Put AI to work at full power. Stay in control of every action.

One approval, one action. Automatic within your limits, or given by the person you name.

RSBIS is designed to sit as a checkpoint in front of payments, releases and other high-risk systems, for people, software and AI. Each action needs its own signed, single-use approval; any change to that action is refused, and approvals are recorded.

The risk and the return

Why now. In 2026, attackers wiped a medical-device maker's computers with its own device-management tool,[1] and AI models under test reached real companies' systems.[2, 3] Authority must be checked where each action takes effect.

$123,000
Average reported loss per business email compromise complaint, 2025.[4] A later account change is refused.

$1 million+
1 in 5 operators' latest major outage cost more; 57% topped $100,000.[5] Unapproved releases are refused.

$1.2 million / yr
One workflow, illustrative: 10,000 routine actions a month safely handed to AI, 10 staff minutes each: 20,000 hours at $60.

How it works: controls built into every action, not added afterward

AUTHORITY Who may approve?
You set who may approve what, up to which amount, and which actions run on their own.

DECISION May this happen?
Before anything happens, a signed, single-use approval for one exact action. Any change to it is refused.

EVIDENCE Can others check it?
Every approval and its evidence, sealed in one tamper-evident trail; refusals and outcomes are planned.

In your control: the authority stays yours. **Traceable:** each action to its approval. **Checkable by others:** offline, from the record.

> **Strengthen the controls you already use.** Every system tells three stories: what was allowed, what was tried and what actually happened. Our published method, Actions Science, keeps them apart, reconciles them and shows where they don't match, and puts business, engineering, finance and GRC/security on the same rules and evidence; RSBIS enforces them and leaves one record your auditors and board can check. Each team keeps its decision rights, with fewer repeated approvals, handoffs and audit preparation.

Two examples. Endless uses.

BUSINESS ACTION	WHAT THE APPROVAL COVERS	BUSINESS VALUE
Supplier payment	$500 to Supplier A, account ending 4417, invoice 104. A changed account or a second payment is refused.	**Protects cash; less correction work.**
Production change	Test within set limits, then deploy only approved build 1.1 to Production. Build 1.2 or a second release is refused.	**Faster delivery; controlled change.**

The same approach applies across workflows, each action with its own approval: releases, refunds, payroll, prices, access grants, contracts, cloud changes and more. Try both examples at **rootzerovault.com/actions-science**.

Start with one action. Expand to every workflow.

Covers people, agents, teams and systems.

Recommendation: supplier payments first, then expand.
Accountability: every action traces to a person or rule.
For your board: proof each covered action was authorized.

Live demo online

1. MedTech Dive, Stryker attack and device-management tool, March 17, 2026, medtechdive.com. 2. OpenAI and Hugging Face, security incident during model evaluation, July 21, 2026, openai.com. 3. Anthropic, four cybersecurity-evaluation incidents, September 9, 2026, anthropic.com. 4. FBI, 2025 Internet Crime Report, April 2026: $3.05 billion across 24,768 complaints, ic3.gov. 5. Uptime Institute, Annual Outage Analysis 2026, May 2026, uptimeinstitute.com.

Root Zero Vault, Inc. Questions or a walkthrough: **deen_saleh@rootzerovault.com** · **rootzerovault.com**

EXAMPLE 2 · SUPPLIER PAYMENT

One pass, one action

Follow one $500 supplier payment from request to evidence.

The gate decides. The guard enforces.

Every system tells three stories: what was allowed, what was tried and what actually happened. Actions Science keeps them apart, reconciles them and shows where they don't match.

THE LINE OF AUTHORITY

1. **THE COMPANY**
   Holds its deed and its payment rules; two officers sign them.
2. **FINANCE OWNER AND PAYABLES CONTROLLER**
   Each appointed in writing; each approves the exact payment.
3. **PAYMENT AGENT · AI**
   Asks to pay matched invoices up to $1,000; moves no money on its own.
4. **THE BANK AND ITS PAYMENT GUARD**
   Under the bank's own terms; holds each payment as pending, then releases it once.

NOT ABOVE, NOT BELOW · ROOT ZERO VAULT

A neutral constitution, outside this line. Its thin law only protects that neutrality and each deed holder's sovereignty; it sets no rules for anyone here and never changes a holder's own rules.

SIX MOMENTS

1. **Onboard** SEPTEMBER
   Supplier A's account ending 4417 goes on the vendor file; a signed declaration of the systems the rules cover, a claim to test ●; proving that coverage needs separate checks ◐
   entries 1, 2
2. **Delegate** OCTOBER 1
   Two approvers are appointed; the agent gets a written scope; written delegation records, withdrawn at any time ●
   entries 3–5
3. **Invoice** OCTOBER 20
   Invoice 104 matches its order and goods receipt; signed for by Supplier A: $500 to account ending 4417 ◐
   entries 6, 7
4. **Pay once** OCTOBER 21
   Refused before the approvals, then held by the bank as pending ◐; distinct signers are counted: one signer identity never counts twice ●; a redirect to account ending 9021 fails ◐; paid once, confirmed by the bank ◐
   entries 8–22
5. **Restrict** OCTOBER 21
   A new payment to account ending 9021 is refused on the restricted route ◐; refusals signed too ◐
   entries 23, 24
6. **Check later** OCTOBER 22
   The reviewer rechecks the day; the gate's decisions rechecked offline ●; and matches the bank's own log ◐
   entries 25

THE REQUEST

The payment agent asks: pay $500 to Supplier A, account ending 4417, for invoice 104, once.

THE PASS

A signed authorization for this exact payment only, usable once, for two minutes.

THE METHOD

Every action follows the ten steps. Page 2 walks one action through them.

RSBIS CORE

Helps with each step, today or as planned. Page 3 says which parts, and where each stands.

RSBIS, FURTHER

Carries it up to the institution. Page 4 runs the harder cases and shows how to check them.

How this example fits. Example 2 at full depth: the same $500 payment as the hand-made 4-page edition, now on sealed records; it adds a second approver for every payment, the invoice Supplier A signed for, and the harder cases, computed.

RSBIS is being built. Every capability in this example is designed and internally tested in its sealed records, and on the RSBIS build plan.

● **In RSBIS today** in code, with tests; not yet live **Designed and tested:** ◐ **coding scheduled** What is built covers one action: an AI model deployment.

An invented company, its supplier and its bank; every person, account, rule and amount is invented. The sealed records are signed with test keys that protect nothing (anyone can recreate the Ed25519 ones), so a signature there proves nothing about a real party.

MOMENT 4 · OCTOBER 21 · PAY ONCE

One payment, through the ten steps

The payment agent asks: pay $500 to Supplier A, account ending 4417, for invoice 104, once.

THE INVOICE
Invoice 104 for $500, matched to PO-77 and its goods receipt. Supplier A signed for it: paid once, to its account ending 4417.

THE RULES IN FORCE
US dollars, up to $1,000 per payment, after a three-way match, with two approvals; beside them, the bank's own terms.

	THE STEP	FOR THIS ACTION		WHO	ENTRIES
1	**Prepare the exact payment**	The agent asks; the bank holds $500 to Supplier A, account ending 4417, invoice 104, as pending: BANK#PENDING-0001.		ACTOR GUARD	8, 10, 11, 15
2	**Verify who is asking**	Check the payment agent's identity and its signed request.		GATE	8–10, 14
3	**Check permission**	The agent may ask to pay matched invoices up to $1,000; it moves no money on its own.		GATE	9, 14
4	**Apply the rules**	US dollars, up to $1,000 per payment, after a three-way match; Supplier A's signed amount and account; the bank's terms.		GATE	9, 14
5	**Check extra approval**	The finance owner and the payables controller each approved exactly this payment.		GATE	9, 12–14
6	**Decide and issue the pass**	Accept: a pass for the pending payment only. Refuse a payment asked for before the two approvals.	CAN REFUSE	GATE	9, 14
7	**Present the pass**	The agent sends the pass to the payment guard, which matches it to BANK#PENDING-0001.		ACTOR	16, 18, 21
8	**Verify the pass**	Signer, presenter, the pending payment, expiry and prior use. A redirect to account ending 9021 fails here.	CAN REFUSE	GUARD	17, 19, 22
9	**Release the payment once**	Release it once, so the same payment can't go twice; done only when the bank confirms.		GUARD	19, 20
10	**Record and reconcile every outcome**	Match the bank's confirmation to the approved payment; keep any mismatch open for the reviewer.		GATE GUARD	20

When something goes wrong

Refused
No approvals yet, a changed account or a second payment: stop and keep the reason. Refusing 9021 restricts the route.

Failed or interrupted
Check with the bank whether the payment was released before trying again.

Unknown
Record it as unknown. Ask the bank by the pending payment's own number before any retry.

Done outside the path
The payment route's log shows a payment with no pass: record an incident, with evidence, and close or name the route.

The account is fixed before anyone approves. The bank holds the exact payment, account ending 4417, as pending before the two approvals name it ◐; RSBIS's exact-action digest: any change makes a different action ●; so a vendor file changed afterward cannot redirect it: the guard refuses account ending 9021 and restricts the route ◐.

RSBIS is being built. Every capability in this example is designed and internally tested in its sealed records, and on the RSBIS build plan.

● **In RSBIS today** in code, with tests; not yet live **Designed and tested:** ◐ **coding scheduled** What is built covers one action: an AI model deployment.

In the sealed record: ENTRIES gives the entries of supplier-payment.specimen.yaml that show each step for this action; every one is signed with test keys, and its decisions are recomputed by the examples tool.

WHAT RSBIS ADDS

What RSBIS adds at each step

The method says what. RSBIS helps achieve it. And goes further.

	THE STEP	RSBIS CORE · HELPS ACHIEVE IT	RSBIS, FURTHER · PLUS
1	Prepare the exact payment	RSBIS's exact-action digest: any change makes a different action ●; applied to a payout ◐; the bank holds it as pending first ◐	The bank's own terms checked beside the company's rules ◐ MOMENT 4
2	Verify who is asking	Checks the requester against its deed ●; the agent's own signed request ◐	An outside sign-in bound to a deed, and unbound ● MOMENT 4
3	Check permission	Written delegation records, withdrawn at any time ●	Each payment within the agent's $1,000 limit ◐ MOMENT 2
4	Apply the rules	Rules that always finish ●; checked on each payment's values ◐	The invoice matched to its order and goods receipt first ◐ MOMENT 3
5	Check extra approval	Two approvers name this exact payment ◐	Distinct signers are counted: one signer identity never counts twice ● MOMENT 4
6	Decide and issue the pass	A one-use pass, signed with two algorithms ●; short-lived and tied to its target by default ◐	The deed's own policy on whether an AI may sign ● MOMENT 4
7	Present the pass	Works only for the agent it was issued to ◐	A failed check restricts the route for new passes ◐ MOMENT 5
8	Verify the pass	The guard checks the pass's signer, expiry and prior use ●; and the match to the pending payment ◐	It rechecks the account Supplier A signed for, which a stolen gate key cannot change ◐ MOMENT 4
9	Release the payment once	Released once; done only when the bank confirms ◐	A rule change stops the passes not yet used ◐ MOMENT 4
10	Record and reconcile every outcome	A signed, hash-linked record a reviewer rechecks offline ●; the bank's confirmation matched to the approved payment, any mismatch kept open ◐	The gate's decisions rechecked offline ●; with each payment's evidence in a compliance export ◐ MOMENT 6

WHERE EACH PART STANDS ON THE PLAN

RSBIS plan records, SDK main fd9d119a, read November 6, 2026; public view; checked by AI

CAP01	Authority and delegation · T3-01, T3-04, T3-07 …	In RSBIS today ● (in part) Coding scheduled ◐
CAP02	Inherited rules and other authorities · T5-55, T5-39, T5-40 …	In RSBIS today ● (in part) Coding scheduled ◐
CAP04	The target prepares the exact action, after checking the requester · T5-55, T2-11, T2-12	Coding scheduled ◐
CAP05	Exact action and agreement binding · T2-06, T1-03, T5-02 …	In RSBIS today ● (in part) Coding scheduled ◐
CAP06	The target's own checks if a gate key is stolen · T5-55, T2-13, T0-54 …	Coding scheduled ◐
CAP07	Each business action done once · T2-11, T2-14, T5-55	In RSBIS today ● (in part) Coding scheduled ◐
CAP10	Recovering an unknown outcome · T2-03, T5-03, T5-26 …	In RSBIS today ● (in part) Coding scheduled ◐
CAP11	Rule and role changes · T1-34, T1-31, T3-01 …	In RSBIS today ● (in part) Coding scheduled ◐
CAP12	Suspension with continuing refunds and remedies · T5-56, T5-62	Coding scheduled ◐
CAP14	Evidence others can check offline · T2-10, T1-35, T1-22 …	In RSBIS today ● (in part) Coding scheduled ◐
CAP16	Signature suites and key eras · T0-01, T0-06, T0-55 …	In RSBIS today ● (in part) Coding scheduled ◐

The tool refuses a page that marks a part in RSBIS today when the plan says otherwise. Everything built covers only the one action RSBIS runs today, an AI model deployment.

RSBIS is being built. Every capability in this example is designed and internally tested in its sealed records, and on the RSBIS build plan.

● **In RSBIS today** in code, with tests; not yet live **Designed and tested:** ◐ **coding scheduled**

HARDER CASES · CHECK IT YOURSELF

Every case, computed

Each case changes one thing; a case that also assumes a second stolen key says so. Its result is the decision the build computed from the record.

CASE	WHAT CHANGES	ENTRIES	COMPUTED RESULT
C29	Correct control: matched, approved twice, paid once	STORY 10–20	PERMITTED
C15	Payment asked for before the two approvals	STORY 8–9	REFUSED · QUORUM_NOT_MET
C07	Redirect to account ending 9021	STORY 15–17	REFUSED · ACTION_BINDING_MISMATCH
C31	A second payment from the same pass	STORY 21–22	REFUSED · ALREADY_COMPLETED
C11	A new payment on the restricted route	STORY 23–24	REFUSED · ENTITLEMENT_SUSPENDED
C16	One person approving twice	CASES 27–32	REFUSED · DISTINCT_PRINCIPAL_REQUIRED
C02	Pass presented by another agent	CASES 13–14	REFUSED · PRESENTER_MISMATCH
C08	Pass taken straight to the bank's wire desk	CASES 15–16	REFUSED · AUDIENCE_MISMATCH
C32	A second pass for the same payment	CASES 12–21	REFUSED · ALREADY_COMPLETED
C01	Stolen gate and agent keys	CASES 22–24	REFUSED · ACTOR_AUTHORITY_MISSING
C24	A negative amount	CASES 25–26	REFUSED · VALUE_INVALID
C05	The agent asks beyond its $1,000 limit	CASES 33–36	REFUSED · SCOPE_EXCEEDED
C12	Cover controller's role ended before the decision	CASES 37–43	REFUSED · SIGNER_ROLE_INACTIVE
C10	Rule change between pass and use	CASES 44–53	REFUSED · POLICY_STALE

NOT RUN HERE, AND WHAT A REAL RUN NEEDS

C33	**Restored consumed state.** Needs a crash-and-restore run of the bank connection, with an outside record of what it already released (T2-14).
C35	**A payment made outside the path.** Not covered: needs the payment route's own log, read independently, and a run that matches each payment to its pass (T5-60, planned).
CRASH01-08	**Crash boundaries.** Needs real process stops and restarts at each boundary (T2-14, T5-06).

Check it yourself

this build: PASS

836
checker checks passed
hashes, links, digests and every signature, against the trust inputs

23 of 23
decisions recomputed
from the frozen checks, matching the record and the expected result

17 of 17
tampered copies refused
each for its stated reason

```
# everything, the scenario built again included (cd tool; see README.txt). In the study pack: sh check_everything.sh
python3 build_example.py verify scenarios/supplier_payment_payout.scenario.yaml \
    --out .. --core <path>/RootZero__Deed_CORE.yaml
# one record, with the checker alone
python3 checker/verify_specimen.py ../records/supplier-payment.specimen.yaml \
    ../records/trust-inputs.json --expected ../records/expected-results.json
```

built **November 6, 2026, 7:47 pm PST (03:47 UTC November 7)** scenario sha-256 **3d67dc6dd0f2b097…** sealing tools from SDK main **70aa7786**
keys **TEST keys only** records **53f0285a9dce5c26**

Offline, with no network. The records (supplier-payment.specimen.yaml supplier-payment-cases.specimen.yaml) are signed with test keys that protect nothing, so verify builds each record again from its scenario and compares the two, and eight fixed requirements hold whatever the scenario says. To show these are the files that were published, compare them with a copy kept elsewhere. A record shows what was recorded, not what was left out.

HOW THE PIECES FIT

Governance, security and assurance, joined at every action

Most organizations run three disciplines with three teams, three tools and three records. **Actions Science joins them at the one place they all care about: the consequential action.** The same ten steps set the rules, enforce them and prove what happened.

> **In this example:** the finance owner's approval of invoice 104 sets the rule, the bank's guard pays $500 to Supplier A once, and the bank's confirmation proves it.

GOVERNANCE · GRC

Set the rules

governance, risk and compliance

Who may do this, within which limits?

Who may act, who must also approve and the rules each action must meet are written down first. The gate applies them, then approves or refuses.

Steps 2–6: verify who is asking, check permission, apply the rules, check extra approval, decide.

SECURITY

Enforce them

prevent and protect

Can it happen only as approved?

The guard checks the pass where the effect happens and admits one exact action, once. Other routes are closed, or watched.

Steps 7–9: present the pass, verify it at the target, do it once. **Test 4:** no way around.

ASSURANCE · IT AUDIT

Prove what happened

check and assure

Can someone else confirm it?

What was allowed, what was tried and what was done are kept apart and reconciled. Every mismatch stays open until its owner explains it.

Step 10: record and reconcile every outcome. **Test 5:** checkable without you.

The same ten steps, three disciplines · step 1 defines the exact action all three depend on

1	2	3	4	5	6	7	8	9	10
Prepare the exact action	Verify who is asking	Check permission	Apply the rules	Check extra approval	Decide and issue the pass	Present the pass	Verify the pass	Do it once	Record and reconcile
BASIS	GOVERNANCE					SECURITY			ASSURE

Three disciplines, working apart

- Policies say who may act; systems may not check them
- Controls block some actions; others go around them
- Audits reconstruct afterward, from logs that show what was allowed, not what was done

Three disciplines, joined at the action

- The rule written at design time is the rule enforced
- The guard admits only the exact approved action, once
- The record proves it, and anyone can recheck it

Start with the highest-risk actions, and the actions that can change them.

Each action brought under the ten steps gains all three: **the rules**, **the enforcement** and **the proof**. Coverage grows one action at a time, with existing tools and teams.

What this page does not claim. Actions Science covers the action layer. It works alongside identity, threat monitoring and vulnerability management; it does not replace them. Each team keeps its own decision rights.