

Nikxius

DESIGN-PARTNER EVALUATION · PROPOSED

Acceptance that an operator can repeat

Ten scenarios · Scope, evidence and limitations

Agree the exact nonproduction target, Runtime/source revision, Kubernetes version, identity composition, native comparator, observers and decision owner before testing. Run the same semantic contract on both paths. A skipped or blocked case is not a pass. Preserve redacted operation records, native observations, test output and elapsed operator time.

Required cases

Case	Pass condition	Existing executable evidence
A · Changed action	Old authority/approval cannot approve another target, image or earlier-operation binding.	Runtime A06/A08/A14; rollback adapter/family tests.
B · Expiry/revocation	Expired/revoked authority prevents a later dispatch claim; no false promise of cancelling in-flight native I/O.	A10, including post-claim limits.
C · Stale native state	Changed UID/version/expected field does not receive the old conditional mutation.	A15 native conflict and adapter checks.
D · Duplicate request	Same actor/key/request returns one identity; changed payload or grant conflicts.	A02 actual HTTP/PostgreSQL; response-loss replay.
E · Crash around dispatch	Original attempt/reservation survives; recovery never assumes an unacknowledged transaction rolled back.	A12/A13/A22/A28; actual PostgreSQL restart separately scoped.
F · Lost response	Real possible/committed effect is not blindly repeated.	A16 native response deliberately discarded after I/O.
G · UNKNOWN	Missing sufficient causal evidence remains UNKNOWN, despite a matching current image.	A17/A18 watch-gap and classification checks.
H · Direct bypass	Protected agent cannot perform equivalent protected writes or obtain the Node identity through agreed tested routes.	A24 native permission probes; customer tool/CI/host review still required.

Case	Pass condition	Existing executable evidence
I · Commit vs health	A committed desired-state change remains distinct from rollout observation and application health.	A20; degraded-state injection is labelled.
J · Customer independence	Customer engineer installs, runs, investigates, exports and stops the evaluation using supplied instructions.	Not established by internal tests. Customer must perform it.

The **15 September fresh local rollback run passed ten automated assertions across A-I**; J remains **NOT_OBSERVED**. Real Kubernetes 1.35.0 and PostgreSQL were used with synthetic identity/MFA. An actual SIGKILL preserved the original attempt; response-loss recovery retained one adapter PATCH invocation. A separate crash after the durable claim but before I/O preserved UNKNOWN with no replay. These counters are not packet capture or application-effect proof.

The included evaluation-validation summary records **485 distinct repository tests across scoped runs**, without double-counting repeated browser tests. The actual PostgreSQL restart test used a typed native-provider double; the native rollback/SIGKILL exercise is separate. Detailed original A01–A28 mapping remains in docs/runtime/ACCEPTANCE.md; these lettered scenarios do not renumber it. Neither run proves customer installation, IdP compatibility or application health.

Run and record

Use the quickstart and customer acceptance runner against disposable resources first. Preserve an earlier committed image operation, then exercise the separately granted rollback. Stop concurrent demo/worker processes when required by the runner; native suites share fixtures and run serially.

For every case record: expected result, actual commit/observation, operation/attempt IDs, grant/command binding, original request key, native observation source, injected fault, repeated-write count, evidence path, operator minutes, assistance and limitations. Fault injection after native I/O is not a claim that every physical network partition was reproduced.

For J, hand a clean environment and the kit to an engineer who did not author the implementation. Count interventions and failed commands. Verify that the engineer can explain why a timeout may remain UNKNOWN and why a new request key is not a safe retry strategy.

Acceptance and purchase are separate

Correct UNKNOWN behavior can pass a test and still impose unacceptable operating cost. Record pass/fail/blocked per case, compare native-stack burden and make a separate continuation decision. Agree the contractual milestone before work; this template creates no automatic acceptance or invoice obligation.