WHITE PAPER · SECURITY & TRUST SERIES

Adversarial by Design: Security Testing & Data Protection in an AI-Native Banking Platform

The security architecture NGDB.AI runs in production — and the VAPT program we run against ourselves, with the same AI workforce that builds the platform. Written as a reproducible methodology for engineering teams, and as a diligence checklist for anyone deciding whether a platform deserves to move their money.

5 layers
identity, transport, data, money path, document integrity

4 cadences
per commit · per deploy · monthly red-team · quarterly VAPT

0 plaintext hops
TLS at the edge, mutual TLS inside, private data networks

Tested, not trusted

A platform that asks businesses to route money through it has no right to be trusted. It has a duty to be tested — continuously, adversarially, and on the record.

Most fintech security pages are written by marketing: a padlock icon, the word “bank-grade,” and a list of acronyms. This paper is written from the other side. It describes the security architecture NGDB.AI runs in production and — more unusually — the vulnerability assessment and penetration testing (VAPT) program we run against ourselves. We publish it because the methodology is reproducible: any engineering team can implement this program, and any customer or investor can use it as a checklist against any platform that wants to move their money.

The threat model: what a trade-finance platform must defend

A trade-finance and digital-banking platform concentrates four things attackers want: a money path (payment guarantees, captures, settlements), identities (verified businesses and their credentials), documents (bills of lading and invoices whose integrity collateral depends on), and data (cash-flow, banking, and KYB records). Each gets its own defensive layer — and each layer gets attacked on a schedule, by us.

© 2026 NGDB.AI, Inc. · ngdb.ai/white-papers

Five layers, from identity to document integrity

1 · Identity and access

OAuth 2.0 and OpenID Connect with PKCE on every client, short-lived access tokens with refresh-token rotation, FIDO2 support for strong authentication, and an eight-role RBAC model enforcing least privilege. Administrative actions are maker-checker: prepared by one identity, executed by another, logged immutably. No shared accounts, anywhere, ever.

2 · Transport security

Every hop is encrypted — not just the edge. TLS at the perimeter with HSTS and modern cipher policy; mutual TLS between internal services, so a workload proves its identity cryptographically before another workload will speak to it; managed certificate rotation so expiry is an event, not an incident; signed webhooks so consumers can verify origin. Databases and caches are reachable only on private networks — there is no public database endpoint to scan.

3 · Data security

Encryption at rest everywhere, with field-level AES-256-GCM on sensitive payloads above it. A tokenization-first posture keeps high-value identifiers out of the blast radius: core services handle tokens, and the systems holding the underlying values are small, isolated, and separately keyed. Keys live in a managed KMS with rotation; secrets live in a secrets manager and never in code, configuration, or logs — pipelines scan for them on every commit. Data is minimized by design: tiered KYB collects exactly what each level of platform activity requires, and no more.

4 · The money path

Everything that moves funds is built on idempotency keys and exactly-once event processing, posted to an immutable double-entry ledger, and screened by an in-line fraud gate in under 100 milliseconds. Irreversible operations are human-gated: an agent or service can prepare a settlement, but a control boundary — not a convention — stands between preparation and execution. Every state transition is an auditable event.

5 · Document integrity

Trade documents are anchored with dual hashes — SHA-256 and SHA3-256 — to a permissioned ledger, so tampering with a bill of lading after the fact is not merely forbidden but cryptographically evident. Two independent hash families means a break in one algorithm does not break the evidence.

© 2026 NGDB.AI, Inc. · ngdb.ai/white-papers

The self-run VAPT program

The unusual part of our program is not the tooling — it is the operating model. The same principle that governs our engineering (the builder is never the verifier) governs our security testing: attack agents are independent of build agents, with their own mandate, and findings are tracked with the same evidence discipline as code. A finding without a reproduction is not a finding; a fix without a regression test is not a fix. The red team never reports to the builder.

Per commit — scan everything, gate the sensitive

Static analysis, dependency scanning against known CVEs, and secret scanning run on every change. Anything touching authentication, money movement, or sensitive data additionally passes a security review gate before merge.

Per deploy — security regressions caught like failing tests

Dynamic scanning runs against a staging twin: authentication flows, session handling, token expiry and rotation, header and transport posture. A regression in security configuration is caught the way a failing test is caught — before traffic.

Monthly — scenario red-teaming from our own threat model

Where generic scanning ends, domain-specific adversarial thinking begins: money-path abuse (replay, double-capture, idempotency bypass), object-level authorization sweeps across tenant boundaries, privilege escalation across the role model, onboarding spoofing against KYB verification, and abuse of negotiation and offer flows. AI agents are well suited to this work — they enumerate systematically, they do not get bored, and they write up every attempt including the failures, which is exactly what an evidence trail needs.

Quarterly — the full VAPT cycle, written to be shown

Reconnaissance as an outsider would do it, a threat-model refresh, exploitation attempts in the staging twin under written rules of engagement (never production data), severity-scored findings, and fix SLAs tracked like uptime: critical findings block the next deploy, high within days, the rest on a dated schedule. Every resolved finding leaves behind a permanent regression test. The quarterly report is written for a bank partner's security review, an investor's technical diligence, or a customer's vendor-risk questionnaire — with sampled evidence: reviewers pick the finding, we reproduce the trail.

Formal third-party attestations have their place — they arrive naturally with sponsor-bank programs and enterprise contracts, and our posture is built so external assessors verify a running program rather than trigger the creation of one. The first penetration test a partner commissions against you should find a platform that has already been attacked, monthly, by someone with source-code-level knowledge and no incentive to be gentle.

© 2026 NGDB.AI, Inc. · ngdb.ai/white-papers

Eight questions to ask any platform that moves your money

1. Is every hop encrypted — including service-to-service traffic inside your perimeter?
2. Where do raw card numbers and personal identifiers physically live, and which systems can read them?
3. When was your last penetration test, who ran it — and does your red team report to the people who built the system?
4. Can you reproduce the complete audit trail for any money movement I pick?
5. What are your fix SLAs when a vulnerability is found, and what blocks a deploy?
6. Where do secrets live, and what stops one from reaching a log file?
7. Can any single actor — human or AI — move money alone?
8. How do I report a vulnerability to you, and what happens when I do?

A platform that answers these in writing is telling you about its architecture. A platform that answers with a padlock icon is telling you about its marketing.

> **Trust is not claimed. It is tested — continuously, adversarially, and on the record.**

Responsible disclosure

We would rather hear it from you than find it ourselves. Security researchers can reach us at kushal@ngdb.ai; reports get a human response, a reproduction attempt, and a fix timeline — and we credit researchers who report in good faith.

About the author · The series

Kumar Kushal is Founder & CEO of NGDB.AI and previously founding CTO of the Reliance Jio digital ecosystem (350M+ users, PCI-DSS Level 1 with zero major breaches).

1 · Trade Without Banks: The Card-Rail Business Model for Global Trade Finance
2 · The Financed Marketplace: B2B Commerce Where Every Order Carries Its Own Credit
3 · The Agentic Enterprise: Building a Banking Platform with an AI Agent Workforce
4 · Governed Velocity: AI-Assisted Engineering Governance in Regulated Fintech
5 · Adversarial by Design: Security Testing & Data Protection (this paper)

All papers: ngdb.ai/white-papers · Contact: kushal@ngdb.ai

© 2026 NGDB.AI, Inc. · ngdb.ai/white-papers