Project Acumen

A Safe, Predictable, Cryptographically Audited Query Compilation Framework

Investment Teaser — Confidential

Reddy & Reddy Family Office (i.f.) — GIFT City, IFSC, Gujarat

July 19, 2026

This teaser summarises, in plain language, an intellectual-property (IP) asset whose performance has been measured under load on two hardware tiers — including instrumentation running inside *the engine itself — and whose value has been appraised under one strict rule:* ***only functions the engine is measured performing are given a dollar value****. Every headline figure below traces back to either (a) raw benchmark telemetry of July 18, 2026, or (b) published market data on what the world currently pays for those functions. Where a number carries a caveat in the underlying analysis, the caveat is repeated here.*

1. The Idea in Plain English

Every time an app looks something up — a doctor pulling a patient's lab history, a bank checking a transaction, a dashboard refreshing a chart — a *query* is sent to a database. Databases answer billions of such queries per day, and every query consumes real, physical resources: processor time, memory, and electricity inside a data centre. Every query is also an attack surface, and every change to the data is something a regulator may one day ask you to prove.

Project Acumen is a translation, safety, and integrity layer that sits between applications and the database. Think of it as a combination of:

- **A disciplined translator.** It converts high-level requests (written in the healthcare-industry standard called FHIR) into safe, predictable database instructions — the same request always produces the identical instructions, at a cost that is measured and does not drift with load.
- **A bouncer at the door (the "Safety Kernel").** Before any request is allowed to touch the database, it is checked for traps: requests that are too deep, circular, or complex enough to hang or crash the system are rejected in under a tenth of a second — before they can consume expensive resources. In dedicated adversarial testing, the bouncer turned away **50,302 out of 50,302** hostile requests without wrongly blocking a single legitimate one.
- **A notary in the vault.** Every change to the data is committed together with a cryptographically signed, hash-chained audit entry — a tamper-evident record with the integrity guarantees of a blockchain, at ordinary database speed. A record cannot be saved without its audit entry, and history cannot be quietly rewritten.

The practical consequence: **the database tier gets a measured, bounded cost per query, a measured shield against hostile input, and a measured tamper-evident audit trail — three properties enterprises currently buy as separate products.**

2. What Has Actually Been Measured

The claims below are not projections. They come from instrumented benchmark runs of July 18, 2026, on two hardware tiers — a production-scale cloud server holding 6.46 million clinical records and a consumer laptop — with every figure reproducible from the raw telemetry files, and with resource behaviour (processor, memory, garbage collection) measured *inside* the engine process itself.

Technical finding	**In plain terms**	**Why an investor should care**
100% absorption of 50,302 hostile requests across two machines; zero false positives	Every attack in a labelled adversarial corpus was turned away; no legitimate request was blocked	The security function is demonstrated, not promised — the basis of the valuation's first lever
Rejecting an attack costs $\sim 5\times$ less than serving a real query (server hardware); a sustained attack storm did not slow legitimate traffic	Attackers pay more than they cost; customers don't notice the battle	Favourable unit economics for an embedded security layer
Every write produces a signed, hash-chained audit entry, atomically, at 21–113 ms	Ledger-grade tamper evidence at ordinary database speed	The compliance function is demonstrated — the valuation's second lever
Per-query processor cost is *flat* from 1 to 50 simultaneous users ($\approx$12–17 ms on cloud hardware)	Load changes waiting time, never the price of the work itself	Capacity planning becomes arithmetic: cost per query $\times$ queries
$O(\log n)$ scaling: an **83.6**$\times$ larger dataset raised worst-case indexed latency only **2.43**$\times$ — and made several query types *faster*	Data can grow enormously while response times barely move	The economics improve as customers grow
24–83 ms 95th-percentile latency on indexed queries; slowest-to-typical ratio of 1.10–1.30$\times$ (industry norm: 5–10$\times$)	Nearly all lookups answer fast, and the slowest are barely slower than the typical	Tight, honest service-level guarantees become possible
Garbage-collection cost measured in-process: $\approx$1 ms per query at every load level; largest single pause 8–11 ms on server hardware	The engine's housekeeping is a small, constant, bounded tax — never a 100 ms-class "hiccup"	Predictable behaviour claimed from measurement, not from marketing
Zero storage reads on the query path; page 500 of results costs the same as page 1 (1.02$\times$)	Everything hot runs from memory; deep dashboards cost nothing extra	A genuine, measured efficiency for analytics workloads
Deterministic compilation: the optimizer's work counters reproduce *exactly* across machines and runs	Run it today on one machine, tomorrow on another — identical behaviour	Strong evidence of engineering quality; simplifies due diligence

Table 1: Directly measured results from the Empirical Performance Analysis Report (benchmark date July 18, 2026) and the Safety-Kernel Empirical Analysis Report. All values reproducible from cited JSON telemetry.

2.1 The Efficiency Question — Stated Honestly

The engine's *absolute* cost is measured with precision: a flat ≈12–17 ms of server processor time per query on cloud hardware, at any level of concurrency. What has **not yet been measured** is that cost *relative to an incumbent stack* doing equivalent work — and an absolute cost is not a savings rate. The appraisal therefore assigns the compute-efficiency lever **a value of exactly zero**, while stating what the published literature supports: mainstream database-access layers burn 6–9× the processor time of hand-tuned queries; Google measures a cross-cutting "datacenter tax" at 22–30% of all its fleet's cycles; and compiled query engines (Microsoft's Hekaton, HyPer, Spark) have repeatedly delivered ten-fold gains against interpreted stacks when measured. Those priors justify *funding the head-to-head benchmark*; they do not substitute for it.

3. How the IP Creates Value

The value-creation logic is a three-link chain, each link simple:

1. **Enterprises already pay for these functions.** The world spends roughly **$8.0 billion** a year on web-application firewalling (the "bouncer" function; mid-consensus of eight analyst estimates) and roughly **$8.5 billion** on database security (whose audit-integrity slice is the "notary" function) — a combined **$16.5 billion** function-market pool, growing 12–19% a year.
2. **Acumen measurably performs the database-tier slice of those functions.** Not all of either market is database-tier work, and the engine does not replace every feature of either product category. The model therefore addresses only a 10–25% slice of each market, and credits the engine with only 30–70% displacement within that slice — fractions bounded by the measured record on both sides (100% absorption and zero false positives on one side; seven documented guard gaps and ∼41% instrumentation coverage on the other).
3. **Whoever owns the IP can charge for the displacement.** An adopter who deploys Acumen keeps most of the displaced spend as its incentive to adopt; the IP owner captures a royalty share (conventionally 25–33%) of the value created. Alternatively, an outright sale to an operator is anchored on the full displacement figure.

In short: **the asset performs, in one engine, two security functions the world already pays $16.5B a year for** — and holds an unpriced option on the compute bill itself (§6).

4. The Economic Model — Four Numbers Multiplied Together

The appraisal rests on a single, auditable formula — a *relief-from-cost* discounted cash flow — in which each factor is either a published figure or bounded by a benchmark measurement:

$$\text{IP Value} = \sum_{t=1}^{10} \frac{\underbrace{\text{Pool}_t}_{\substack{\text{function-market}\\ \text{spend, yr } t}} \times \underbrace{s}_{\substack{\text{database-tier}\\ \text{share}}} \times \underbrace{d}_{\substack{\text{measured}\\ \text{displacement}}} \times \underbrace{c(t)}_{\substack{\text{adoption}\\ \text{fraction}}}}{(1+15\%)^t}$$

In words: take what the world pays for these two functions each year, keep only the database-tier share, credit only the fraction of the function the engine measurably performs, assume only a slow, realistic uptake (an S-curve saturating at just **10% capture by year 10**), and then discount every future dollar at a demanding 15% per year. No terminal value is added — everything beyond year 10 is given away for free as an extra margin of conservatism.

4.1 Worked Illustration (Base Case)

Year 1: \$16.5B function pool × 15% database-tier share × 50% displacement = \$1.24B addressable; × 0.8% adoption ≈ \$10M captured, worth ≈\$9M today. Repeating this for each of ten years (pool growing 10%/yr — below every published growth estimate — and adoption ramping along the S-curve) and summing the discounted values gives the Base-case result. By year 10 the annual displaced spend reaches ≈\$290M; discounted, the stream sums to **\$535 million**.

4.2 Valuation Range

Scenario	Lever A threat absorption	Lever B audit integrity	Yr-1 addressable	IP value (NPV)
Low — defensible floor	\$104M	\$110M	\$495M	**\$214 Million**
Base — central estimate	\$259M	\$275M	\$1.24B	**\$535 Million**
High — optimistic ceiling	\$605M	\$514M	\$2.59B	**\$1.12 Billion**

Table 2: Appraised value of the IP as the present value of function-market spend displaced over ten years, discounted at 15%. The compute-efficiency lever is excluded (carried at zero; see §6).

4.3 Two Ways to Read the Number

- **Outright sale** to an operator who realises the full displacement: anchored on the total economic value — **\$535M** in the Base case.
- **Licensing / royalty structure**: the owner captures a conventional 25–33% split, i.e. **\$134M–\$176M** of owner-capturable value in the Base case, with licensees keeping the remainder as their incentive to adopt.

4.4 Stress Test

Because reasonable people will disagree about the discount rate and the adoption ceiling, the Base case is stress-tested across both:

Discount rate ↓ / Adoption ceiling →	5%	10%	20%
10%	\$362M	\$724M	\$1.45B
15% *(chosen)*	\$267M	**\$535M**	\$1.07B
20%	\$202M	\$404M	\$808M
25%	\$156M	\$312M	\$624M

Table 3: Base-case NPV under alternative assumptions. Even at a punitive 25% discount rate and half the assumed adoption, the measured-function core remains a nine-figure asset — before the unpriced lever below.

5. How the Appraisal Is Constructed

Three design choices define the valuation methodology:

1. **Measured functions, not projected revenue.** The model contains no revenue-capture assumptions, no comparable-company multiples, and no strategic premia. A function is

priced only if the empirical record measures the engine performing it, and only against what the market currently pays for that function.

2. **The measured limits cap the model.** The displacement fractions are bounded by the same reports that support the claims: seven adversarial patterns that pass the guards (one running to a 20-second timeout at production scale), ∼41% server-side counter attribution, memory pressure under attack storms, and rejection economics that invert on consumer-grade hardware. The Low scenario exists to bound these from below.

3. **Every input is externally checkable.** Function-market figures are published analyst estimates (with their full ranges disclosed); benchmark figures are reproducible from raw JSON telemetry via one-line commands cited in the engineering reports; and the model's remaining levers (adoption ceiling, discount rate) are exposed in the sensitivity grid above rather than buried in the result.

6. The Unpriced Lever — Compute Efficiency

The largest potential value driver is deliberately **not** in the headline. If a head-to-head benchmark demonstrates that Acumen answers equivalent queries for less compute than incumbent stacks, the conversion machinery is already fixed: against the four major cloud providers' $437.7B year-1 infrastructure pool (published FY2025 capital expenditure plus running costs) and its 8–15% database slice, **each percentage point of demonstrated efficiency converts to ≈$217M of net present value** under the same conservative adoption and discounting as the core appraisal.

Illustratively — *not* as appraised value — a benchmark landing anywhere in the literature-supported 10% / 25% / 50% band would evidence **$2.2B / $5.4B / $10.9B** of additional value, dwarfing the measured-function core. The gating evidence item is one well-defined experiment: identical workload and hardware, Acumen versus an incumbent production stack and versus hand-tuned queries, with processor time attributed to the server process on all sides — the same instrumentation discipline the July 2026 analysis already applies. **Producing that benchmark is the highest-return action available to the owner of this asset.**

7. Further Upside Not Counted

- **Automating the database administrator.** A schema-aware Dynamic Indexing Engine removes routine human index management — a measurable labour-cost saving for large IT estates, not yet benchmarked in labour terms.
- **Capacity-planning determinism.** A flat, measured cost per query and compressed latency tails permit tighter provisioning headroom than variance-heavy stacks — quantified as behaviour, not yet as dollars.
- **Adjacent markets.** The dedicated API-security market (≈$1.5B, fast-growing) overlaps the bouncer function and is excluded from the pool; regulated-industry premium pricing for ledger-grade integrity (finance, government records) is likewise uncounted.

8. Summary for the Busy Reader

- **What it is:** a query translation, safety, and integrity engine, measured to reject 100% of a 50,302-request attack corpus with zero false positives, to sign and chain every data change at ordinary database speed, and to hold a flat, bounded cost per query at any load — on a corpus of 6.46 million records.
- **Who pays:** enterprises that currently spend ∼$16.5B a year on the two security functions this engine performs at the database tier.
- **What it is worth:** **$214M floor / $535M base / $1.12B ceiling** as function spend displaced over 10 years (15% discount, no terminal value); **$134M–$176M** owner-capturable under a conventional royalty split.
- **The option on top:** a compute-efficiency lever carried at *zero* until one benchmark is run — worth ≈$217M per demonstrated percentage point, with published literature supporting a 10–50% band ($2.2B–$10.9B illustrative).
- **Why believe it:** conservative by construction — only measured functions are priced, the measured limits cap the model, adoption is held to a 10% ceiling, growth below every published estimate, and every figure is reproducible from raw telemetry or public market data.
