

Scaling Pega Agentic Safely: Two Governance Capabilities Your Programme Needs

Bounded Error Verification · Cryptographic Authority Binding

InnoFissionAI · For clients using Pega Infinity '26 · v1 · September 2026

What this document covers

This document states what these two capabilities do and what they guarantee. It does not describe how they work. Technical specification is available to named individuals under NDA.

CONFIDENTIAL

PATENT APPLICATIONS PENDING

Two capabilities addressed

01

Bounded Error Verification

Proof that Pega agent decisions are correct, measured on live work.

02

Cryptographic Authority Binding

Proof that what executed matches what was approved in Pega.

Why You Need These Now

Your first Pega agentic pilot worked. The demo was impressive. Your Pega case lifecycle held perfectly. Then governance asked three questions:

1

"How do we know the Pega agent's decision is actually correct?"

Not from testing. From ongoing proof, measured on live Pega work.

2

"Can you prove what executed in our downstream systems matches what was approved in Pega?"

Pega records that we approved. It does not prove what executed matches what was approved.

3

"What evidence will we show auditors when a Pega decision is disputed?"

Can we reconstruct the proof two years later?

Why your Pega team cannot answer these

Your Pega team cannot answer these. Not because Pega is weak — Pega is excellent at orchestration and audit. These questions sit outside what any workflow platform can answer. They require proof that spans from Pega approval through your downstream systems and back.

The load-bearing stall

The stall is not technical. It is governance. And it is load-bearing: it is the difference between a Pega pilot that gets funded for scale and one that gets archived.

You need two capabilities that sit beside Pega, integrate with Pega, and answer the exact questions your risk committee will ask before approving scale.

Capability One: Bounded Error Verification

The problem it solves

Your Pega Predictable AI front-loads reasoning to design time. That is the right choice — it delivers consistency, auditability, and flat per-case pricing. By construction, it is strongest on known paths and weakest on cases that do not resemble the ones your Pega workflow was designed against.

Those novel cases fall out to your Pega approval queues today. They are expensive. They are the majority of the cost in exactly the high-variability Pega processes your CFO most wants to automate.

☐ **Headline proof:** "83% of your disputed invoices accepted autonomously at $\leq 0.8\%$ error rate. 17% routed to your dispute team." Coverage at the error rate you choose. Your governance team sets tolerance; we report the coverage it buys.

What this capability delivers

- A domain reasoner running entirely on your infrastructure, trained on your own Pega case history. No data egress, no metered token cost, no dependence on a frontier vendor's roadmap.
- Verified against your own Pega business rules, not against a hand-written policy or a second opinion from another model. Your existing Pega rules are the boundary. Your Pega team wrote the verifier.
- Routes to your Pega approval queue on uncertainty, not on the model's uncalibrated confidence. Clear decision: accept or escalate. No hedging.
- Full decision provenance — reasoning, sources, which Pega rules were evaluated, the exact model version. Retrievable from your Pega case history years later.
- Surfaces Pega rulebase drift. When the reasoner persistently disagrees with your Pega rules, that usually means the rules are stale — a billable finding your governance team should act on.

Integration with Pega

Called as a Tier 1 read-only component from a Pega case step, over Pega's native MCP exposure. The case stays entirely deterministic within Pega. Pega's flat per-case pricing is preserved because the reasoner's cost is fixed and internal.

How it is proved

One sentence, measured on your actual Pega case history:

"83% of your disputed invoices accepted autonomously at $\leq 0.8\%$ error rate. 17% routed to your dispute team."

Capability Two: Cryptographic Authority Binding

The problem it solves

Your Pega approval process is flawless. The audit trail shows who approved in Pega, when, with full context. All correct, all auditable. That is Pega excellence.

What Pega does not do — what no workflow platform does — is prove that the action which executed in your downstream systems is byte-identical to the action which was approved in Pega. Pega owns the approval. Your bank owns the execution. No one owns the span. And until a dispute lands or a regulator inquires, that gap is invisible.

- ❏ **Headline proof:** "100% of your Pega terminal actions signed. Zero instances of payload drift between Pega approval and downstream execution." Binary proof. Either the binding held or it did not.

What this capability delivers

- A signed record for every terminal Pega action, binding the approver's verified identity, the exact Pega payload that was approved, the constraint set in force at that instant, and the executed result. If anything changes between Pega approval and downstream execution, the action does not fire.
- Verification by any auditor, independently, without trusting Pega and without access to Pega. Verification that depends on the vendor is not verification.
- Independent monitoring from outside Pega, using Pega's supported real-time extract feed. Cannot be suppressed if Pega itself is compromised. Highest-value alert: a Pega action executed with no corresponding Pega approval record.
- Engagement evidence, not timestamps. Did the approver actually examine the Pega payload, or rubber-stamp the queue? Engagement telemetry — time spent, sections reviewed — is part of the signed record. This converts an EU AI Act Article 14 claim from a statement into proof.
- One record, multiple uses. The same signed evidence satisfies EU AI Act logging requirements, DORA incident timelines, fair-lending adverse-action traceability, and dispute resolution.

Integration with Pega

Sits immediately before your Pega connectors to systems-of-record. The last deterministic gate between Pega's case and any downstream mutation. Uses Pega's supported data extract path; no platform changes required.

How it is proved

"100% of your Pega terminal actions signed. Zero instances of payload drift between Pega approval and downstream execution."

How They Work Together with Your Pega Programme

The two questions answered

The **Reasoner** answers: "Is this decision sound enough?"

The **Notary** answers: "Did what execute match what was approved?"

Together they answer all eight questions your risk committee will ask before approving scale.

Eight risk committee questions — answered

The question	How these solve it
How do we know the Pega agent's decision is correct?	Reasoner: bounded error rate, measured on live Pega work
What happens if the agent is uncertain?	Reasoner: calibrated routing to your Pega queue, with reason stated
Who authorised this Pega action?	Notary: verified identity cryptographically bound to the Pega payload
How do we know the approver actually looked?	Notary: engagement evidence in the signed record
Could the agent have acted outside Pega permissions?	Deterministic Pega gate before every effectful action
Would we know if it had?	Notary: independent monitoring, cannot be suppressed
Can we reconstruct this in a Pega audit, years later?	Both: decision provenance in Pega history + verifiable authority record
What if Pega itself is compromised?	Notary: verification is independent, does not depend on Pega

- ❑ A programme that answers all eight gets governance approval. One that cannot, does not — regardless of how good your Pega implementation was.

Worked Example: A Pega Disputed Invoice

An invoice arrives in your Pega accounts-payable case. It does not match the PO. Supporting docs are scanned PDFs. Contract terms sit in a separate policy system. The amount is material.

Today, with Pega alone

- Specialist stages partial payment in Pega. Manager approves in four seconds from mobile. Pega records: timestamp, approver, approval. Case closes.
- Twelve months later, supplier disputes it. Your CFO's team spends a week reconstructing: What was the context? What was the reasoning? Was it within authority? Did someone actually consider it?
- Pega's audit trail records the decision, not the deliberation.

With both capabilities, alongside Pega

- The Reasoner assembles context from all three systems, applies your Pega business rules, reaches a conclusion. Where it agrees and confidence is calibrated, it stages the payment in Pega with reasoning and sources attached — visible to specialist and manager.
- Where the contract language is unusual and the reasoner is unsure, it routes the case back to the specialist with analysis half-done. Manager sees what changed, what it was based on, where reasoning was weakest.

The approval in Pega is signed by the Notary, binding approver identity to that exact payload. When your Pega connector fires to the bank, the Notary gate confirms the binding still holds. If anything shifted — amount, account, vendor ID — the payment does not go out.

- Twelve months later, supplier disputes it. Your CFO produces a single verifiable record: what was proposed, on what evidence, who authorised it in Pega, under which rule version, with cryptographic proof nothing changed between Pega approval and bank execution. Dispute resolved.

The commercial difference is not the twenty minutes saved. It is that the process becomes eligible for Pega autonomy at all — and that evidence exists without anyone having planned for this specific dispute.

First Ninety Days: Proof Before Scale

Do not commit to scaling your Pega agentic programme until governance approves it.

Phase	Duration	Your Pega involvement	Outcome
Prove	Weeks 1–4	Notary runs against one Pega approval path	An independent auditor verifies a Pega decision. Go/no-go on evidence.
Bound	Weeks 5–8	Pega process suitability assessed	Coverage-at-error target agreed with your business.
Demonstrate	Weeks 9–12	Reasoner shadows live Pega traffic	Coverage and error measured on real Pega case mix. Nothing acted upon.



Prove

Weeks 1–4, Notary verification



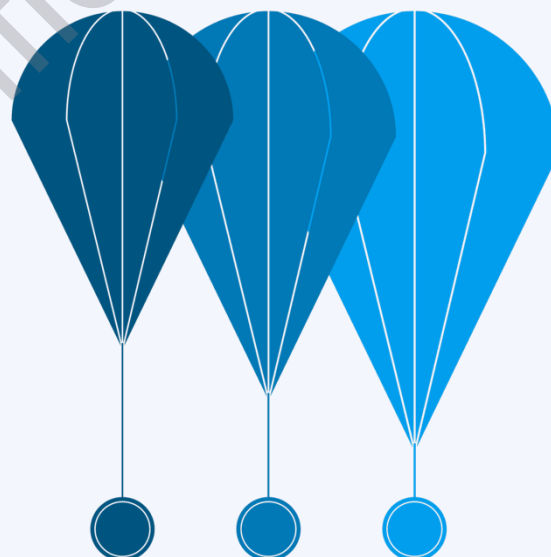
Bound

Weeks 5–8, coverage target agreed



Demonstrate

Weeks 9–12, Reasoner shadows live traffic



Nothing goes to production Pega and no autonomy is granted in this window. Output is an evidence pack your risk committee uses to decide: scale or stop.

Why This Matters for Your Business

CFO perspective

Failed agentic pilots on Pega cost credibility and capital. This framework kills them in week four, not week sixteen. £50k to learn something is not deliverable costs far less than £500k to discover it in production.

CIO perspective

This operates on your Pega infrastructure, trains on your Pega case data, validates against your Pega rules. Not a black-box vendor system. Your data stays inside your estate.

Chief Risk Officer perspective

Governance approval is no longer "do you believe the vendor?" It is "do you trust this evidence?" Evidence is auditor-verifiable and regulator-defensible.

Chief Information Security Officer perspective

The Notary catches tampering between Pega approval and downstream execution — whether from a compromised Pega instance, a hijacked connector, or drift in a backend system.

CXO perspective

These capabilities let you scale your Pega agentic programmes with governance confidence. You do not have to choose between speed and safety.

Regulatory Coverage: One Record, Multiple Uses

The Notary's signed record is not a bespoke artefact for a single regulation. The same evidence satisfies multiple concurrent obligations across your regulated estate.

EU AI Act

Article 14 human oversight: engagement telemetry converts a claim into proof. Logging requirements satisfied by the signed record.

DORA

Incident timelines reconstructed from the verifiable authority record. No dependency on Pega availability during incident response.

Fair Lending

Adverse-action traceability: the exact payload, rule version, and approver identity are cryptographically bound and independently verifiable.

Dispute Resolution

A single verifiable record replaces a week of CFO team reconstruction. Proof that nothing changed between Pega approval and downstream execution.

What independent verification means

Verification by any auditor, independently, without trusting Pega and without access to Pega. Verification that depends on the vendor is not verification. The Notary's design principle: the signed record must be checkable by a party who has never seen your Pega instance and never will.

What engagement evidence means

Did the approver actually examine the Pega payload, or rubber-stamp the queue? Engagement telemetry — time spent, sections reviewed — is part of the signed record. This converts an EU AI Act Article 14 claim from a statement into proof. It is the difference between asserting human oversight and demonstrating it.

Capability Comparison at a Glance

Both capabilities are designed to sit beside Pega, not replace it. Each addresses a distinct governance gap that Pega — by design — does not cover.

Dimension	Bounded Error Verification (Reasoner)	Cryptographic Authority Binding (Notary)
Core question answered	Is this decision sound enough?	Did what execute match what was approved?
Proof type	Statistical — bounded error rate on live Pega case mix	Binary — cryptographic, independently verifiable
Infrastructure	Runs on your estate; no data egress	Sits before Pega connectors; no platform changes
Pega integration point	Tier 1 read-only call from Pega case step via MCP	Pega supported data extract path; real-time feed
Governance output	Coverage-at-error report; rulebase drift alerts	Signed record per terminal action; engagement telemetry
Regulatory use	Decision provenance; audit reconstruction	EU AI Act, DORA, fair lending, dispute resolution
Failure mode caught	Novel cases outside Pega workflow design envelope	Payload drift; compromised connector; rubber-stamp approval
Headline metric	83% autonomous at $\leq 0.8\%$ error; 17% routed	100% signed; zero payload drift instances

83%

Autonomous resolution

Disputed invoices accepted at $\leq 0.8\%$ error rate

8

Questions answered

All risk committee questions resolved by the two capabilities

100%

Actions signed

Every terminal Pega action cryptographically bound

90

Days to evidence

Proof pack for your risk committee, before any production commitment

The Next Steps

One regulated Pega process. One Pega approval gate. Ninety minutes.

Pick a Pega process where agentic roadmap has stalled in governance. We stand up the Notary against one Pega approval path and show your risk committee auditing a Pega decision independently — no slideware, a working verification.

If governance asks "will you answer these eight questions?" — we answer them in a fortnight. If they ask "can you prove it?" — we prove it with evidence, not assertions.

If it does not survive their questions, you learn that in week two rather than in month five of a full Pega programme.

What you get from the first session

01

Working verification

Your risk committee audits a real Pega decision independently — not a slide, a live demonstration.

02

Eight questions answered

In a fortnight, with evidence. Not assertions.

03

Go/no-go in week two

Not month five of a full programme.

Confidential. Patent applications pending. Technical specification available under NDA.

InnoFissionAI · Enterprise Automation & AI · Applied AI Engineering ·
info@innofissionai.com

PEGA INFINITY '26

V1 · SEPTEMBER 2026

CONFIDENTIAL