THINKING OS™

SEAL External Proof Packet

Unauthorized Filing Refusal at the Final Submit Boundary
Evaluator-visible proof of a real pre-execution governance control

Public • Redacted • v3.0 • July 2026

Doc ID: TOS-SEAL-EPP-v3.0-7j5k6l

Document: SEAL External Proof Packet
Version: v3.0 (Public • Redacted) – July 2026
Owner: Thinking OS™
Use: May be shared externally. For information only; not legal advice.

Narrow Claim

This packet is designed to let legal, risk, insurer, regulator-facing, and technical evaluators assess one narrow claim:

SEAL is a pre-execution authority gate that sits in front of the final filing step and returns approve, refuse, or supervised outcomes before a motion or submission leaves the firm.

For each governed outcome, SEAL produces a reviewable decision artifact. This packet shows evaluator-visible behavior and evidence surfaces for that workflow. It does not expose non-public runtime details.

What This Packet Is For

This is not a product brochure, architecture guide, or diligence binder. It is a proof packet for one governed workflow:

wrong-authority filing refusal at the final submit boundary.

Its purpose is to let a serious buyer or evaluator determine, from the outside, whether:

- the runtime is real
- refusal behavior is real
- decision artifacts are real
- the downstream gate behavior is demonstrable for the scoped workflow under the configured evaluator conditions
- SEAL is a runtime control, not a chat wrapper, dashboard, or policy slideshow

What You Can Verify in This Packet

Inside this packet, you will see:

- why the Commit Layer exists for high-risk legal actions
- where SEAL sits in a wired legal workflow
- one SEAL-generated approval artifact
- three SEAL-generated refusal artifacts from different families
- one supervised / override outcome
- one execution receipt set showing runtime behavior and artifact linkage
- one scoped non-bypassability proof page
- one page on who owns the rules, the runtime, and the proof
- one page on what SEAL does not do

Disclaimer: *For information only. Not legal advice. This packet is a public evaluation reference for one narrow governed workflow.*

Proof Provenance & Evaluation Boundary

This packet is a public, redacted proof packet for one scoped SEAL Legal Runtime workflow:

Unauthorized / wrong-authority filing refusal at the final-submit boundary.

It is designed to show evaluator-visible runtime behavior, governed outcomes, decision artifacts, execution receipt linkage, and scoped non-bypassability evidence for that workflow. It does not expose non-public runtime internals, secrets, customer data, or production infrastructure details.

Evidence Source

The exhibits are redacted outputs from a Thinking OS™ internal / evaluator-controlled runtime environment configured to demonstrate the scoped workflow.

The packet may include:

- governed request examples
- approve, refuse, and supervised / override outcomes
- sealed decision artifacts
- audit trace references
- policy and rule-basis references
- integrity / review references
- execution receipt examples
- downstream handling or rejection evidence where shown

What This Packet Shows

For the scoped workflow and configured test conditions, this packet shows that SEAL can:

- receive a structured governed-action request before the final-submit boundary
- evaluate the request against configured role, authority, consent, workflow, and policy conditions
- return approve, refuse, or supervised outcomes
- produce reviewable decision artifacts
- preserve linkage between request, decision, artifact, and execution / handling evidence where configured
- reject direct downstream attempts where the demo workflow requires a valid governed path

What This Packet Does Not Show

This packet does not prove:

- customer-specific deployment readiness
- correctness of any specific law firm's policy
- correctness of any jurisdiction-specific legal rule
- production non-bypassability inside a buyer's environment
- that SEAL replaces lawyer judgment, court rules, GRC, identity systems, DMS, filing tools, or matter systems
- that any buyer has deployed SEAL in production

Redactions and Evaluation Note

The public packet redacts or excludes runtime internals, secrets and credentials, tenant-sensitive identifiers, operational contact details, live internal destinations, security-sensitive configuration, non-public policy internals, and customer or prospect information.

This packet should be evaluated as proof of scoped runtime behavior, not as a representation of a completed customer production deployment. A production or enforcement deployment requires separate client-specific scoping, authority mapping, integration review, security review, and written agreement.

For information only. Not legal advice.

Table of Contents:

Why Existing Controls Stop Before the Firm Is Committed

Most firms already have important controls around legal work: identity systems, GRC, matter systems, document systems, filing tools, workflow tools, AI gateways, and audit logs.

Those controls matter. But most govern **around** the action: who can log in, what policy says, what document exists, what workflow step occurred, what the AI tool produced, or what happened afterward.

SEAL governs a narrower point:

Before the filing leaves the firm, is this actor authorized to take this action, in this matter, under this authority, with the required evidence?

That is the Commit Layer: the pre-execution authority gate before the institution is committed.

Existing control	What it controls well	Where it stops	Why the Commit Layer is still needed
IAM / SSO	Who can access systems	Does not prove authority for a specific filing or action	Valid login is not valid authority to bind
GRC / policy	Written rules, control ownership, audit posture	Usually not inline at final-submit	Policy must become a runtime verdict
Matter system	Matter, client, workflow context	Does not usually approve or refuse execution	Context must connect to authority before action
DMS	Document access, versioning, storage	Controls the file, not the act of filing or sending	The risk is often the commitment, not the document
Filing / workflow tools	Local process steps and submission flow	Platform-specific, inconsistent across systems	The firm needs one authority surface across workflows
AI gateway	AI interaction, prompt/output/model-call governance	Controls AI interaction, not institutional commitment	Safe output can still drive unauthorized action
Audit / logging	What happened after the fact	Cannot stop the action	A record after harm is not refusal before harm
Commit Layer / SEAL	Runtime approve / refuse / supervised override	Only governs workflows wired through it	Provides an authority decision before the firm is committed

Existing systems govern around the action. The Commit Layer governs the action boundary itself.

What SEAL Governs

SEAL governs the **action boundary**, not a model, prompt, document, or dashboard.

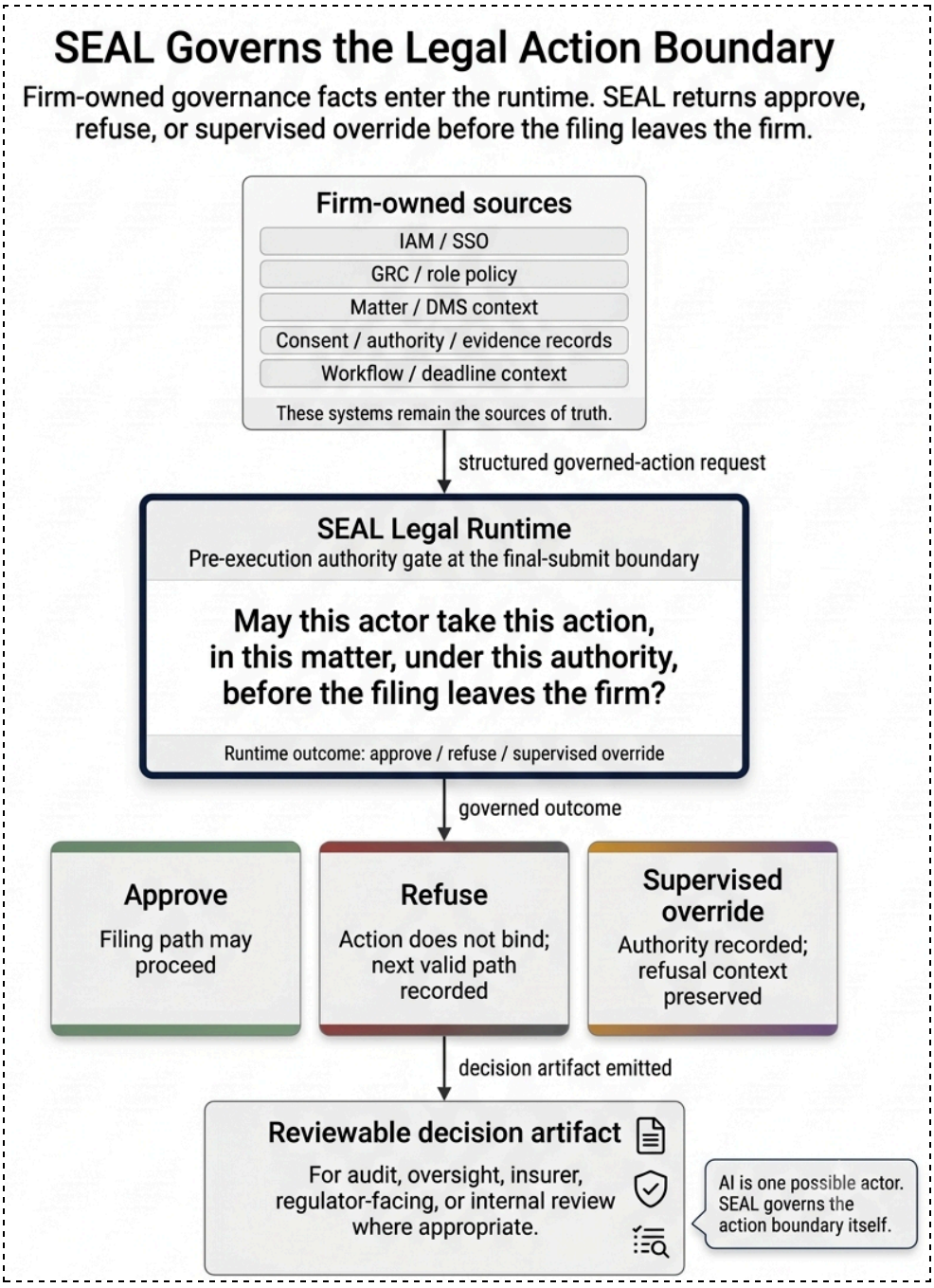
The actor may be a lawyer, staff member, service account, workflow automation, AI agent, script, or integrated system. The control question does not change:

May this actor take this legal action, in this matter, under this authority, before the filing leaves the firm?

SEAL receives a bounded governed-action request from firm-owned systems, evaluates it at the final-submit boundary, and returns one governed outcome:

Approve. Refuse. Supervised override.

Each outcome produces a reviewable decision artifact showing what the control did at the moment of action.



SEAL is not another application for lawyers to log into and not a dashboard beside the workflow. For designated governed paths, firm-owned systems provide the governance facts, SEAL evaluates the action at the final-submit boundary, and downstream filing or workflow actions proceed only from the configured governed outcome.

Proof of Approval Under Governed Conditions

This exhibit shows a governed approval outcome for the scoped workflow: wrong-authority filing refusal at the final-submit boundary. The request involved a motion-to-extend-time filing path evaluated under configured role, authority, consent, matter, and policy conditions before the filing path proceeded. SEAL approved the request before the filing path proceeded and produced a reviewable decision artifact with sensitive values redacted. The exhibit is generated from a synthetic, evaluator-controlled request and shown with public redactions only.

The proof point is narrow: approval was a governed commit, not an ungoverned pass-through.

Artifact Format	v1.2
Runtime Decision Contract	v1
Generated By	SEAL Runtime
Runtime Mode	Enforce

Executive Summary — Evaluator View

Display-only hierarchy of upstream governance/runtime metadata. Detailed evidence records follow below.

PLAIN-ENGLISH CONTROL MEANING

Governance approved and required downstream execution completed successfully.

1. Outcome

DECISION
Approved

DID THE ACTION BIND?
Yes

WHY
Governance approved and required downstream execution completed successfully.

2. Governance vs Runtime

Layer	Result
Governance Decision	Approval
Runtime Mode	Enforce
Execution	Attempted / succeeded
Binding Effect	Allowed

3. Authority / Reason

AUTHORITY BASIS
Policy: SEAL-APP-ALLOWED-001 • Policy set: [redacted] • Role: Attorney • Consent: issuer=[redacted], subject=user-123 • Deadline: [redacted]; approx. 48.00 hours remaining

CONTROL MEANING
Governance approved and required downstream execution completed successfully.

4. Next Valid Path — Tenant-Owned Handoff

TENANT-OWNED NEXT STEP
Approval and execution are recorded in this artifact. The firm owns post-approval recordkeeping, review, retention, and downstream workflow.

SEAL records the governed approval and execution receipt; the firm owns matter recordkeeping, downstream workflow, review, and retention.

5. Evidence Anchors

ARTIFACT ID
[redacted]

DECISION ID
[redacted]

GOVERNANCE APPROVAL HASH
[redacted]

TIMESTAMP
[redacted]

DECLARED / NOT DECLARED

"Not Declared" means the value was not provided to SEAL at runtime. SEAL did not infer or backfill it.

Evaluator View — Approval Executive Summary

SEAL-generated runtime artifact from a synthetic, evaluator-controlled request, shown with public redactions only. This view below shows the approval outcome, enforcement posture, execution result, binding effect, firm-owned handoff, and redacted evidence anchors. The narrow proof point: when configured authority conditions are satisfied, approval is recorded before the filing path proceeds.

Execution Record — Approved, Enforced, Succeeded, Aligned

Runtime Outcome & Execution Record

Runtime and execution proof record. Values are rendered from upstream runtime/executor metadata; this notifier does not infer approval, execution success, gate status, binding effect, or decision alignment.

Runtime decision path

Governance, pre-execution runtime, final runtime, and alignment facts as supplied by upstream governance/runtime layers.

Governance Decision	approval
Pre-Execution Runtime Decision	approval
Final Runtime Decision	approval
Decision Alignment	aligned

Execution receipt

Receipt-only execution metadata. This notifier renders the supplied execution capsule and does not call or operate downstream systems.

Status	ok
Attempted	True
OK	True
Required	True
Effective Enforcement	True
Execution Outcome	succeeded
Execution Receipt Alignment	aligned
Blocks Effective Completion	False
Control Meaning	Governance approved and required downstream execution completed successfully.

Control meaning

Plain-language runtime/control meaning supplied by upstream runtime or governance metadata when present.

Plain-Language Control Meaning	Governance approved and required downstream execution completed successfully.

This execution view confirms the approval did not remain a policy note or dashboard event. Governance decision, final runtime decision, execution outcome, and execution alignment all point to the same result: approved, enforced, succeeded, aligned. Public redactions remove endpoint, routing, and infrastructure details while preserving evaluator-visible proof of controlled execution.

How to Read This Approval Artifact

This artifact should be read as evidence of runtime authority control, not legal judgment. SEAL did not decide whether the filing was strategically wise, legally sufficient, or jurisdictionally correct. It evaluated whether configured authority conditions were satisfied before the filing path proceeded.

Approval is not the absence of governance. It is a governed commit before the firm is bound.

What to verify

1. **Governed request —** the filing-path action was presented to SEAL before execution.
2. **Approval outcome** — SEAL returned approval under the configured authority posture.
3. **Runtime alignment** — the governance decision and runtime decision matched.
4. **Execution result** — the downstream action completed after the governed approval.
5. **Evidence surface** — reviewable evidence references exist, with sensitive values redacted.

Proof of Refusal Before Harm

Scenario

The following exhibits show three governed refusal families for the scoped final-submit workflow:

1. wrong authority / role not authorized
2. role disallowed / structurally barred actor
3. missing consent / authority condition not satisfied

In each case, the request reached SEAL before the filing path proceeded. SEAL returned refusal in enforce mode, the action did not bind, and the artifact preserved the refusal reason, next valid path, and redacted evidence anchors.

What SEAL returned

SEAL returned sealed refusal artifacts before the filing path proceeded. Each artifact shows the configured refusal family, runtime posture, binding effect, next valid path, and redacted evidence-reference categories.

Why it matters

This is the wedge in concrete form: wrong actor, wrong authority, or missing required authority evidence is refused before external exposure. SEAL does not merely warn, annotate, or reconstruct later. It owns the runtime “no” before the firm is committed.

Caption

This exhibit shows governed refusals produced before the filing path proceeded. Each artifact records what was attempted, which configured policy context applied, why the action was blocked, and which evidence-reference categories were preserved.

Refusal A — Wrong authority / role not authorized

Artifact Format	v1.2
Runtime Decision Contract	v1
Generated By	SEAL Runtime
Runtime Mode	Enforce

Executive Summary — Evaluator View

Display-only hierarchy of upstream governance/runtime metadata. Detailed evidence records follow below.

Plain-English Control Meaning

Governance refused this action because Role 'General Counsel' is not authorized for the Administrative Law vertical under the configured role policy.

1. Outcome

DECISION
Refused

DID THE ACTION BIND?
No

WHY
Governance refused this action because Role 'General Counsel' is not authorized for the Administrative Laws vertical under the configured role policy.

2. Governance vs Runtime

Layer	Result
Governance Decision	Refusal
Runtime Mode	Enforce
Final Runtime Outcome	Refusal / did not bind
Decision Alignment	Aligned
Binding Effect	No

Refusal-first outcome: this table separates governance decision, runtime posture, final runtime outcome, alignment, and binding effect.

3. Authority / Refusal Reason

AUTHORITY BASIS
Refusal code: SEAL-ROLE-NOT-AUTHORIZED • Policy set: [redacted] • Role: General Counsel • Deadline: [redacted] approx. 48.00 hours remaining

CONTROL MEANING
Governance refused this action because Role 'General Counsel' is not authorized for the Administrative Laws vertical under the configured role policy.

4. Next Valid Path — Tenant-Owned Handoff

TENANT-OWNED NEXT STEP
Provide authorized role or firm owned review metadata for governed resubmission — Owner: Tenant identity role mapping or supervisory review owner — Required: trusted reviewer identity or trusted claims, legal role authorized for this vertical/action under the configured policy, Firm owned supervisory review metadata when firm policy permits review of this refusal category, resubmission through the governed path only after role authorization, role mapping, or Firm owned supervisory review metadata is accepted by the governed path

SEAL records the governed next path; the firm owns remediation, review, escalation, record-keeping, and workflow.

5. Evidence Anchors

ARTIFACT ID
[redacted]

DECISION ID / CAPSULE ANCHOR
[redacted]

AUDIT TRACE ID
[redacted]

GOVERNANCE REFUSAL HASH
[redacted]

TIMESTAMP
[redacted]

Declared / Not Declared

"Not Declared" means the value was not provided to SEAL at runtime. SEAL did not infer or backfill it.

Scenario

A governed filing request was submitted under the strict runtime path for **Administrative Law / Motion To Extend Time**. The acting role resolved to **General Counsel**, but that role was not configured as an authorized filing role for this governed motion under the tenant's policy.

What SEAL returned

SEAL returned a sealed refusal with code **SEAL-ROLE-NOT-AUTHORIZED**, plus an audit trace reference, policy reference, rule-basis evidence, redacted evidence and integrity references

Why it matters

This shows that SEAL enforces workflow-specific authority, not job-title prestige. Even a senior legal role is blocked when it is outside the firm's configured authorization scope for the action at issue.

Bottom line

Seniority is not authority. SEAL checks whether the actor is authorized for this governed action under the configured rules.

Generated from a synthetic, evaluator-controlled request and redacted for public review. Control outcome and evidence-reference categories are preserved; sensitive values, routing details, operational contacts, and live destinations are redacted.

Refusal B — Role disallowed / structurally barred actor

Artifact Format	v1.2
Runtime Decision Contract	v1
Generated By	SEAL Runtime
Runtime Mode	Enforce

Executive Summary — Evaluator View

Display-only hierarchy of upstream governance/runtime metadata. Detailed evidence records follow below.

Plain-English Control Meaning

Governance refused this action because Role 'Paralegal' is explicitly disallowed under the Administrative Law vertical policy.

1. Outcome

DECISION
Refused

DID THE ACTION BIND?
No

WHY
Governance refused this action because Role 'Paralegal' is explicitly disallowed under the Administrative Laws vertical policy.

2. Governance vs Runtime

Layer	Result
Governance Decision	Refusal
Runtime Mode	Enforce
Final Runtime Outcome	Refusal / did not bind
Decision Alignment	Aligned
Binding Effect	No

Refusal-first outcome: this table separates governance decision, runtime posture, final runtime outcome, alignment, and binding effect.

3. Authority / Refusal Reason

AUTHORITY BASIS
Refusal code: SEAL-ROLE-DISALLOWED • Policy set: [redacted] • Role: Paralegal • Deadline: [redacted] approx. 48.00 hours remaining

CONTROL MEANING
Governance refused this action because Role 'Paralegal' is explicitly disallowed under the Administrative Laws vertical policy.

4. Next Valid Path — Tenant-Owned Handoff

TENANT-OWNED NEXT STEP
Rerun with authorized role or firm owned review metadata — Owner: Tenant identity role mapping or supervisory review owner — Required: trusted reviewer identity or trusted claims, legal role authorized for this action under the configured vertical policy, Firm owned supervisory review metadata when firm policy permits review of this refusal category, resubmission through the governed path only after role authorization, role mapping, or Firm owned supervisory review metadata is accepted by the governed path

SEAL records the governed next path; the firm owns remediation, review, escalation, record-keeping, and workflow.

5. Evidence Anchors

ARTIFACT ID
[redacted]

DECISION ID / CAPSULE ANCHOR
[redacted]

AUDIT TRACE ID
[redacted]

GOVERNANCE REFUSAL HASH
[redacted]

TIMESTAMP
[redacted]

Declared / Not Declared

"Not Declared" means the value was not provided to SEAL at runtime. SEAL did not infer or backfill it.

Scenario

A governed legal action was submitted under the strict runtime path for **Administrative Law / Motion To Compel Production**. The acting role resolved to **paralegal** from trusted identity and group context.

What SEAL returned

SEAL returned a sealed refusal with code **SEAL-ROLE-DISALLOWED**, plus an audit trace reference, client policy reference, rule-basis evidence, redacted evidence and integrity references

Why it matters

This proves refusal is not limited to unknown identity or missing mapping. Even when identity is known and mapped, SEAL still refuses when the mapped role is structurally barred under the configured policy.

Caption

This exhibit shows a governed refusal produced before the action left the firm. The artifact records who attempted the action, what was attempted, which policy context applied, and why the action was blocked.

Generated from a synthetic, evaluator-controlled request and redacted for public review. Control outcome and evidence-reference categories are preserved; sensitive values, routing details, operational contacts, and live destinations are redacted.

Refusal C — Missing consent / authority condition not satisfied

Artifact Format	v1.2
Runtime Decision Contract	v1
Generated By	SEAL Runtime
Runtime Mode	Enforce

Executive Summary — Evaluator View

Display-only hierarchy of upstream governance/runtime metadata. Detailed evidence records follow below.

Plain-English Control Meaning

Governance refused this action because No valid client consent record or consent token-of-record was declared for this matter and motion.

1. Outcome

DECISION
Refused

DID THE ACTION BIND?
No

WHY
Governance refused this action because No valid client consent record or consent token-of-record was declared for this matter and motion.

2. Governance vs Runtime

Layer	Result
Governance Decision	Refusal
Runtime Mode	Enforce
Final Runtime Outcome	Refusal / did not bind
Decision Alignment	Aligned
Binding Effect	No

Refusal-first outcome: this table separates governance decision, runtime posture, final runtime outcome, alignment, and binding effect.

3. Authority / Refusal Reason

AUTHORITY BASIS
Refusal code: SEAL-CONSENT-001 • Policy set: [redacted] • Role: Attorney • Consent status: missing_or_unknown • Deadline: [redacted] approx. 48.00 hours remaining

CONTROL MEANING
Governance refused this action because No valid client consent record or consent token-of-record was declared for this matter and motion.

4. Next Valid Path — Tenant-Owned Handoff

TENANT-OWNED NEXT STEP
Attach consent record or token for governed resubmission — Owner: Tenant owned matter team or consent system — Required: client consent record or consent token, matching matter / motion scope, resubmission through the governed path with updated consent evidence

SEAL records the governed next path; the firm owns remediation, review, escalation, record-keeping, and workflow.

5. Evidence Anchors

ARTIFACT ID
[redacted]

DECISION ID / CAPSULE ANCHOR
[redacted]

AUDIT TRACE ID
[redacted]

GOVERNANCE REFUSAL HASH
[redacted]

TIMESTAMP
[redacted]

Declared / Not Declared

"Not Declared" means the value was not provided to SEAL at runtime. SEAL did not infer or backfill it.

Scenario

A governed filing request was submitted under the strict runtime path for **Administrative Law / Motion To Extend Time**. The actor, motion, and timing posture were otherwise valid, but the consent condition required for that governed filing was not satisfied at the time of action.

What SEAL returned

SEAL returned a sealed refusal with code **SEAL-CONSENT-001**, plus an audit trace reference, policy reference, redacted evidence and integrity references

Why it matters

This proves refusal is not limited to role or authority mapping. SEAL also refuses when a required consent or authority condition is missing before the filing path proceeds.

Caption

This exhibit shows a governed refusal produced before the filing left the firm. The artifact records who attempted the action, what was attempted, which policy context applied, and why the request was blocked. Refusals are governed runtime outcomes tied to the firm's configured rules and inputs, not post-hoc opinions.

Generated from a synthetic, evaluator-controlled request and redacted for public review. Control outcome and evidence-reference categories are preserved; sensitive values, routing details, operational contacts, and live destinations are redacted.

Proof of Supervised Override

SEAL does not collapse every governed action into allow-or-block. Where configured, a baseline refusal can be routed into supervised override with named authority and reviewable evidence.

This exhibit shows a SEAL-generated runtime artifact from a synthetic, evaluator-controlled request. The baseline governance decision refused the action, but a supervised override was recorded and bound to the original refusal before the filing path proceeded.

The proof point is narrow: supervised override is not silent bypass. It is a governed exception with authority evidence before the firm is bound.

Artifact Format	v1.2
Runtime Decision Contract	v1
Generated By	SEAL Runtime
Runtime Mode	Enforce

Executive Summary — Evaluator View

Display-only hierarchy of upstream governance/runtime metadata. Detailed evidence records follow below.

PLAIN-ENGLISH CONTROL MEANING

Baseline governance refused the action, but runtime approved it under supervisory override, and required downstream execution completed successfully.

1. Outcome

DECISION
Supervised override — approved
DID THE ACTION BIND?
Yes
WHY
Baseline governance refused the action, but runtime approved it under supervisory override, and required downstream execution completed successfully.

1A. Supervised Override Evidence

PARENT REFUSAL
SEAL-ROLE-NOT-AUTHORIZED • Parent decision: [redacted]
OVERRIDE ACTOR
[redacted]
AUTHORITY RECORDED
Yes
SIGNED AUTHORITY REQUIRED
Yes
SIGNED AUTHORITY RECORDED
Yes
BOUND TO PARENT DECISION
Yes (upstream parent reference supplied)
OUTCOME
Approved through supervised override

Display-only summary of the upstream override_contract. Detailed Formal Override Path record appears below.

2. Governance vs Runtime

Layer	Result
Baseline Governance Decision	Refusal
Runtime Mode	Enforce
Final Runtime Outcome	Approval
Decision Alignment	Diverged
Binding Effect	Allowed

3. Authority / Reason

AUTHORITY BASIS
Policy: SEAL-OVERRIDE-APPROVED • Policy set: [redacted] Role: Partner • Consent: issuer=[redacted] subject=[redacted] • Deadline: [redacted]; approx. 48.00 hours remaining • Override: approved
CONTROL MEANING
Baseline governance refused the action, but runtime approved it under supervisory override, and required downstream execution completed successfully.

4. Next Valid Path — Tenant-Owned Handoff

TENANT-OWNED NEXT STEP
Approval and execution are recorded in this artifact. The firm owns post-approval recordkeeping, review, retention, and downstream workflow.

SEAL records the governed approval and execution receipt; the firm owns matter recordkeeping, downstream workflow, review, and retention.

5. Evidence Anchors

ARTIFACT ID
[redacted]
DECISION ID
[redacted]
GOVERNANCE APPROVAL HASH
[redacted]
TIMESTAMP
[redacted]

DECLARED / NOT DECLARED

"Not Declared" means the value was not provided to SEAL at runtime. SEAL did not infer or backfill it.

Evaluator View — Supervised Override Executive Summary

SEAL-generated runtime artifact from a synthetic, evaluator-controlled request, shown as generated with public redactions only. This view below shows the baseline refusal, supervised override approval, enforcement posture, divergent decision alignment, allowed binding effect, firm-owned handoff, and redacted evidence anchors. "Diverged" means the baseline refusal was preserved while the final runtime outcome changed only through the configured supervised override path.

Outcome — Supervised Override: Refusal Preserved, Authority Recorded

Formal Override Path

Display-only rendering of the upstream override_contract capsule. Overrides only count when the signed authority / callback path is recorded and bound to the refused parent decision. SEAL does not approve, override, create tickets, or operate tenant workflow from this artifact.

Override contract outcome

Upstream-supplied override_contract envelope and terminal override outcome.

Override Attempted	true
Override Outcome	approved

Refused-parent linkage

Parent and baseline decision references echoed from the override contract. This notifier does not validate linkage.

Baseline Refusal Code	SEAL-ROLE-NOT-AUTHORIZED
Prior Code	SEAL-ROLE-NOT-AUTHORIZED

Authority and actor record

Authority, reviewer, and override-actor metadata echoed as supplied. Current reviewer is not substituted as override actor.

Override Kind	supervisory_override
Override Reason	Partner supervisory override after recorded baseline refusal. Formal Override Path is complete and bound to the same matter/docket.

Verification and recording posture

Rule-basis, authority, and signed-authority recording facts are rendered exactly as supplied by upstream runtime/governance metadata.

Rule-Basis Verified	true
Authority Verified	true
Signed Authority Required	true
Signed Authority Recorded	true

This override-path view shows that the action was not treated as an ordinary approval. The baseline refusal remained recorded, the override was bound to the refused parent decision, and supervisory authority was required, verified, and recorded. Public redactions remove identities, matter identifiers, decision anchors, routing, policy, and infrastructure details while preserving evaluator-visible proof of governed exception handling.

How to Read This Supervised Override Artifact

This artifact should be read as evidence of governed escalation, not legal judgment. SEAL did not decide whether the filing was strategically wise, legally sufficient, or jurisdictionally correct. It recorded that the baseline governance outcome was refusal, then allowed the action only after the configured supervised override path was satisfied.

A supervised override is not a bypass. It is a recorded exception with named authority before the firm is bound.

What to verify:

1. **Baseline refusal** — the original governance decision refused the action.
2. **Supervised override** — the final outcome changed only through the configured override path.
3. **Authority evidence** — supervisor authority was required and recorded.
4. **Parent linkage** — the override was bound to the original refusal decision.
5. **Evidence surface** — reviewable evidence references exist, with sensitive values redacted.

Scenario

A governed filing request for **Administrative Law / Motion for Summary Judgment** did not clear directly and was routed into supervised review. A supervising partner then submitted a linked override with documented supervisory authority and override evidence.

What SEAL returned

SEAL returned a supervised-override approval with a decision / trace reference, linked baseline-refusal context, and override metadata showing the supervising authority and basis for the override. Public evidence and integrity values are redacted.

Why it matters

This proves SEAL does not collapse governed actions into allow-or-block only. It preserves the original refusal, requires supervised authority, and records the override path before execution completes.

Generated from a synthetic, evaluator-controlled request and redacted for public review. This view shows baseline refusal, supervised override approval, enforce mode, allowed binding effect, firm-owned handoff, and redacted evidence anchors. "Diverged" means the original refusal was preserved while the final outcome changed only through the configured supervised override path.

Execution Receipts and Artifacts Tell the Same Story

This exhibit shows that SEAL is not merely producing a static artifact after the fact. A governed request reached the runtime, SEAL returned a pre-execution decision, a reviewable decision artifact was emitted, and the downstream execution receipt reflected the same governed outcome.

The proof point is narrow: the decision record and execution receipt align before the institution is committed.

Block 1 — Governed request received

A governed filing-path request entered the runtime with the declared role, workflow, stage, legal environment, and jurisdiction context needed for evaluation.

Core declared inputs
Client/runtime-supplied request context. SEAL renders these values as supplied.

Declared Role	Attorney
Declared Vertical	Administrative Law
Declared Scenario	Motion to Extend Time
Case Stage	Filing
Legal Environment / Workflow Input	Administrative Law
Requested Turnaround	Standard

Block 2 — Decision returned

SEAL returned approval at the pre-execution control point. The governance decision, pre-execution runtime decision, final runtime decision, and decision alignment all point to the same result: approved and aligned.

Approval Code	SEAL-APP-ALLOWED-001
Enforcement Posture	Enforce (runtime decisions are applied in-line with governance policy).
Governance Decision	approval
Pre-Execution Runtime Decision	approval
Final Runtime Decision	approval
Decision Alignment	aligned

Block 3 — Artifact emitted

The runtime emitted a reviewable decision artifact with artifact, decision, timestamp, and integrity-reference categories visible. Sensitive values are redacted.

Block 4 — Downstream handling event

The execution receipt shows downstream handling completed in alignment with the governed approval: approved, enforced, succeeded, aligned.

Runtime Outcome & Execution Record

Runtime and execution proof record. Values are rendered from upstream runtime/executor metadata; this notifier does not infer approval, execution success, gate status, binding effect, or decision alignment.

Runtime decision path

Governance, pre-execution runtime, final runtime, and alignment facts as supplied by upstream governance/runtime layers.

Governance Decision	approval
Pre-Execution Runtime Decision	approval
Final Runtime Decision	approval
Decision Alignment	aligned

Execution receipt

Receipt-only execution metadata. This notifier renders the supplied execution capsule and does not call or operate downstream systems.

Status	ok
Attempted	True
OK	True
Required	True
Execution Outcome	succeeded
Execution Receipt Alignment	aligned
Control Meaning	Governance approved and required downstream execution completed successfully.

Control meaning

Plain-language runtime/control meaning supplied by upstream runtime or governance metadata when present.

Plain-Language Control Meaning	Governance approved and required downstream execution completed successfully.

SEAL-generated runtime artifact from a synthetic, evaluator-controlled request, shown with public redactions only. This view shows that the governed request, runtime decision, and downstream execution receipt all resolve to the same control story: approved, enforced, succeeded, aligned. Public redactions remove endpoint, routing, policy, identity, and infrastructure details while preserving evaluator-visible proof of downstream handling.

Scoped Non-Bypassability for the Demo Workflow

For the scoped demo workflow, the final-submit path is governed. A downstream action proceeds only when the workflow receives a valid governed outcome from SEAL. This exhibit does not claim production non-bypassability in every buyer environment; it shows scoped non-bypassability under the configured demo workflow.

The proof point is narrow: when the workflow is wired through SEAL, the governed path succeeds and direct unguided attempts are rejected.

Block 1 — Approved path through SEAL

Artifact Format	v1.2
Runtime Decision Contract	v1
Generated By	SEAL Runtime
Runtime Mode	Enforce

Executive Summary — Evaluator View

Display-only hierarchy of upstream governance/runtime metadata. Detailed evidence records follow below.

PLAIN-ENGLISH CONTROL MEANING

Governance approved and required downstream execution completed successfully.

1. Outcome

DECISION
Approved

DID THE ACTION BIND?
Yes

WHY
Governance approved and required downstream execution completed successfully.

2. Governance vs Runtime

Layer	Result
Governance Decision	Approval
Runtime Mode	Enforce
Execution	Attempted / succeeded
Binding Effect	Allowed

3. Authority / Reason

AUTHORITY BASIS
Policy: SEAL-APP-ALLOWED-001 • Policy set: [redacted] • Role: Attorney • Consent: issuer=[redacted] subject=user-123 • Deadline: [redacted], approx. 48.00 hours remaining

CONTROL MEANING
Governance approved and required downstream execution completed successfully.

4. Next Valid Path — Tenant-Owned Handoff

TENANT-OWNED NEXT STEP
Approval and execution are recorded in this artifact. The firm owns post-approval recordkeeping, review, retention, and downstream workflow.

SEAL records the governed approval and execution receipt; the firm owns matter recordkeeping, downstream workflow, review, and retention.

5. Evidence Anchors

ARTIFACT ID
[redacted]

DECISION ID
[redacted]

GOVERNANCE APPROVAL HASH
[redacted]

TIMESTAMP
[redacted]

DECLARED / NOT DECLARED

"Not Declared" means the value was not provided to SEAL at runtime. SEAL did not infer or backfill it.

SEAL-generated runtime artifact from a synthetic, evaluator-controlled request, shown as generated with public redactions only. This view shows the valid governed path: approved, enforced, executed, allowed, with evidence references preserved and sensitive values redacted.

Approved path through SEAL

A governed filing request for Administrative Law / Motion To Extend Time passed through SEAL and reached the downstream execution path only after governed approval. The approval artifact and execution receipt share the same decision story: **approval code SEAL-APP-ALLOWED-001**, redacted decision / trace reference, execution outcome succeeded, and alignment aligned.

Block 2 — Direct attempt outside the governed path

A direct attempt to reach the downstream action path without a valid governed SEAL outcome was rejected by the configured demo workflow.

Check	Result
Governed SEAL outcome present	No
Downstream attempt accepted	No
Outcome	Rejected
Reason category	Invalid governed path

Block 3 — Repeat rejection under the same configured conditions

The same direct attempt was repeated and rejected again under the same configured conditions. This shows the demo workflow rejected unguided downstream attempts consistently, not as a one-off result.

Attempt	Governed outcome present	Result
Attempt 1	No	Rejected
Attempt 2	No	Rejected

Bottom takeaway

Within the configured demo workflow, valid downstream completion depends on a governed SEAL outcome. Direct attempts outside the governed path are rejected.

Who Owns the Rules, the Runtime, and the Proof

The firm owns	Thinking OS is responsible for
• policies and authority rules • identity / role sources • matter and workflow context • supervision model • workflow routing scope • retention, use, and disclosure of decision artifacts and matter records	• operation of the SEAL runtime within agreed scope • faithful enforcement of configured conditions • decision artifact generation • runtime security, availability, and isolation within agreed scope

Caption

The firm retains responsibility for policies, people, authority, workflow scope, and the use of resulting records. Thinking OS is responsible for operating the SEAL runtime within agreed scope, enforcing configured conditions, and producing reviewable decision artifacts. SEAL does not supply legal judgment or replace the firm's GRC, identity, matter, or filing systems.

What This Runtime Is Not

SEAL is a pre-execution authority gate for governed legal actions. It enforces firm-owned rules at the point of action and produces sealed decision artifacts. It does not replace legal judgment, attorney supervision, or the firm's own systems of record.

SEAL does not:

- draft, edit, sign, or file legal documents
- provide legal advice or choose litigation strategy
- replace lawyers, supervisors, or ethics counsel
- replace GRC, identity, matter, or case-management systems
- become the system of record for matters or filings
- determine legal merits, ethical obligations, or professional judgment

SEAL can refuse; it cannot file.

Approvals show that configured governance conditions were satisfied at the moment of action. They do not certify strategy, merits, ethics, or professional judgment. Attorneys remain responsible for all decisions, filings, and communications.

Public evaluation reference. Redacted.

Describes evaluator-visible behavior and evidence surfaces only; no non-public runtime details are included.

For information only. Not legal advice.