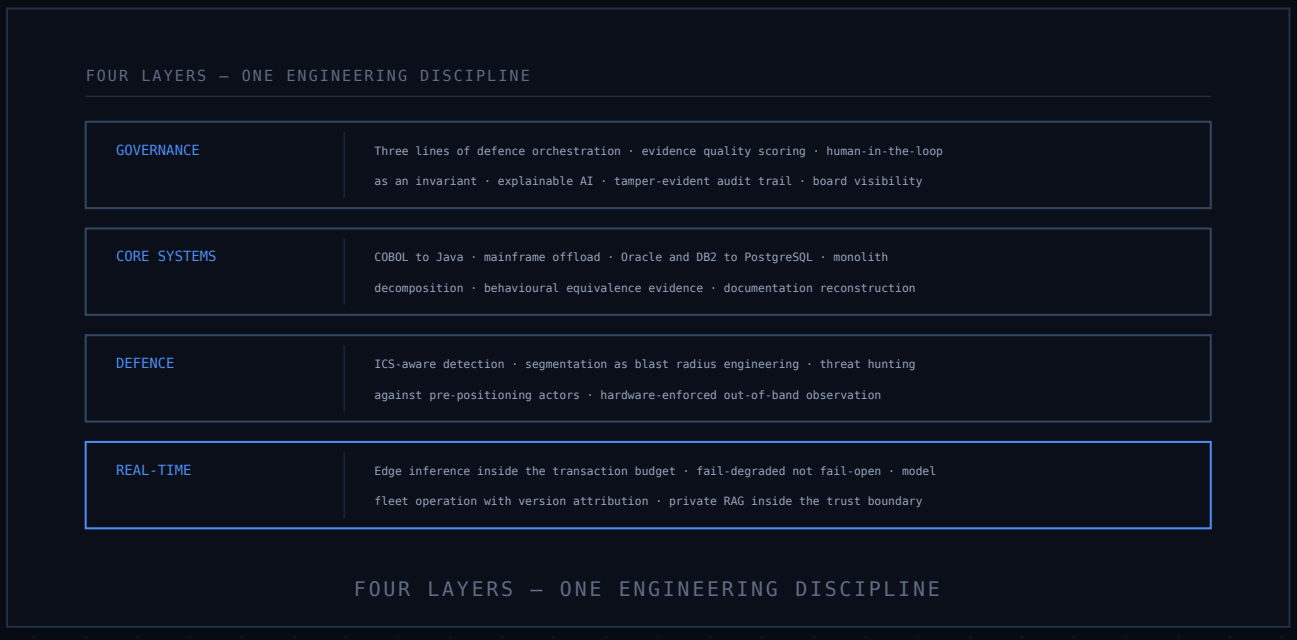
Most Vendors Sell You One Layer and Call It Modernisation.

A bank's problems do not arrive in the shape of a vendor's product category. The core is undocumented COBOL and the migration cannot be validated. The AI in the governance process cannot name who decided. The adversary the regulator is warning about is already inside somebody's network using nobody's malware. And the new payment rail has quietly removed the reversal window that half the fraud controls assumed.

These look like four procurements. They are four faces of one discipline: **build the thing so that the evidence exists, and so that the failure mode is the safe one.** That is what this practice does, and this document is the map of where we do it.

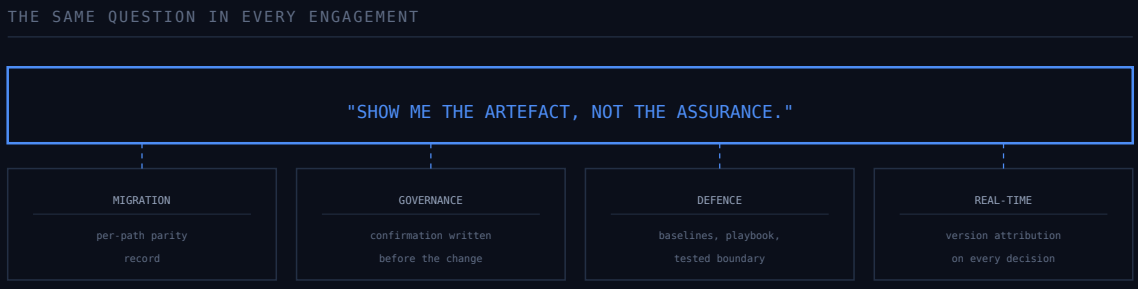
Every engagement referenced here is real. Where a client is not named, it is because the engagement is bound by non-disclosure, the platform is a third party's product, or the sector does not publicise its defensive work. Each has a full case study behind it, and we would rather walk your team through the engineering than the brochure.

THE MAP



WHAT ACTUALLY CONNECTS THEM

It would be easy to claim these engagements are connected because we did all of them. That is a statement about us and of no use to you. **They are connected because the same question decides whether each one succeeded.**



A COBOL migration is judged by whether a parity record exists for every path. An AI governance platform is judged by whether the confirmation was written before the state changed. A defence programme is judged by whether the baselines and the playbook survive the consultant's departure. A fraud fleet is judged by whether you can name which model version made a given decision eleven months later.

In every case the deliverable that matters is not the system. It is the artefact the system produces — the thing your model risk function, your internal audit team, or your examiner can inspect without taking anyone's word for anything. Build for that from the first sprint and the system tends to come out right. Build the system and add the evidence afterwards, and you discover at month eighteen that the evidence cannot be reconstructed.

THE COMMON FAILURE

The control exists in the interface. It is a warning, a status field or a policy, and it does not survive contact with the API.

THE COMMON FIX

The control lives in the domain, enforced by ordering or by architecture. Not because someone remembers — because the path does not exist.

THE COMMON TEST

Try to break it deliberately. A control nobody has attacked is a claim, and claims are what get discovered during an examination.

LAYER 01

Core Systems

The estate that runs the bank, written by people who have retired, documented against an architecture that no longer exists. Everyone knows what it costs to leave alone. The reason it survives is that nobody can prove a migration would be safe.

01

COBOL to Java and Spring

Deterministic generation — the same source yields the same output every run. Reproducible by construction, which is the precondition for validating anything at all.

02

Mainframe offload

Batch moved to distributed infrastructure with the system of record intact throughout, and timing treated as a correctness property rather than a performance target.

03

Oracle and DB2 to PostgreSQL

Reconciled row by row. Empty-string semantics, decimal scale, collation and isolation behaviour made explicit rather than inherited.

04

Monolith decomposition

Boundaries recovered from transactional behaviour rather than the org chart, under a strangler pattern with rollback available at every phase.

05

Behavioural equivalence

Every migrated path replayed against the original. Output, intermediate state, write sequence and error behaviour compared. No path accepted on sampling.

+

Documentation reconstruction

Specifications rebuilt from source and validated by prediction — an engineer who has never seen the estate reads the doc and forecasts behaviour.

Delivered: a Tier-1 Thai commercial bank. All four migrations executed on one estate, under one evidentiary standard, without a big-bang cutover. Most vendors perform one of the four and call it modernisation. The engagement is under non-disclosure, so the client is not named and metrics are not published.

THE QUESTION THAT SEPARATES VENDORS

Is your code generation deterministic? If the same input produces different output on different runs, how does my model risk function validate it — and what exactly do you regenerate when a reviewer challenges a transformation eleven months later?

LAYER 02

Governance and AI Accountability

Every institution deploying AI into a governance process will eventually be asked to prove a human was genuinely accountable for what the system decided. Most will discover at that moment that their human-in-the-loop was a screen rather than an invariant.

01
Three lines orchestration
Control testing, risk and compliance, and internal audit coordinated as one system — so the three lines do not silently collapse into one.

02
Evidence quality scoring
Evidence scored before anyone may rely on it. Below the floor, the reliance action is unavailable rather than discouraged.

03
Reliance with auto-suspension
Reliance bound to a specific evidence version, so it revokes itself when its basis changes. Nobody has to notice.

04
HITL as an enforced invariant
The confirmation record is written before the state changes. If the write fails, the state never changes.

05
Explainable AI
Reasoning captured at decision time, not generated on request. A record, not a story about the decision.

+
Tamper-evident trail
Append-only with a hash chain. Trust in the record does not rest on trust in the administrator.

Delivered: an assurance intelligence platform for a prudential regime that requires exactly this demonstration, built as engineering partner. Not named — the platform is a third party's product and the engagement is under non-disclosure. The regulatory machinery translates directly: the regime's evidence and reliance requirements map onto SR 11-7 model risk validation and the 2023 Interagency guidance on third-party risk.

THE QUESTION THAT SEPARATES VENDORS

What happens when the human confirmation write fails? If the action still proceeds and the failure is logged, your human-in-the-loop control is a user interface convention, and it will be identified as one at the least convenient possible moment.

LAYER 03

Defence

The OCC's Spring 2026 Semiannual Risk Perspective names foreign state-sponsored actors as a continuing threat to the federal banking system. CISA, NSA and the FBI assess that state-linked actors are pre-positioning inside IT networks specifically to reach the operational systems behind them. **This is not a hypothetical section.**

01
ICS-aware detection
Detection that parses what a command means, not merely that a connection occurred. Passive taps only — the monitor cannot interfere with what it monitors.

02
Segmentation as blast radius
Designed on the premise that the enterprise network is already compromised. The only control on this list that still works after prevention has failed.

03
Threat hunting
Going to look for an adversary who has triggered no alerts, because against a pre-positioning actor, waiting is the same as not looking.

04
Behavioural baselining
What normal looks like on this specific estate. Anomaly detection without a baseline is a search for whatever the searcher expected.

05
Hardware-enforced observation
An FPGA co-processor with its own clock and memory. Host compromise no longer implies detection compromise.

+
Capability transfer
The playbook and baselines built to be operated by the client. Capability that departs with the consultant is not capability.

Delivered: a national power grid operator, alongside a NERC-CIP-certified partner — ICS-aware IDS, segmented architecture and a threat hunting playbook, deployed without an outage. Separately, an FPGA-based detection system on a Xilinx ten-layer board: **95% detection accuracy across threat categories, sub-10-microsecond latency, ransomware patterns identified within 30 seconds, zero host overhead** — measured in a simulated environment against defined attack scenarios.

THE QUESTION THAT SEPARATES VENDORS

What can an attacker holding a valid corporate credential reach in your payments environment today — and what would your detection layer report on a host whose kernel is already under adversary control?

LAYER 04

Real-Time and AI on Your Own Data

Instant payments were sold to your institution as a product feature. They arrived as an architecture problem, because every fraud control that assumed a reversal window quietly stopped being a control.

01
Edge inference
Scoring inside the transaction's own latency budget rather than competing with it. No round trip to be slow, and none to fail.

02
Fail-degraded, not fail-open
Network loss costs the freshest intelligence, not the ability to decide. There is no timeout for an attacker to induce.

03
Model fleet operation
Version attribution on every decision, staged rollout, tested rollback. Mixed versions are the steady state, not an anomaly.

04
Private RAG
Retrieval and inference inside your boundary. The documents do not leave, because the network path does not exist.

05
Permission at retrieval
Filtered before the search, not applied to the answer. What the model cannot retrieve, it cannot leak.

+
Grounding and refusal
Every claim traced to a passage; unsupported spans flagged. A model that cannot say "I do not know" will say something else instead.

Delivered: real-time fraud detection on point-of-sale terminals — inference on constrained hardware already deployed in the lane, evaluated on recall at a fixed false-positive rate rather than on accuracy. The engineering transfers directly to FedNow and RTP, where settlement is final and detection after the fact is a report about money that is gone.

THE QUESTION THAT SEPARATES VENDORS

What do your terminals do when the scoring service does not answer in time? And where were your documents when they were embedded — not where answers are generated, where the content was when the vectors were produced?

MAPPED TO YOUR SUPERVISORY REALITY

The OCC under the current administration is supervising on a risk basis and has been explicit about safeguarding rather than smothering. **That changes the pitch: this is not about helping you pass an examination. It is about taking risk safely enough to move faster than the institution next door.**

YOUR CONSTRAINT	WHAT WE BRING	MAPS TO
Model risk validation	Deterministic generation, captured reasoning chains, version attribution on every decision, per-path parity evidence.	SR 11-7
Operational resilience	Incremental cutover with rollback at every phase; segmentation designed for a compromised enterprise network.	OCC OP RESILIENCE
Third-party risk	Evidence quality scoring and reliance gating applied to vendor assurance rather than accepted on attestation.	INTERAGENCY TPRM 2023
State-sponsored actors	Detection and hunting built against an adversary who brings no malware, uses valid credentials, and waits.	OCC SRP SPRING 2026
Ransomware recovery	Detection at thirty seconds rather than thirty minutes changes what your recovery objective must withstand.	FFIEC
Irrevocable rails	Fraud decisions inside the transaction budget, with no fail-open path to induce.	FEDNOW / RTP
Data residency	Retrieval and inference inside the boundary; sovereign infrastructure where the public endpoint is not an option.	RESIDENCY
Audit defensibility	Append-only hash-chained trails and inspectable artefacts across every layer above.	INTERNAL AUDIT

HOW WE ENGAGE

BUILD

Outcome-owned delivery pods. We own the outcome, not a headcount line. Pods are priced per month with a three-month minimum.

RESCUE

Programmes that have stalled — usually because the evidence cannot be produced, not because the code does not work.

SCALE

Capability that stays after we leave. Baselines, playbooks and documentation built to be operated by your own team.

Three entities, one practice. **Tech4Biz Solutions** delivers the engineering. **CodeNuk** is the deterministic generation platform behind the migration work. **AuditTrax Labs**, our US entity in Atlanta, performs technical due diligence — which is frequently how these conversations start, because the same discipline that produces evidence is the one that knows what to look for when the evidence is missing.

Show me the artefact

Every engagement in this document is judged by the same standard: whether the thing your examiner, your model risk function or your internal audit team can inspect actually exists, and whether it was produced by the system rather than assembled afterwards to explain it. That is the whole practice. Everything else is implementation detail.

WHERE THIS GOES NEXT

Each layer in this document has a full case study behind it — the challenges we actually hit, how we resolved them, and how we proved it worked. They are more interesting than this overview and considerably less comfortable, which is the point.

Tell us which layer is the one keeping you up, and we will send the one that matters.

TECH4BIZ SOLUTIONS | CODENUK | AUDITTRAX LABS

tech4bizsolutions.com codenuk.com auditraxlabs.com

contact@tech4biz.io