



DESIGN-PARTNER EVALUATION · PROPOSED

Make the native stack the benchmark

Comparison guide · Same operation, same failure contract

Nikxius earns a place only if it removes a meaningful deficiency or operating cost. Use the customer’s strongest credible native path, including the engineering already invested in it. Do not compare a controlled Runtime with a deliberately unrestricted kubectl script.

Assemble the real alternative

Existing layer	Include in the comparison
Identity / authorization	Workload identity, token lifecycle, Kubernetes RBAC, named-resource permissions and actual alternate paths.
Native admission	Validating admission policy/webhooks and their exact field/state constraints.
GitOps / CI/CD	Current approved writer, protected review, repository state, controller reconciliation and emergency procedures.
Native API	UID/resourceVersion/field tests, returned persisted object, native events/audit/watch retention.
Workflow / retry	Durable run identity, approvals, dispatch state, retry policy, restart and unresolved outcome ownership.
Evidence / operations	Existing logs, audit exports, dashboards, alerts and operator recovery instructions.

If GitOps owns the field, assess the source-of-truth change and resulting controller effect. A direct patch demonstration cannot claim equivalence merely because both eventually show the same image. Record differences in the declared operation and attribution boundary.

Test identical questions

Can old approval authorize changed bytes? What prevents stale-state mutation? What survives a crash? Who determines whether retry is safe after a timeout? What evidence establishes the original effect? Can an agent bypass the path? What remains unresolved after an unhealthy rollout?

Apply acceptance cases A–J to both approaches, documenting cases that are not semantically comparable. Preserve errors and unresolved outcomes. Native capability that meets a case is a pass, not something to discount because it lives across several tools.

Measure effort honestly

Measure	Capture for each path
Setup	Prerequisites, configuration minutes, custom integration time and people involved.
Code/config	Changed non-generated lines and files, plus existing systems reused. Lines alone are not quality.
Operation	Approvals, manual interventions, normal-run time and incident/recovery minutes.
Failure	Duplicate native writes, stale-action acceptance, missing evidence, duration and ownership of UNKNOWN.
Ongoing burden	Credential rotation, upgrades, retention, backups, monitoring and support dependency.

Time both from the same start condition. Separate founder implementation, customer configuration, custom engineering and recurring operation. Publish neither invented savings nor a composite score that hides an unacceptable failure. The checkout's docs/design-partner/NATIVE_STACK_COMPARISON.md contains the reproducible comparator and run-evidence references; **customer-specific savings and superiority remain unmeasured** until the customer baseline is run.

Decide

The 15 September local native reference passed **six component checks**, including native authorization/admission, conditional writes, revocation and response-loss recovery from retained history. Its single-process file journal is not a complete distributed workflow, GitOps or customer-identity composition. The result demonstrates useful native capability; it does not establish the cheaper complete customer path. See the included native comparison reference and its `result.json` for the exact scope.

Choose the native stack if it meets the contract economically. Continue with Nikxius only when the customer records a concrete advantage, accepts its additional operational component and identifies a production decision and budget owner. If the path cannot be made mandatory, the initial boundary is not established.