Ond&Co

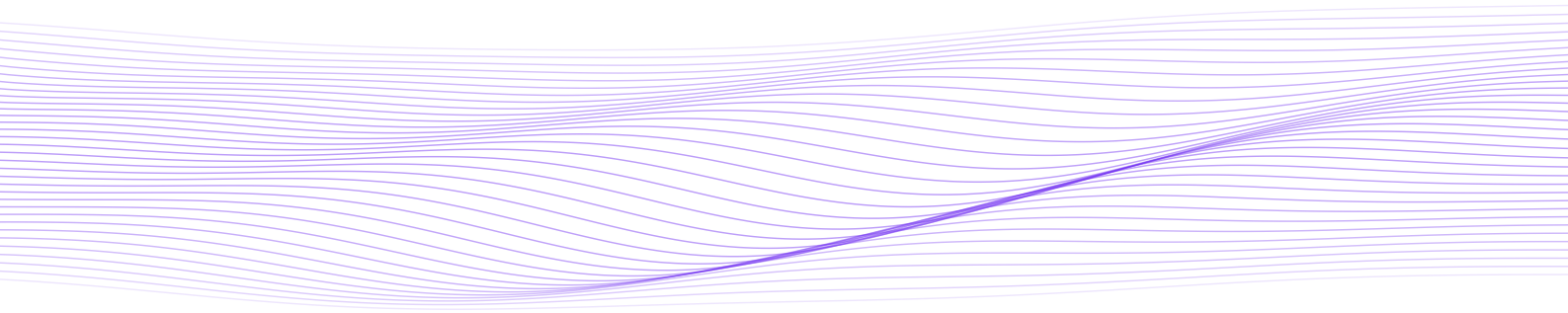
OSuite.ai

INCIDENT-INSPIRED TECHNICAL CASE

Service access is not payment authority.

Inspired by public disclosures about the Binance incident, we tested whether a compromised service credential could become unauthorized payment authority. The controlled experiments exposed gaps in our earlier gateway-based composition and informed a tighter resource-side execution boundary.

Matoi now connects human review, current authority, exact transfer details, an atomic ledger commit and a signed receipt in one controlled-ledger workflow.



■ SDK 2.12.0 published.

□ Profile disabled pending deployment qualification.

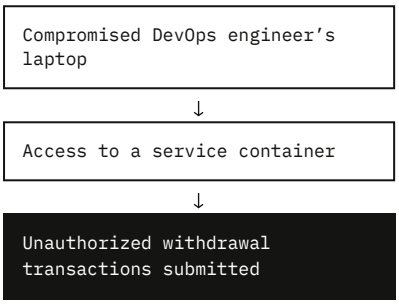
Controlled-ledger scope. Not bank or blockchain settlement. Not a reconstruction of Binance's systems, and not evidence that OSuite would have prevented the incident. Sources, test scope and versions: Appendix A3.

What was disclosed, and what we chose to test.

Identity, approval and a gateway can all be present while a compromised business service still reaches the system that moves money. We studied one segment of that risk. We keep three kinds of content separate.

A · PUBLIC DISCLOSURE

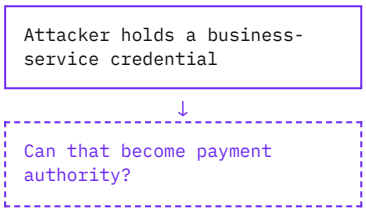
Initial Binance disclosure, Oct 29, 2026 [1][2]



Binance said its initial investigation suggested this path. It found no evidence of a wallet private key leak. The cited initial disclosures did not establish the complete attack chain.

B · OUR EXPERIMENT HYPOTHESIS

Controlled ledger, synthetic effects



We asked whether holding a service credential is enough to move funds beyond what a person approved and what is currently authorized. This is our test setup. It does not claim Binance's internal systems work this way.

C · STRESS TESTS WE ADDED

Not confirmed incident causes

- Concurrent duplicate requests
- Revocation racing execution
- Write paths that skip the gateway
- Lost responses and crashes

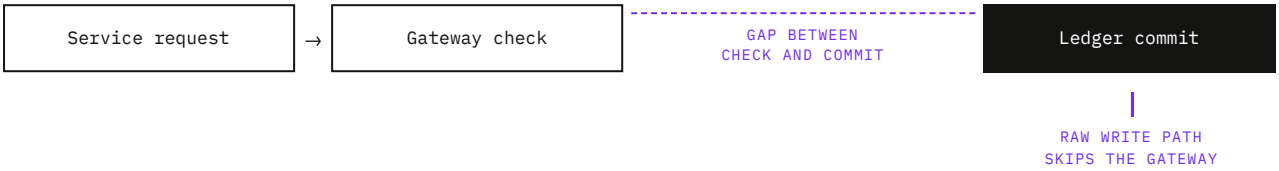
We added these conditions because they test the execution boundary. None of them is presented as what happened at Binance.

> The question we studied: once an attacker holds a service credential, what still stands between that credential and a committed payment?

Column A cites only Binance's initial disclosure of October 29, 2026. Amounts, later updates and third-party estimates of mechanism and attribution are not used in this document.

A gateway check was not enough.

Our earlier composition checked requests at an upstream gateway, and the ledger committed later. The first controlled tests found gaps in that arrangement. They were gaps in our own design.



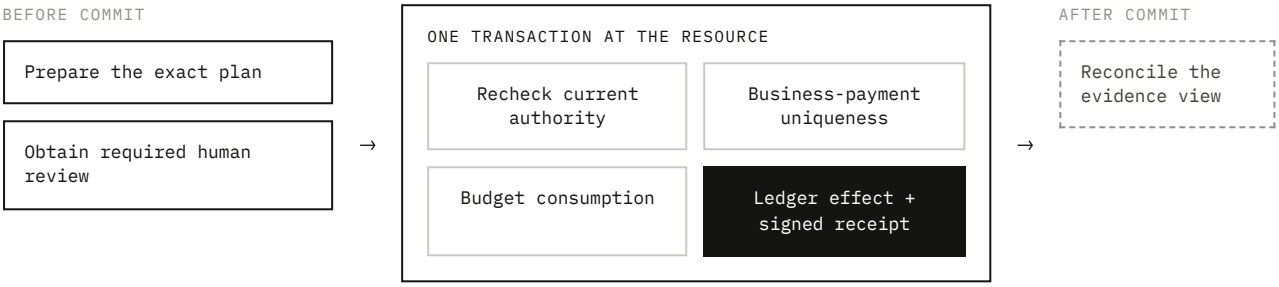
01	Concurrent duplicate payment	With checks only at the gateway, concurrent requests for the same business payment could each be admitted before the ledger recorded the first.
02	Revocation racing execution	Authority revoked after the gateway check but before the commit did not reliably stop the commit. Earlier approval was standing in for current authority.
03	Bypassing the gateway	A write path that does not pass through the gateway is not governed by it. A service credential with direct access to the resource can skip the check entirely.

> **The lesson is about where the check sits, not who supplies it.**
>
> In the same comparison, a conventional, independent resource-side authorization check was also effective. A gateway alone, or an earlier approval alone, was not enough.

Check and commit belong together.

At the controlled resource boundary, current authority, business-payment uniqueness, budget consumption and the ledger effect participate in one transaction.

The controlled-ledger profile binds the payment reference, source account, destination, asset, network and exact amount. The SDK and MCP paths connect that request to OSuite Studio review and the resource owner.



GAP FOUND	CONTROL IN THE IMPLEMENTED PROFILE
A different request repeats the same business payment	Payment identity is checked at the ledger.
Authority changes before the commit is authorized	Current authority and approval are rechecked inside the transaction.
A request exceeds the permitted debit	Grant and account limits are enforced with the ledger update.
A response disappears after a possible commit	Recovery reads the original operation instead of blindly sending again.

Scope: The guarantee depends on the protected resource boundary. It does not cover an unrelated raw write path or make a remote bank part of this transaction. Earlier approval does not bypass current checks. A receipt is evidence of a commit, not a reusable permission.

Specific behavior, not an abstract score.

We then tested concurrency, lost responses and crash recovery in the resource-side controlled-ledger profile. Each result below is an observed behavior of the implemented profile.

CONCURRENCY

100 → 1

100 concurrent requests for the same business payment produced exactly one commit.

LOST RESPONSE

No resend

When a response was lost after a possible commit, the client read the original operation instead of sending the payment again.

CRASH

Same op

Recovery queried the original operation. An uncertain payment was not automatically resent.

LEGITIMATE WORKLOAD · FINAL RUN

1,800 scheduled actions. 1,800 known commits.



1,711 known committed within 2 seconds

89 took longer than 2 seconds

Every scheduled action reached a known committed outcome. This measures completion of legitimate requests under load. It is not an attack-detection rate or a security-accuracy figure. Both runs, the measurement method and the acceptance record are in Appendix A2.

Synthetic ledger effects, not bank settlement. No exchange performance comparison.

Released as code. Deliberate in rollout.

The controlled-ledger implementation is complete. The profile remains disabled pending deployment qualification; existing customers are not automatically opted in.

WHERE IT FITS
You operate the ledger or payment service that records the effect.
Every write to it can be routed through the protected resource boundary.
Agents or services request payments that people must approve.
An operator can reconcile unresolved outcomes.

WHAT IT DOES NOT YET PROMISE
Protection for raw write paths that bypass the boundary.
A remote bank or blockchain inside the same transaction.
Coverage of every tool and payment provider.
A real-world incident-reduction percentage, production qualification or an SLA.

When correctly enabled at an enforced resource boundary, these controls can help reduce duplicate-payment, stale-authority and blind-retry failures. Their value is a more explicit boundary between what an agent requests, what people approve, what remains authorized and what the resource actually commits.

Exact effects. Current authority. Recoverable evidence.

Discuss your execution boundary or a pilot with OSuite.

support@osuite.ai

Show what is known. Keep uncertainty visible.

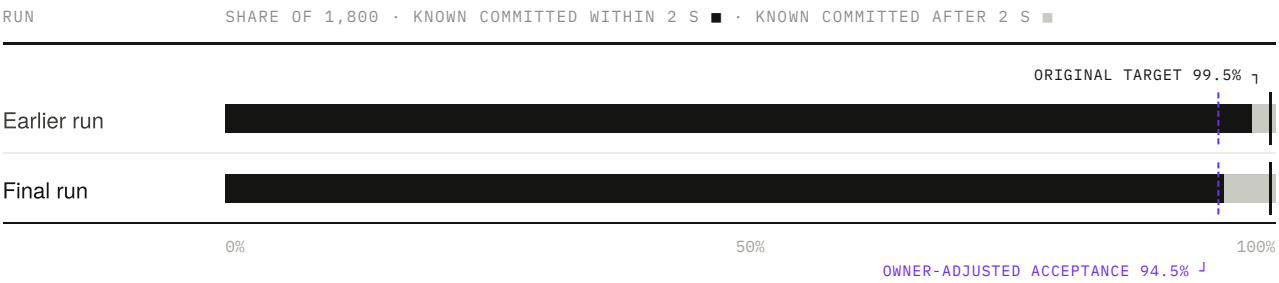
STATE	EXACT EXPLANATION	NEXT ACTION
`Awaiting approval`	Required review is not complete. No execution permission should be inferred.	Wait for the required reviewer.
`Checking execution conditions`	Review has been recorded. Current execution conditions are still being checked.	Wait for this operation's result. Do not submit again.
`Commit verified; view updating`	A verified receipt confirms the ledger commit. The evidence view is still updating.	Keep the receipt and refresh the read-only view.
`Last recorded state; view stale`	This is a prior observation, not a fresh execution result.	Read the original operation's status.
`Outcome unknown`	The response does not establish whether execution completed. Silence is not proof that nothing happened.	Read status for the same operation. If unresolved, request operator reconciliation.

You can continue unrelated work while waiting. Do not resend an unknown operation, create a replacement request or manually mark it complete. Refreshing evidence must not trigger another transfer.

Two finite lab runs, both reported.

Two finite controlled-ledger lab runs. The final result is shown alongside the earlier, better result.

MEASURE	EARLIER RUN	FINAL RUN
Eventually known committed	1,800 / 1,800	1,800 / 1,800
Known committed within 2 seconds	1,760 / 1,800	1,711 / 1,800
Within-2-second completion rate	97.7778%	95.0556%
Original 99.5% target	Not met	Not met



Method: 10 requests per second over a 180-second schedule, with 10 clients and batches. Time runs from scheduled arrival to known commit, including preparation, queueing and two automated reviewer requests per batch. Human think time is not measured. Every planned request remains in the denominator.

ACCEPTANCE RECORD

After reviewing the results, the owner accepted a 5-percentage-point tolerance, lowering finite performance acceptance from 99.5% to 94.5%. The final 95.0556% meets that adjusted criterion. The original target remains unmet. This is an owner-adjusted acceptance decision, not a statistical confidence interval, production qualification or an SLA. Any statement that the profile passed performance acceptance must disclose that the threshold was adjusted after the results were reviewed.

Latency measurement, not security accuracy. Synthetic ledger effects, not bank settlement. No exchange performance comparison.

Where to check the evidence.

PUBLIC DISCLOSURE

[1] Binance, statement on initial investigation of the security incident
X / @binance · October 29, 2026

[2] Binance Suffers Security Breach Affecting $6.3 Billion in Assets
The Hacker News · October 2026 · secondary report of Binance's statements

RESEARCH

[3] Matoi: Selective Revalidation of Consequence Evidence for Agent Authorization
arXiv 2610.21394v1 · Z. Wang · October 15, 2026 · research preprint, not peer-reviewed

Results in this brief come from internal engineering validation conducted October 31–November 5, 2026. They are separate from the cited preprint. Raw experiment archives are not publicly released.

RELEASE

[4] OSuite SDK 2.12.0 release notes
github.com/OndCo/osuite-sdk · deployment scope and known limitations

TEST SCOPE

- Controlled ledger with synthetic effects; no bank or blockchain settlement.
- Finite lab runs; reviewers were automated and human think time was not measured.
- Failure conditions were added by OSuite and are not confirmed incident causes.
- No customer-production telemetry or independent certification.

VERSIONS

SDK	2.12.0 · published [4]
Controlled-ledger profile	Disabled · pending qualification
Customer opt-in	Not automatic
Document date	November 5, 2026