

Nikxius

DESIGN-PARTNER EVALUATION · PROPOSED

Questions before an evaluation

Technical FAQ · Scope and honest limits

Why not use RBAC and admission?

Use them. They are part of the baseline and the Nikxius deployment. RBAC constrains the credential's resource/verbs; native admission can constrain writes. Evaluate whether the composed stack also binds the exact task, native baseline, durable attempt, unresolved outcome and operator recovery economically. Nikxius has no automatic claim to win.

Does this replace GitOps, Temporal or the IdP?

No. Existing identity, workflow and authoritative writer ownership remain. The current typed adapter calls the registered Kubernetes API; it is not a generic GitOps or Temporal integration. A competing controller must be reconciled in the design, not ignored during the demo.

Is this a published SDK or hosted service?

The Node client is exported by the private control-plane workspace. Runtime source is private and UNLICENSED without agreed evaluation terms. Execution requires no Nikxius SaaS control plane. The public website is not a customer execution endpoint.

Can the agent still bypass it?

Yes, if it retains another equivalent writer or access to the Node's credential. Review direct permissions, secrets, token creation, exec, CI/tools and host paths. Local denial tests establish the tested paths, not every possible customer route.

What if the response is lost?

The attempt may have executed. Preserve UNKNOWN and the original identity; use read-only reconciliation. Do not issue another mutation simply because a timeout occurred. Sufficient retained native history can establish the original transition; current matching state alone cannot.

Can UNKNOWN stay forever?

Yes. Missing or compacted history can prevent resolution. The task/resource reservation remains. Customer operational ownership and separately governed emergency access must handle that obligation. A UI acknowledgement does not manufacture evidence.

Is execution exactly once?

There is no universal exactly-once claim. Nikxius preserves request/attempt identity, avoids blind replay and uses native preconditions. PostgreSQL and Kubernetes are separate systems; delayed requests and missing history remain explicit limits.

Does revocation cancel an action?

It blocks later checked admission/dispatch claims. It does not retract a native request already accepted or guarantee every delayed sender has stopped. The exact native conditional request is a separate guard.

Does a signature prove a correct outcome?

It can establish integrity and Node-key attribution of the retained record. It does not establish business correctness, independent native truth or a complete record where history was lost. Unsigned source-qualified exports remain possible.

What proves that the application recovered?

Application-specific health checks owned by the customer. A committed Deployment spec change and controller rollout observation are distinct from application health. The rollback is itself a new consequential operation and can fail or remain unknown.

Is it production-ready or independently certified?

Local conformance and this proposed nonproduction evaluation do not establish production acceptance, external penetration testing or certifications. Customer IdP, native policy, backups, retention, support and security review must be validated for the intended environment.