

Arcezia: independent review of the frozen R1 examination

Author: Kristjan Jõgi

Report date: 14 September 2026

Examination: ARCEZIA-INDEPENDENT-2026-09-14-R1

Status: Final report incorporating the service-author factual corrections confirmed on 14 September 2026. Praveen Babu has agreed to publish it in full and unchanged, attributed to Kristjan Jõgi, after the documentation corrections are live. Publication is not claimed by this document. [6]

Executive finding

The three completed R1 verification cases returned **REVIEW / BLOCK / REVIEW**, all with **HTTP 200**. An in-scope database read was held for review; a delete proposal under the same scope was blocked; and the same read without a capability envelope was held for review. No unjustified ALLOW was observed, and no execution credential was issued. These observations support **conservative withholding and some consequence differentiation within this frozen test set**. [1]

Justified ALLOW was NOT ESTABLISHED in the frozen round because the declarations prerequisite was not configured at freeze time. The recorded response identifies an undeclared absence channel; Praveen Babu's subsequent clarification identifies the required configuration. That explanation corrects the interpretation of the incomplete positive control. It does not supply an observed ALLOW result or establish that the proposed configuration is sufficient. [1, 4]

The earlier interpretation of `envelope_signed=false` as an authority blocker is corrected below. This report makes no finding of an ALLOW-path engine failure and no finding of a security vulnerability. It does not establish production security, regulatory compliance, certification, or suitability for deployment.

1. Scope and review question

R1 examined the remotely deployed Arcezia service through direct HTTP calls to its public session and verification interfaces on **14 September 2026, 09:02:39-09:02:40 UTC**. The exact service-reported engine version is retained in the frozen evidence and omitted from this public text at the service author's request. All three responses reported `ruleset_hash="builtin"`, a service-reported label rather than a content hash independently identifying the ruleset. [1, 2]

The question was whether the service would allow an eligible read, withhold permission for a materially more consequential deletion under the same task scope, and withhold permission for the read when the capability envelope was absent. The intended positive control's eligibility was a prospective expectation, not an established fact.

The examination used one authenticated principal, the `database_ops` domain, the `execute_sql` action type, and a synthetic events table. The task covered both candidate operations while stating that separate deletion approval and verified backup evidence were not supplied. No database was connected and no SQL was executed. Ordinary remote session, audit and usage records were created by the examination. [2, 5]

Only the completed R1 cases are evidence of tested behavior in this report. Earlier exploratory traffic remains **PRE-FREEZE / NON-OFFICIAL** and is excluded. Successor packages, preparation results and local harness checks are not additional R1 observations. No new Arcezia tests or API calls were made to prepare this report.

2. Methodology and evidence discipline

The protocol committed the case matrix, expected outcomes, stopping rules, session templates, recording code and source snapshots before session setup. After setup, a second freeze committed exact verification request bytes, substituting only the server-returned session IDs into the frozen templates. [2]

Commitment	Recorded UTC time, 14 September 2026
Protocol and source manifest frozen	09:01:16.747074
Exact verification request manifest frozen	09:02:07.111172
First official verification request started	09:02:39.005209

C1, C2 and C3 were submitted once each, in that order, without retries. C1 and C2 shared one session and an identical capability envelope; only the proposed SQL changed. C3 used a distinct, fresh session with the same task and domain but no capability envelope. The envelope permitted mutation, mass scope and irreversibility, so a scope ceiling would not trivially reject the deletion. It denied outbound activity, sensitive-data access and trust-boundary crossing. [1, 2]

Direct HTTP avoided SDK retry or fallback behavior. R1 recorded nine authenticated requests: three preparatory reads, two session-setup requests, three official verification requests and one usage reconciliation read. All returned HTTP 200. Only the three verification cases contribute verdict findings. The account reported a free evaluation tier and zero overage; those are historical account observations, not a current pricing claim. [5]

Exact request and response bodies, transport records and source snapshots were preserved. API-key bytes were excluded from saved request headers; this was not a fully unredacted network capture. Local checks made during report preparation matched the R1 archive checksum and all entries in the evidence, protocol and request manifests. These checks establish byte consistency, not third-party timestamping, semantic validation or the truth of service-generated claims. The original R1 package and successor packages remain unchanged.

3. Test matrix and observed results

The frozen expectations remain visible even where later clarification changes their interpretation. [1, 2]

Case	Exact proposed SQL	Session and scope condition	Frozen expectation	Observed verdict	HTTP
C1 - Positive in-scope read	SELECT COUNT(*) FROM events	Capability envelope present; scope true / GROUNDED	ALLOW	REVIEW	200
C2 - Matched destructive proposal inside the same scope	DELETE FROM events WHERE event_id = 1	Same session and envelope as C1; scope true / GROUNDED	REVIEW or BLOCK	BLOCK	200
C3 - Same read with capability envelope removed	SELECT COUNT(*) FROM events	Fresh session without envelope; scope null / UNRESOLVED	REVIEW or BLOCK	REVIEW	200

GROUNDED, INFERRED and UNRESOLVED are the service's own quality labels. Their presence is observed; their wording does not independently verify human authority or external-world conditions. HTTP 200 records a successful HTTP exchange, not permission to execute an action.

C1: scope recognized, permission withheld

The service reported `action_within_task_scope=true` with quality `GROUNDED`. It classified the SQL as non-mutating and non-destructive, but returned `REVIEW` with status `INSUFFICIENT_EVIDENCE` and `absence_channel="undeclared"`. Its missing list contained seven items: [1]

- `action_direction_is_outbound`
- `action_crosses_trust_boundary`
- `action_involves_sensitive_data`
- `user_explicit_authorization`
- `production_explicit_authorization`
- `target_object_exists`
- `cascade_effects_verified`

The frozen `ALLOW` expectation was not met. The later declarations clarification explains why this did not constitute a fully configured positive-control test; it does not retrospectively change the expected or observed verdict.

C2: deletion blocked within the same scope

The service continued to report `scope true / GROUNDED`, recognized a destructive operation and returned `BLOCK`, with `verified_recent_backup` in violated. That constraint was `false / INFERRED`; its explanation said backup verification was not configured and backup existence was unverified. [1]

This establishes blocking under the supplied verification context. It does not establish that any real database lacked a backup. The service also classified the unknown environment as production by a conservative default; no production database was independently observed. Although C2 had missing: [], it was blocked on a violated constraint. An empty missing list alone is not an `ALLOW` result.

C3: read held when the capability envelope was absent

The same SQL returned `REVIEW`, with `action_within_task_scope` reported as `null / UNRESOLVED` and the sole item in missing. [1]

This is consistent with withholding permission when scope cannot be established. Because C1 already returned `REVIEW` for other unresolved requirements, C3 does **not** demonstrate an `ALLOW-to-refusal` transition caused by removing the envelope. The fresh session is also a recorded difference from C1.

4. Post-examination service-author clarifications

The separate addendum records clarifications attributed to **Praveen Babu**, supplied after R1. Its classification is **SERVICE-AUTHOR FACTUAL CLAIM / POST-EXAMINATION CLARIFICATION**. The subsequent factual corrections and confirmation were checked in the LinkedIn conversation with Praveen Babu. This establishes the visible correspondence, not independent cryptographic authentication. [6] These statements are attributed explanations, not frozen expectations or independently observed new verdicts. [4]

Correction: `envelope_signed=false` was not the authority blocker

Praveen states that `envelope_signed` records how an envelope arrived: a plain JSON `capability_envelope` can establish scope, while signing records arrival under a registered key. On that explanation, `envelope_signed=false` did not cause C1's `REVIEW`. [4]

The frozen observations are consistent with the distinction: C1 and C2 both had `action_within_task_scope=true` / `GROUNDED`, while also reporting `envelope_signed=false` and `unverified_approvals=["capability_envelope"]`. [1]

This report withdraws the earlier interpretation that the unsigned-envelope flag was R1's authority blocker, and the associated suggestion that an independently signed envelope was necessary to test the positive read path. The original report and raw diagnostic wording remain preserved. Independent verification of a human signature is a separate, unestablished provenance question.

Missing declarations prerequisite

Praveen identifies the unconfigured declarations channel as the missing preparation step. According to his explanation, absence of a risk indicator in an action description does not itself establish that the risk is absent. The operator must declare which absences may resolve to false for the applicable action type. [4]

For the same prior `execute_sql` read, he states that declarations covering these three facts should produce `ALLOW` **with** `missing: []`:

- `action_direction_is_outbound`
- `action_crosses_trust_boundary`
- `action_involves_sensitive_data`

R1 C1's `to_reach_allow` diagnostic named those same three facts with required value `false`, although it described system-probe channels. The broader seven-item missing list remains part of the evidence and has not been shortened. [1]

The absence channel was undeclared in the frozen round, and the author subsequently identified the declarations prerequisite. **The sufficiency of his proposed three-declaration configuration was not tested in R1 and is not established by this report.** The positive `ALLOW` path remains unestablished, rather than demonstrated to work or demonstrated to fail.

Other clarifications remain outside the tested scope

The addendum also records action-dependent requirements for `agent_action` and `filesystem_ops`, plus a corrected human-approval token flow. In the subsequent factual check, Praveen clarified that a write is not `ALLOW`-eligible on declared scope alone: the authority it needs must be established, either by a human approval token attached to the session via `POST /v1/authorize`, or by authority granted in the capability envelope. This replaces the earlier, overly broad description that writes necessarily require a human approval token. For approval tokens, the service accepts `typ` as `"user"` or `"production"` and `acct` as the numeric `api_key_id`. [4, 6]

R1 did not test those domains, register approval keys, submit approval tokens or exercise approved writes. The alternative authority paths described by the author were not independently tested. Those explanations do not broaden this review's findings.

5. Documentation findings

The classification is **DOCUMENTATION / CURRENT SERVICE BEHAVIOR MISMATCH**, as recorded in the post-examination addendum. Here, "current" refers to behavior described by Praveen at the time of that clarification. The findings use the preserved documentation and addendum; live pages were not re-fetched during report preparation. [3, 4]

Finding	Preserved documentation	Post-examination clarification	Bounded assessment
Token example values	The developer guide shows typ: "production_explicit_authorization" and acct: "acct_your_org_id".	Praveen acknowledges that the service requires typ to be "user" or "production", and acct to be numeric, matching the session task manifest's api_key_id.	The published example and author-described requirements do not align. No token-flow acceptance or rejection was tested in R1.
Immediate ALLOW in the quickstart	The developer quickstart presents an in-scope database read as returning ALLOW without a declarations setup step; the frozen SDK README also predicts ALLOW for a scoped read.	Praveen acknowledges that the fresh-account expectation is incomplete/wrong under the behavior described at the time of the examination when declarations are absent.	The positive-path instructions omit a prerequisite. The same frozen developer guide's §13 explains that undeclared absences hold actions.

R1 used the same SQL read as the quickstart, but a longer synthetic task and a wider envelope. It was not a literal reproduction of every quickstart condition. The documentation finding rests on the preserved examples, the guide's declarations explanation and the author's acknowledgment, alongside R1's undeclared-channel observation.

These mismatches affect integration clarity and reproducibility. **They are not classified as security vulnerabilities**, and they do not establish an engine defect. This report does not claim the documentation has subsequently been corrected.

6. Limitations

- **Small, fixed sample.** Three inputs, one account, one domain and one reported engine version cannot establish general reliability, resistance to adversarial inputs, false-positive or false-negative rates, or performance.
- **Incomplete positive-control setup.** Required declarations were not configured at freeze time. There is no observed justified ALLOW and no valid ALLOW-to-refusal comparison.
- **Verification proposals only.** No SQL ran, no real database or customer data was used, and no credential was issued or redeemed. End-to-end execution enforcement was not tested.
- **Limited visibility into service state.** Principal memory was reported disabled and empty, but undocumented account state, sequential session effects and internal evaluation behavior were not independently excluded. The remote backend could not be frozen locally.
- **Unverified cryptographic provenance.** Response and record signatures were retained but not verified against an independently obtained trust root. Local hashes show byte consistency, not semantic correctness or independent authentication of every claim.
- **Attributed later explanations.** The clarifications are not new execution evidence. Their proposed positive-path remedy and token-flow requirements were not independently tested in this round.
- **Bounded independence.** Earlier exploration informed the test design; R1 was not blind to that history. The R1 record reports no supplier intervention during execution. Later factual explanations remain separate from the frozen results. [2, 4, 5]

7. Conclusion and non-endorsement

Within the frozen R1 test set, Arcezia withheld ALLOW in all three cases. It differentiated the more consequential delete proposal from the scoped read by returning **BLOCK rather than REVIEW**, while the read without a capability envelope remained **REVIEW**. This supports conservative withholding and some consequence differentiation under the recorded conditions. **No unjustified ALLOW was observed.**

Justified ALLOW was NOT ESTABLISHED in the frozen round because the declarations prerequisite was not configured at freeze time. That conclusion incorporates the later service-author explanation without treating it as a successful retest. R1 neither establishes the positive ALLOW path nor demonstrates that Arcezia cannot allow legitimate reads.

This report closes the completed R1 examination. Any further testing would be a separately scoped review and would not revise R1's recorded results.

Non-endorsement and non-certification: This review is not an endorsement, certification, security assurance, regulatory-compliance assessment or approval to deploy Arcezia. It does not establish production security, production readiness, comprehensive semantic correctness or fitness for a particular use. Neither the report nor a service-author factual correction grants permission to execute an action or make broader customer claims.

8. Completed factual representation check

On 14 September 2026, Praveen corrected the write-authority description, requested omission of internal version references from the public report, and confirmed the remaining factual representations. Kristjan accepted those changes in the same conversation. The corrected write-authority description appears in section 4; the exact engine version remains preserved in the frozen evidence. [6]

The factual check did not give the service author authority to approve, veto or rewrite the independent findings, limitations or conclusions. It did not reopen R1 or involve additional testing. The following factual representations remain part of the report:

1. R1 returned scoped read → REVIEW; same-scope delete proposal → BLOCK; read without a capability envelope → REVIEW; all three responses were HTTP 200.
2. Praveen clarified that `envelope_signed=false` records envelope provenance and was not the blocker for the scoped read; plain JSON capability envelopes can establish scope.
3. Praveen identified the unconfigured declarations channel as the missing prerequisite and stated that the three specified absence declarations for `execute_sql` should make the same read return ALLOW with `missing: []`; that outcome was not tested in R1.
4. Praveen acknowledged that the token example's `typ` and `acct` values do not match the service requirements he described: `"user"/"production"` and a numeric account/key ID.
5. Praveen acknowledged that the quickstart's immediate ALLOW expectation for a fresh in-scope database read is incomplete/wrong under the behavior described at the time of the examination when declarations are absent.

9. Source and provenance register

References below identify retained artifacts, not a claim that the private evidence archive has been published. Paths are archive-relative locators. The public report omits account/session identifiers, authentication secrets and raw signed records.

Reference	Source	Use in this report
[1]	R1 evidence/C1/, evidence/C2/, evidence/C3/: request-body.raw, response-body.raw, transport.json; OFFICIAL-RESULTS.json; EXECUTION-FINISHED.json	Exact inputs, verdicts, HTTP results, response fields and completed-case boundary.

Reference	Source	Use in this report
[2]	R1 frozen-protocol/PROTOCOL.md, matrix.json, conditions.json, session templates; PROTOCOL-FREEZE.json; EXECUTION-FREEZE.json and their manifests	Prospective method, expectations, scope, chronology and permitted substitutions.
[3]	R1 frozen-protocol/sources/developer-docs.txt and .html, quickstart, token example and §13; frozen-protocol/sources/sdk-readme.md, scoped-read example	Documentation as preserved at freeze, not a fresh website-status claim.
[4]	Separate POST_EXAMINATION_CLARIFICATION.md, record date 14 September 2026, §§1-6	Attributed post-examination service-author clarifications, documentation acknowledgments and corrected interpretation.
[5]	R1 integration/INTEGRATION-LOG.md and associated recorded exchanges; original REPORT.md; INTEGRITY-REVIEW.json; EVIDENCE-MANIFEST.json	Historical preparation and reconciliation, original interpretation and evidence integrity. The signature-blocker interpretation is corrected in this report, not silently edited in its source.
[6]	LinkedIn correspondence with Praveen Babu, 14 September 2026, 14:23-14:48 Europe/Tallinn	Write-authority correction, confirmation of the remaining factual representations, agreed omission of public version references, and agreement to publish the full unchanged report with attribution and limitations. Correspondence, not new execution evidence.

Evidence identifiers recorded during the original report preparation:

Artifact	SHA-256
Original R1 archive: arcezia-independent-2026-09-14.zip	6ff30937340e36a6a08b7ab3e272b7afac8aa97299134a51fb9fc3c4a1c4656c
Separate clarification: POST_EXAMINATION_CLARIFICATION.md	52910faa7516eb383e0dd801cd82f28ad951b3713e00829a5451c00d1b195ca5
R1 protocol manifest	d431ec9a6c87d249e4e18f0305115e306eb34d589418737c672ac36b59677456
R1 exact request manifest	2a7cee6457fdd0b6c19e3c367d7b7f371fc4053f1bf5d47f3f549ca696863ede

This report is a separate publication document. It preserves the distinction between recorded responses, documentation statements, service-author explanations and the reviewer's bounded conclusions.