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 • v2.0 • May 2026

Doc ID: TOS-SEAL-EPP-v2.0-7j5k6l

Document: SEAL External Proof Packet
Version: v2.0 (Public • Redacted) – May 2026
Owner: Thinking OS™
Use: May be shared externally. For information only; not legal advice.

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 3 Minutes

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 in this packet 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 Exclusions

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
- customer or prospect information

Evaluation Note

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 This Layer Exists

Most institutional and AI governance stops too soon.

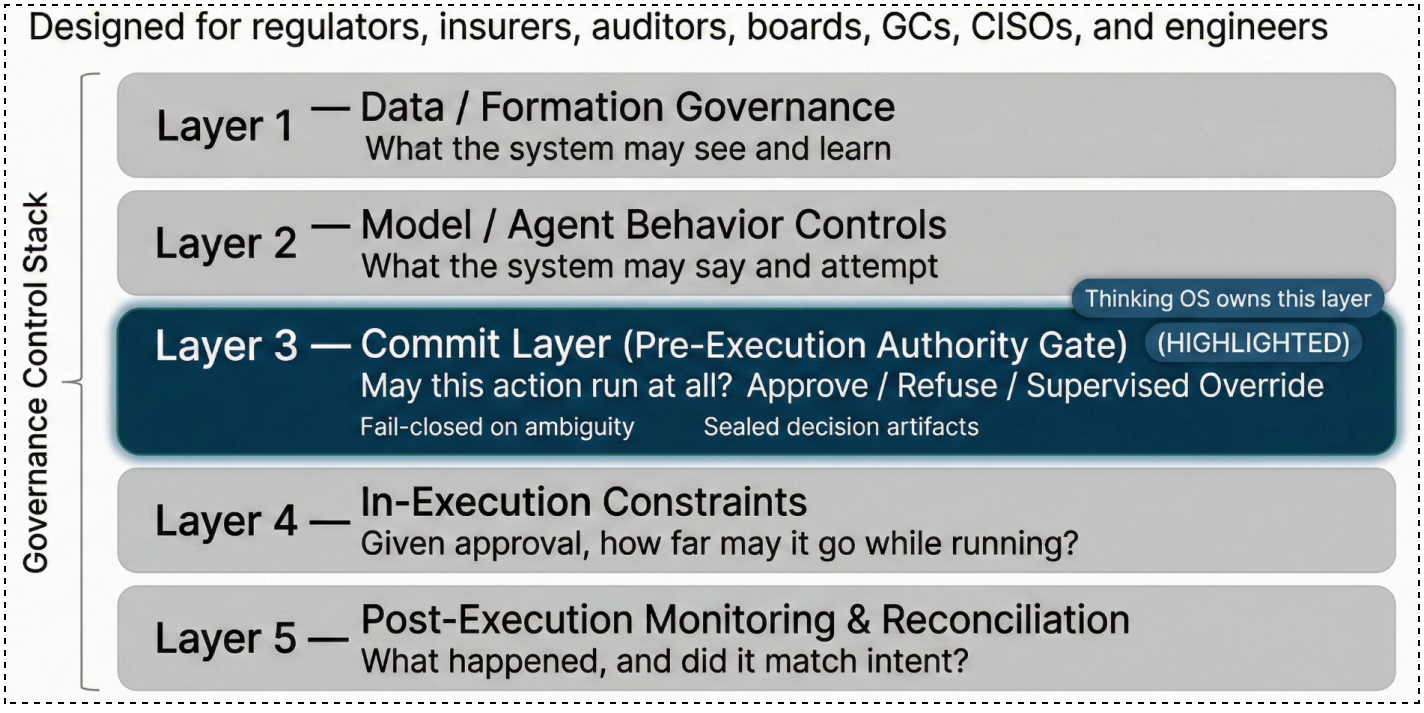
Organizations built data governance, model governance, and identity and access controls first. All of those layers matter. But they do not answer the one question that matters most at the moment of action:

For this actor, this action, right now — may it run at all?

That question belongs to the **Commit Layer**: the pre-execution authority gate in front of high-risk actions.

In legal workflows, that distinction matters because liability attaches when something is filed, sent, approved, or moved under the firm's name — not when a policy is written, a dashboard is reviewed, or a model is monitored later.

SEAL operates at that missing layer for governed legal workflows.



Most governance layers govern what a system may see, say, or how it is reviewed later. **SEAL operates at the missing Commit Layer:** the pre-execution authority gate in front of file / send / approve / move. It returns approve, refuse, or supervised outcomes before the action executes.

This is why SEAL is not ordinary guardrails or after-the-fact monitoring. It governs the moment before the action becomes real.

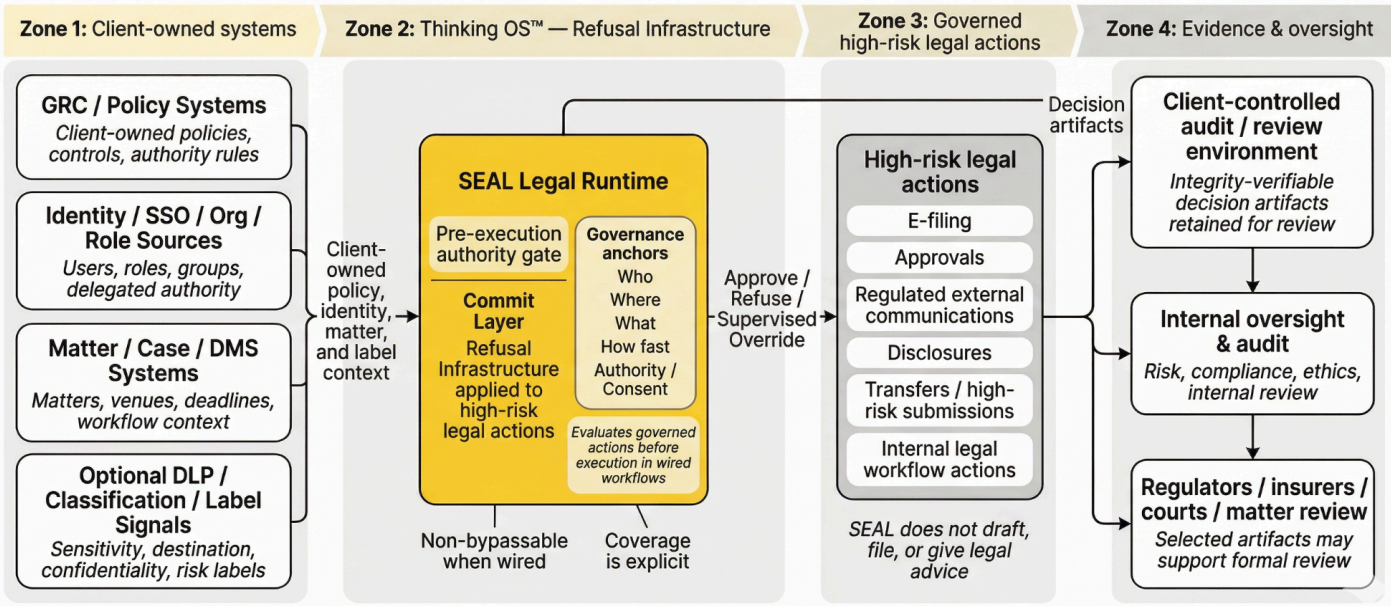
Where SEAL Sits in the Legal Stack

SEAL is not another application for lawyers to log into. It is a pre-execution control layer that sits between the firm's own policy, identity, matter systems and designated high-risk legal actions.

In a wired workflow, firm-owned systems provide the policy, identity, matter, and label context. SEAL evaluates that governed request at the Commit Layer and returns one of three outcomes: approve, refuse, or supervised override. Downstream tools and workflows proceed only from that governed outcome, and decision artifacts are produced for audit, oversight, and later review where appropriate.

This is the point of control in the workflow: not after the filing, not in a dashboard, and not as a policy reminder after the fact.

Figure 1 — Public Control Map: Where SEAL Sits in the Legal Stack
Action Governance at the Commit Layer for High-Risk Legal Actions



Conceptual control map for public evaluation. Describes evaluator-visible behavior and evidence surfaces, not non-public runtime details.

SEAL sits between firm-owned policy, identity, matter sources and designated high-risk legal actions. Your systems feed SEAL; your drafting and e-filing tools run only once SEAL has approved; and courts, regulators, insurers, and internal oversight functions can review the resulting decision artifacts where appropriate.

This page shows that SEAL is a control point in the workflow, not a dashboard or sidecar.

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 submitted under the firm's configured role, authority, consent, matter, and policy conditions. SEAL approved the request before the filing path proceeded and produced a reviewable decision artifact with public values redacted. This is a SEAL-generated runtime artifact from a synthetic, evaluator-controlled request, shown as generated with public redactions only.

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

Executive Summary — Evaluator View

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 • 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
Action approved and executed; preserve this artifact with the matter record and follow firm record-keeping procedures.

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

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 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, SEAL approves before the filing path proceeds and preserves a reviewable decision artifact.

Outcome 1 — Approval: Governance and Execution Aligned

Runtime Outcome & Execution Record

Governance Decision	approval
Pre-Execution Runtime Decision	approval
Baseline Governance Code	Not Declared
Final Runtime Decision	approval
Decision Alignment	aligned
Runtime Translation	Not Declared

Execution Receipt

Status	ok
Attempted	True
OK	True
Hard Fail	False
Required	True
Mode	enforce
Execution Outcome	succeeded
Governance / Execution Alignment	aligned
Blocks Effective Completion	False
Control Meaning	Governance approved and required downstream execution completed successfully.
Endpoint Name	[redacted]
Endpoint Attempted	True
Endpoint OK	True
Endpoint HTTP Status	[redacted]
Endpoint Error Code	Not Declared
Endpoint Error / Reason	ok

Executor Gate

Status	ok
Attempted	True
OK	True
Hard Fail	False
Required	True
Mode	enforce
Blocks Effective Completion	False
Reason Code	Not Declared
Reason	succeeded
Control Meaning	Governance approved and required downstream execution completed successfully.

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. The 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 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** — a high-risk filing-path action reached 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 public values redacted.

Proof of Refusal Before Harm

Scenario

A filing request was submitted for `administrative_law / motion_to_extend_time` under the strict runtime path. The motion, timing, and consent posture were otherwise valid, but the acting role did not resolve to an authorized legal role from trusted identity claims.

What SEAL returned

SEAL returned a sealed refusal before the filing path proceeded. The artifact preserved the refusal family, audit trace reference, policy reference, and redacted integrity reference.

Why it matters

This is the wedge in concrete form: wrong actor, wrong authority, blocked before external exposure. It shows that SEAL does not merely warn or annotate. It refuses the action at the governed boundary. Unknown or unmapped roles fail closed into governed refusal, not silent passage.

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

Executive Summary — Evaluator View

Plain-English Control Meaning

Governance refused this action because Role 'general_counsel' is not in authorized_roles for vertical 'administrative_law'.

1. Outcome

DECISION
Refused

DID THE ACTION BIND?
No

WHY
Governance refused this action because Role 'general_counsel' is not in authorized_roles for vertical 'administrative_law'.

2. Governance vs Runtime

Layer	Result
Governance Decision	Refusal
Runtime Mode	Enforce
Final Runtime Outcome	Refusal / did not bind
Decision Alignment	Not Declared
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
Policy: 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 in authorized_roles for vertical 'administrative_law'.

4. Next Valid Path — Tenant-Owned Handoff

TENANT-OWNED NEXT STEP
provide_recognized_identity_or_role_mapping_and_rerun — Owner: tenant_identity_or_role_mapping_owner — Link: [redacted] — Required: recognized reviewer identity or trusted claims, tenant-owned group-to-legal-role mapping, no conflicting payload identity fields, re-run request after identity or role-map signal is corrected

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

5. Evidence Anchors

ARTIFACT ID
[redacted]

DECISION 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.

SEAL-generated runtime artifact from a synthetic, evaluator-controlled request, shown as generated with public redactions only. Control outcome and evidence-reference categories are preserved; public values, routing details, operational contacts, and live destinations are redacted.

Refusal B — Role disallowed / structurally barred actor

Executive Summary — Evaluator View

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_law vertical policy.

2. Governance vs Runtime

Layer	Result
Governance Decision	Refusal
Runtime Mode	Enforce
Final Runtime Outcome	Refusal / did not bind
Decision Alignment	Not Declared
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
Policy: SEAL-ROLE-DISALLOWED • Policy set: [redacted] • Role: Paralegal

CONTROL MEANING
Governance refused this action because Role 'paralegal' is explicitly disallowed under the administrative_law vertical policy.

4. Next Valid Path — Tenant-Owned Handoff

TENANT-OWNED NEXT STEP
provide_recognized_identity_or_role_mapping_and_rerun — Owner: tenant_identity_or_role_mapping_owner — Link: [redacted] — Required: recognized reviewer identity or trusted claims, tenant-owned group-to-legal-role mapping, no conflicting payload identity fields, re-run request after identity or role-map signal is corrected

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

5. Evidence Anchors

ARTIFACT ID
[redacted]

DECISION 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.

Redacted public exhibit. Reviewer identity, matter-routing details, operational contact details, and live internal link destinations removed. Control outcome and evidence-reference categories are preserved; public values, routing details, operational contacts, and live destinations are redacted.

Refusal C — Missing consent / authority condition not satisfied

Executive Summary — Evaluator View

Plain-English Control Meaning

Governance refused this action because Client ID consent explicitly denied or structurally revoked.

1. Outcome

DECISION
Refused

DID THE ACTION BIND?
No

WHY
Governance refused this action because Client ID consent explicitly denied or structurally revoked.

2. Governance vs Runtime

Layer	Result
Governance Decision	Refusal
Runtime Mode	Enforce
Final Runtime Outcome	Refusal / did not bind
Decision Alignment	Not Declared
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
Policy: 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 Client ID consent explicitly denied or structurally revoked.

4. Next Valid Path — Tenant-Owned Handoff

TENANT-OWNED NEXT STEP
attach_consent_record_or_token_and_rerun — Owner: matter_team_or_client_consent_system — Link: [redacted] — Required: client consent record or consent token, matching matter / motion scope, re-run request with updated 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
[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.

Redacted public exhibit. Reviewer identity, matter-routing details, operational contact details, and live internal link destinations removed. Control outcome and evidence-reference categories are preserved; public values, routing details, operational contacts, and live destinations are redacted.

Proof of Supervised Escalation

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.

Executive Summary — Evaluator View

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-ADVISORY-001 • 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 • Deadline: safe lane: Tenant Owned Supervisor Review • 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
Action approved and executed; preserve this artifact with the matter record and follow firm record-keeping procedures.
SEAL records the governed next path; the firm owns remediation, review, escalation, record-keeping, and workflow.

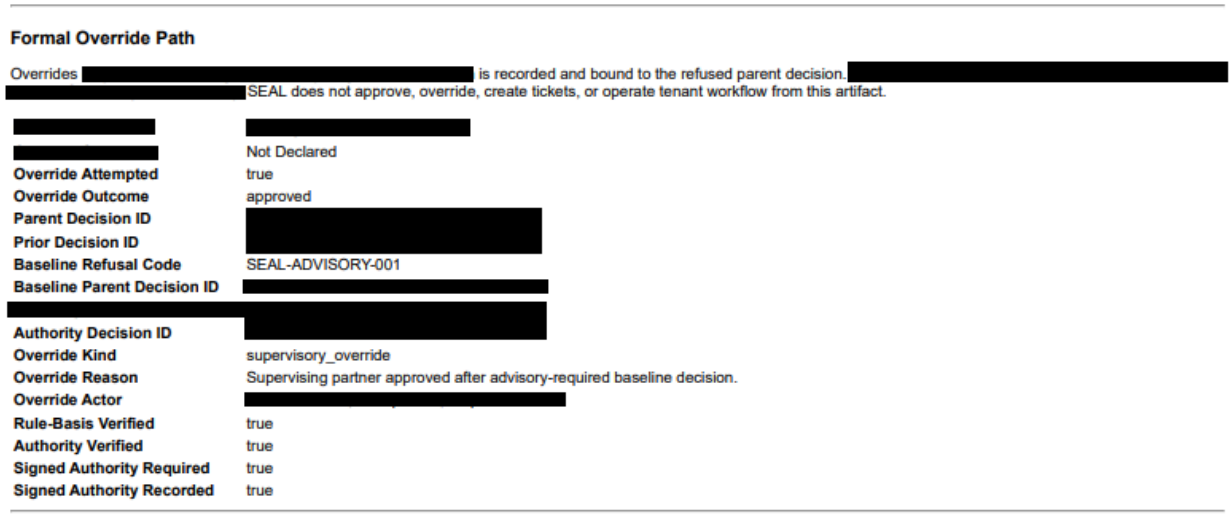
5. Evidence Anchors

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 baseline refusal, supervised override approval, enforcement posture, divergent decision alignment, allowed binding effect, firm-owned handoff, and redacted evidence anchors.

Outcome — Supervised Override: Refusal Preserved, Authority Recorded



Formal Override Path

Overrides [redacted] is recorded and bound to the refused parent decision. [redacted]
[redacted] SEAL does not approve, override, create tickets, or operate tenant workflow from this artifact.

[redacted]	[redacted]
[redacted]	Not Declared
Override Attempted	true
Override Outcome	approved
Parent Decision ID	[redacted]
Prior Decision ID	
Baseline Refusal Code	SEAL-ADVISORY-001
Baseline Parent Decision ID	[redacted]
[redacted]	
Authority Decision ID	[redacted]
Override Kind	supervisory_override
Override Reason	Supervising partner approved after advisory-required baseline decision.
Override Actor	[redacted]
Rule-Basis Verified	true
Authority Verified	true
Signed Authority Required	true
Signed Authority Recorded	true

This override-path view shows why the action was not treated as an ordinary approval. The baseline refusal remained recorded, a supervised override was applied, authority was required and recorded, and the override was bound to the parent decision. Public redactions remove identity, authority identifiers, 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 public 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.

SEAL-generated runtime artifact from a synthetic, evaluator-controlled request, shown as generated with public redactions only. Control outcome and evidence-reference categories are preserved; public values, identity details, routing details, operational contacts, and live destinations are redacted.

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.

Declared Role	Attorney
Declared Vertical	Administrative Law
Declared Scenario	Motion to Extend Time
Case Stage	Filing
Legal Environment	Administrative Law
Requested Turnaround	Standard
Jurisdiction	US-VA

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. Public values are redacted.

Executive Summary — Evaluator View

5. Evidence Anchors

ARTIFACT ID [redacted]

DECISION ID [redacted]

GOVERNANCE APPROVAL HASH [redacted]

TIMESTAMP [redacted]

Block 4 — Downstream handling event

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

Runtime Outcome & Execution Record

Governance Decision	approval
Final Runtime Decision	approval
Decision Alignment	aligned
Execution Receipt	

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

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

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

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

Executive Summary — Evaluator View

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 • 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
Action approved and executed; preserve this artifact with the matter record and follow firm record-keeping procedures.

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

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 public 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 call attempt to commit target

A direct attempt to reach the downstream commit target 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 governance 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 on its own

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.