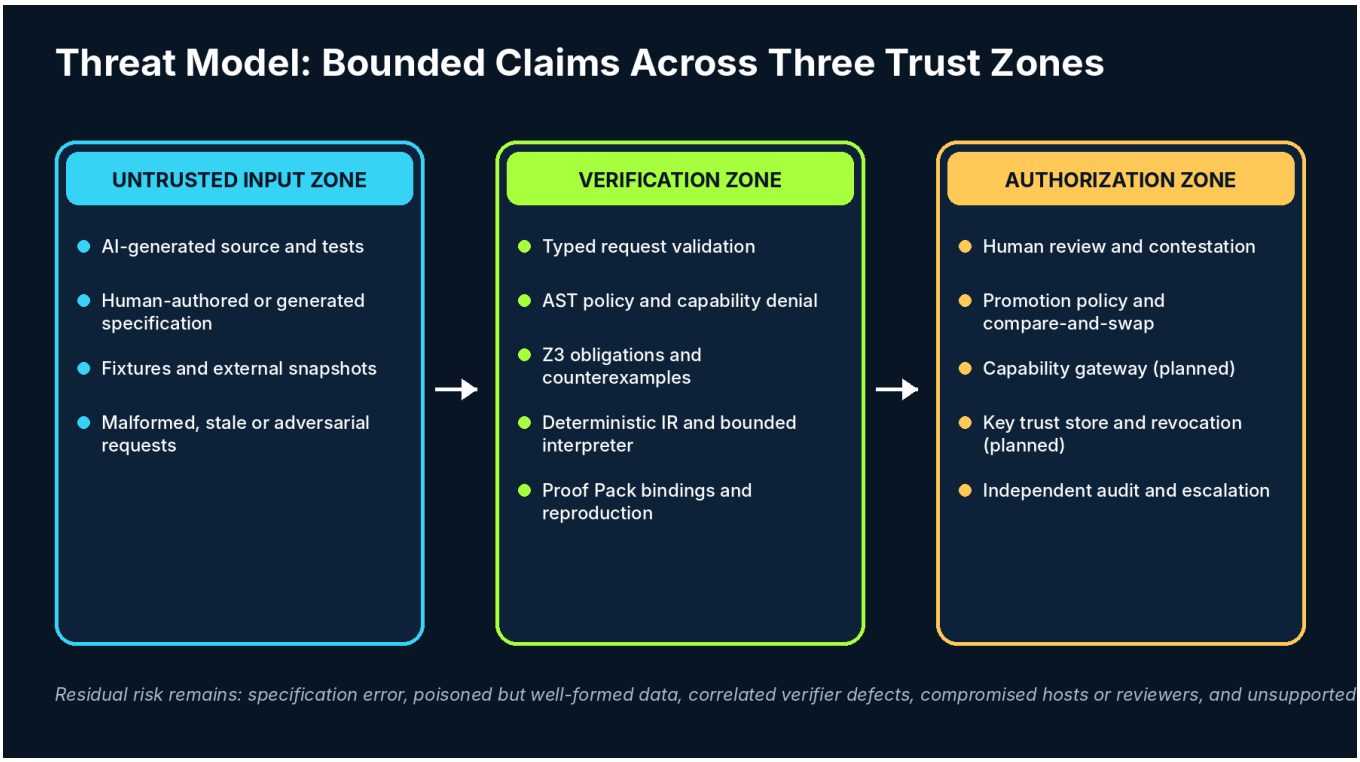

Figure 5. The architecture narrows risk through explicit zones; it does not eliminate external-world or host-level uncertainty.

ASSET / PROPERTY	THREATS CONSIDERED	RESIDUAL OR OUT-OF-SCOPE RISK
Specification and policy integrity	Unauthorized change, revoked version, missing provenance, inconsistent invariant.	Approved but wrong policy; colluding or compromised approver.
Source and execution identity	Forbidden syntax/capabilities, source-to-IR drift, interpreter mismatch, unsupported profile.	Unknown policy bypass, dangerous allowed construct, implementation defect.
Inputs and state	Fixture tampering, stale snapshot, state race, missing binding.	Poisoned but correctly hashed data; dishonest source provider.
Proof Pack	Artifact substitution, root mismatch, replay under wrong fingerprint, signature substitution.	Compromised trusted key/channel; deletion or suppression without transparency witness.
Worker and host	Resource exhaustion, unauthorized Unix-socket peer, timeout, partial output, network access.	Kernel/runtime vulnerability, malicious administrator, supply-chain compromise.
Human authority	Automation bias, hidden UNKNOWN/BLOCKED, accidental auto-promotion.	Malicious or negligent reviewer; governance failure.

Docker security depends on kernel isolation, daemon configuration, capabilities, seccomp and host hardening; a container is not an absolute security boundary [4-6]. The planned TEE experiment evaluates an additional confidential-VM boundary, but TEE adoption cannot correct specification errors or application logic defects [10].

14 Milestones and Budget

A realistic 90-day working prototype for a $10,000 grant request

Figure 6. Three delivery phases preserve a public outcome while allowing research adaptation.

WORKSTREAM	USD	DELIVERABLE
Standalone packaging, schemas and specification provenance	$2,000	Public repository/release path, CLI, versioned request/proof schemas, canonical examples, provenance and approval contracts.
Hardened worker/API, snapshot binding and shadow promotion	$3,000	Networkless bounded service, state-version binding, STALE outcome, two-phase compare-and-swap promotion and redacted evidence.
Proof Explorer and contestation UX	$1,500	Service-generated packs, proof-scope inspection, counterexamples, mutation controls, reject/escalate journey.
Adversarial benchmark and external reproduction	$2,500	Corpus, metrics, cross-platform reproduction, two independent reruns and public limitations report.
Documentation and human evaluation	$1,000	Threat model, non-claim matrix, technical report, demo and comprehension/contestation study.
TOTAL	$10,000	Prime Intellect credits, if awarded, are reserved for generating/evaluating adversarial candidates; deterministic verification remains authoritative.

Milestone acceptance gates and adaptive fallback

The public outcome remains fixed while implementation choices may change when benchmark evidence identifies a safer, simpler or more reproducible architecture. Every material change is recorded with rationale, impact, evidence and revised acceptance criteria.

MILESTONE	ACCEPTANCE GATE	FALLBACK / ADAPTATION
Day 30	Standalone installation; versioned specs; policy and snapshot contracts; reproducible canonical examples.	Reduce supported semantic profile rather than weaken fail-closed behavior.
Day 60	Public read-only service and Proof Explorer; shadow/two-phase promotion demonstrated on controlled workloads.	Keep uploads curated until isolation and abuse gates pass.
Day 90	Benchmark, human study, independent reproductions, final report and public walkthrough.	Publish negative findings and revised architecture for adaptive workstreams.

REPORTING DISCIPLINE Monthly evidence snapshots will distinguish shipped capability, measured results, revised hypotheses, deferred work and unresolved risks. Stretch work cannot displace the four committed outcomes.

15 Commercialization Path

From a grant-funded public truth-seeking artifact to a governed enterprise product

Figure 7. Open evidence standards create trust; enterprise deployment and governance create the commercial boundary.

STAGE	CUSTOMER / USER JOB	EVIDENCE REQUIRED BEFORE ADVANCING
Grant prototype	Independent reviewer inspects and contests AI-generated executable analysis.	Public journey, benchmark, reproduction and comprehension evidence.
Design-partner pilots	Financial, compliance, insurance or scientific teams evaluate customer-controlled analytical workloads in a segregated environment.	2-4 design partners, integration evidence, false-block review, cost/latency, operator feedback.
Enterprise gateway	Organization governs AI-generated actions through private workers, policy/spec registry, approvals, Proof Pack retention and support.	At least one paid pilot, recurring workflow, measurable audit/risk outcome, acceptable marginal cost.
Validation ecosystem	Independent operators or implementations attest to portable Proof Packs and transparency checkpoints.	Demonstrated demand for multi-party trust, governance model and non-correlated validation.

The commercial hypothesis is not that customers pay for abstract “AI safety.” They may pay when EvidenceBound reduces a recurring, expensive approval or audit problem: controlled promotion of AI-generated analytical code, reproducible evidence, policy governance, private deployment, and incident-ready support. Willingness to pay remains unverified until a customer completes and pays for a pilot.

OPEN / PUBLIC LAYER	PRIVATE ENTERPRISE LAYER
Verification schemas, Proof Pack standard, CLI, canonical examples, benchmark methodology and public contestation interface.	Managed/on-prem workers, policy governance, tenant controls, validator/key management, enterprise integrations, evidence retention, SLA and support.

PRODUCTION GATE No production enablement is implied by the grant. Production secrets, KMS/HSM, DNS, UID/GID, systemd installation, customer data, billing, authentication or deployment changes require separate owner-approved work.

16 References and Evidence Record

Primary technical references and repository anchors

REF	SOURCE	URL
[1]	Cosmos Institute, “Cosmos Grants - Fast funding for AI prototypes,” AI for Truth-Seeking, typical $1k-$10k award and 90-day working-prototype period.	https://www.cosmos-institute.org/grants
[2]	Microsoft Research, Online Z3 Guide. Z3 checks satisfiability and may return sat, unsat or unknown.	https://microsoft.github.io/z3guide/docs/logic/basiccommands/
[3]	Python Software Foundation, Python ast module documentation.	https://docs.python.org/3/library/ast.html
[4]	Docker, Docker Engine security: namespaces, cgroups, capabilities and daemon/configuration risks.	https://docs.docker.com/engine/security/
[5]	Docker, Seccomp security profiles for Docker.	https://docs.docker.com/engine/security/seccomp/
[6]	Docker, none network driver.	https://docs.docker.com/engine/network/drivers/none/
[7]	Josefsson and Liusvaara, RFC 8032: Edwards-Curve Digital Signature Algorithm (EdDSA), including Ed25519.	https://www.rfc-editor.org/rfc/rfc8032.html
[8]	The Update Framework, roles and metadata for secure update trust and revocation concepts.	https://theupdateframework.io/docs/metadata/
[9]	in-toto, software supply-chain integrity and provenance framework.	https://in-toto.io/
[10]	AMD, Secure Encrypted Virtualization (SEV/SEV-SNP) confidential-computing overview and attestation material.	https://www.amd.com/en/developer/sev.html

Repository Evidence Record and Final Non-Claim

Repository anchors identify the exact evidence used for current capability claims. They do not convert future grant work or undeployed enterprise controls into production acceptance.

EVIDENCE	ANCHOR
Grant roadmap	docs/grants/EVIDENCEBOUND_MAS_COSMOS_2026.md; Issue #466 - EvidenceBound-MAS Cosmos 90-day public prototype.
Core package	packages/evidencebound_core/ - request models, source policy, formal verification, deterministic execution, semantic verification, Proof Pack and schemas.
Cryptographic seal contract	docs/evidencebound/CRYPTOGRAPHIC_SEAL_CONTRACT.md and proof-pack manifest schema; open-core merge 2719e078f9573049483e1f7f88041bc0b6622752.
Enterprise boundary	packages/evidencebound_enterprise/ and deploy/evidencebound-worker/; enterprise merge 319452ffb832853ec12c845b9286b250dfd87dcc.
Exact-head QA	ca3739d1d3c9a5465ce9b6e8464276d0e9642ecb - enterprise/core gates, Ruff, strict mypy, pytest, schema/audit drift, cross-stack gates and current-tree/full-history secret scans.
Public review surface	https://signalreview.co/proof-center and https://signalreview.co/proof-center#proof-scope.
Commercial supporting annex	EvidenceBound-MAS Commercialization and Deployment Strategy - 14-Day Enterprise PoC Protocol and Proposed Service Level Framework. Targets remain conditional until measured and contracted.

FINAL NON-CLAIM This document is a technical whitepaper and grant roadmap. It is not a security certification, legal opinion, regulatory approval, service-level agreement, production acceptance record, guarantee of model factuality, or universal theorem of software correctness.

Prepared by Ruslan Vrublevskyi | Kyiv, Ukraine | moneyparking@gmail.com | https://signalreview.co | https://github.com/moneyparking