The Killer Agentic AI Use Case for Banks and Insurers

Near-Zero Governance Overhead, Weeks to First Value, and Permanent Market Dominance

Rajesh Iyer
iyer70@gmail.com

Abstract. You have never seen your efficient frontier. Every financial institution runs two or three repricing scenarios per cycle, picks the best, and calls it strategy. That is not strategy. That is blind sampling from a surface that has never been mapped. The frontier—the boundary of achievable growth and profitability—exists in every P&C insurance book and every bank credit strategy. No carrier and no bank has ever computed it. The constraint was never analytical capacity. It was architecture. A cache makes it visible. The repricing space is finite and discrete. Identical factor vectors always produce identical outcomes. Compute once, retrieve forever, compute on cache miss. This single architectural move collapses multi-hour runs to minutes. Hundreds of strategies become evaluable where three were the ceiling. The frontier emerges for the first time. A five-agent pipeline navigates it: the Explorer reprices the book via the cache, the Shaper projects portfolio composition through demand models at the individual policy level, the Evaluator scores profitability and growth, the Guardian screens for regulatory constraint, and the Orchestrator returns a ranked frontier to the decision-maker. No new models. No regulatory approval. No governance committee. Weeks to first value. Months to full scale. The advantage compounds permanently.

1. Introduction

Better telematics. Better alternative data. Better credit scoring. None of it changed the number that matters: how many strategies get evaluated before one is deployed. The answer is still two or three. It has been two or three for thirty years. The models got smarter. The mechanism stayed the same.

A repricing cycle begins when loss trends, competitive dynamics, or portfolio analytics signal that current rates or credit strategies are misaligned with risk. An actuary or credit strategist builds a response. Two or three scenarios are modeled. One is selected. It is deployed.

This paper argues that the two-to-three scenario ceiling is not a product of analytical capacity. It is a product of compute architecture. And when the architecture is corrected, something important becomes visible for the first time: the efficient frontier.

The architecture introduced here produces mathematically identical output to the existing approved engine, stored rather than recomputed. It is not an operations research model. It invokes no regulatory prohibition on algorithmic pricing. A governance committee reviewing this system has nothing to review. Not by oversight. Structurally.

We make a specific claim about return on investment. This is the highest-ROI agentic AI use case in financial services, on three grounds. First, governance friction is structurally zero: the system produces no model output, so there is nothing to validate, approve, or explain that is not already validated, approved, and explainable. Second, time to first value is measured in weeks, not quarters. Third, the frontier gap revealed by a narrow proof of concept funds the full program.

2. The Aleatory Contract and the Cost of Mispricing

Insurance is an aleatory contract. The policyholder pays a certain premium. The insurer accepts an uncertain obligation. Once written, there is no renegotiation. Akerlof's foundational analysis of information asymmetry in markets [Akerlof, 1970, pp. 489–492] shows precisely this mechanism at work in insurance: when buyers cannot assess quality before purchase, adverse selection concentrates risk in the remaining book—Akerlof demonstrates that quality uncertainty drives out high-quality sellers, leaving the pool progressively riskier. A policy priced below its actuarially indicated rate is a locked consequence; the full cost of that mispricing accrues regardless of any subsequent correction.

Banking carries the same irreversibility. Once a loan is approved and drawn, the credit risk is on the book at the price set at origination. Rate changes can address new originations. They cannot reprice the existing book. Tirole's treatment of oligopolistic pricing dynamics [Tirole, 1988, ch. 5, pp. 208–218] establishes the theoretical basis: in concentrated markets where a small number of firms set prices simultaneously, any single carrier's mispricing that persists across repricing cycles creates durable competitive disadvantage because rivals observe market share shifts and adjust their own pricing in subsequent periods—a ratchet that cannot be unwound by correcting the price at the next cycle.

This asymmetry is the foundational pressure on repricing quality: the consequence of getting the price wrong is locked in at the moment of commitment, but the ability to correct it is deferred to the next deployment cycle. The gap between commitment and correction is the financial cost of a slow repricing engine.

3. The Hidden Epistemological Problem

3.1. Blind Sampling

The actuary evaluates three scenarios, selects the best, and proceeds. This is called rigorous. It contains one assumption that is never examined: that the best of three is close to the best possible.

It may not be. The space of possible repricing configurations across territories, perils, tiers, coverage structures, deductible levels, and competitive response as-

sumptions is vast. Three points sampled from that space tell an institution almost nothing about where the efficient frontier lies. The selected scenario could be optimal. It could be far from it. There is currently no way to know, because the frontier has never been computed.

3.2. The Invisible Frontier

Figure 1 makes the problem precise. In the left panel, three scenarios are evaluated and one is selected. The efficient frontier is invisible. In the right panel, the same institution operates with a repricing agent. Hundreds of configurations are evaluated. The frontier emerges as a concave arc bowing toward the upper-right—the classical efficiency boundary introduced by Markowitz [Markowitz, 1952, pp. 82–88], who defined an efficient portfolio as one for which no other portfolio offers higher expected return at the same variance, transposed here from asset allocation to repricing strategy: an efficient repricing strategy is one for which no other strategy offers higher profitability at the same growth target, or higher growth at the same profitability target.

A note on geometry. The strategy space here is finite and discrete, not a continuous surface. With M strategies evaluated, the frontier is strictly the upper-left convex hull boundary of M points in $(\Delta\text{DWP}, \Delta\text{CR})$ space—piecewise-linear by construction, not smooth. The concavity property follows not from the variance-covariance structure of continuous assets (which drives Markowitz) but from the economic monotonicity of the growth-profitability trade-off at the policy level: raising rates on preferred segments simultaneously improves margin and increases lapse, while rate decreases do the reverse. These objectives pull in opposite directions at every policy, so no strategy can dominate in both dimensions simultaneously. With hundreds of strategies populating the convex hull, the piecewise-linear boundary approximates a smooth concave curve. Figure 1 illustrates this approximation.

The gap between current position and frontier is now a measurable quantity. It has a dollar value and a direction. That gap, once visible, is what makes the case for the full program. It was always there. Nobody was looking for it.

4. The Intuition: Repricing as Search

The rest of this paper is engineering. Here is the core insight that makes it work.

4.1. The Geometric Insight

The repricing space is finite and discrete. In P&C insurance, filed factor tables define a finite grid: territory takes one of a few hundred values, vehicle symbol one of a few thousand, driver age falls into defined bands, tier is categorical. In banking, credit strategy grids are equally discrete: score bands, LTV buckets, DTI tiers, product categories.

This discreteness has a critical consequence: identical factor vectors always produce identical outcomes. A policy or account with the same rating factors will receive the same repriced result, every time, without exception. This is not an approximation. It is definitional.

In a personal lines auto book of two million policies, the number of distinct pricing profiles is orders of magnitude smaller than the policy count for a typical filed factor structure: a plan with 300 territories, 4 tiers, 20 age bands, and 8 vehicle symbols produces approximately 192,000 distinct profiles against 2,000,000 policies. The repricing engine need only execute on those distinct profiles. All other records resolve by lookup.

Fast search infrastructure makes this lookup essentially instantaneous. Johnson, Douze, and Jégou demonstrated that GPU-accelerated approximate nearest-neighbor indices achieve single-digit-millisecond query latency at billion-vector scale with recall exceeding 95% [Johnson et al., 2021, pp. 537–541]—specifically, their FAISS library achieves 10^9-vector search in under 10 ms on a single GPU by exploiting SIMD parallelism and product quantization. In the repricing context, the relevant search space is not the raw policy count but the much smaller set of distinct factor-vector profiles. Against a key space of 10^5–10^6 distinct profiles, a GPU-resident exact hash index returns results in microseconds, not milliseconds. The repricing engine that consumed approximately 15 hours for a full personal lines auto book—the typical runtime for a scenario evaluation on a two-million-policy book requiring full actuarial computation across all records—now runs in minutes, because the compute fires only on cache misses against a much smaller key space.

This is simultaneously a high-performance computing use case. The Explorer's cache layer runs on GPU-native infrastructure. The Shaper and Evaluator operate at the scale of millions of policy-level computations per scenario. The platform is agentic AI at the interface layer and HPC at the execution layer, each enabling the other.

4.2. Relation to Prior Work

The idea of storing computed results to avoid recomputation has two classical foundations. Bellman's principle of optimality [Bellman, 1957, ch. III, pp. 83–84] establishes that optimal substructure solutions can be stored and reused without recomputation—the foundational insight behind dynamic programming. Michie coined the specific term *memo function* and demonstrated its application to machine learning [Michie, 1968, p. 19]: a function that caches its outputs, indexed by input, so that repeated calls with the same argument return the cached value instantly. The contribution here is not the cache primitive itself but its application to a domain where it has never been applied: a regulatory-filed discrete pric-

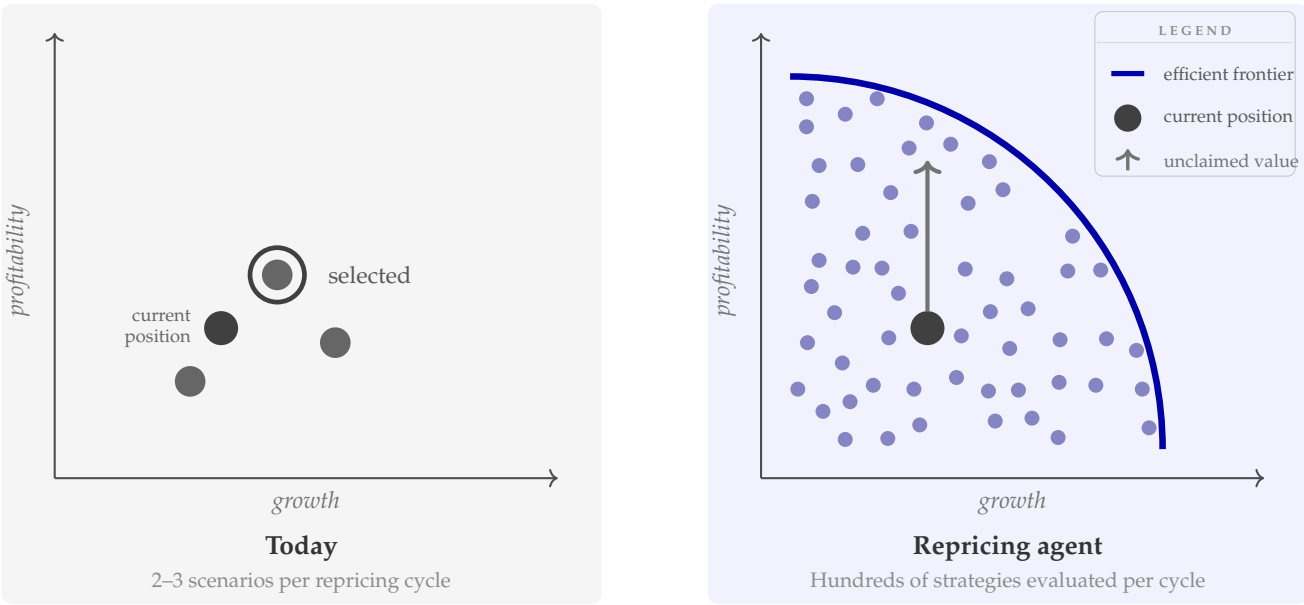

Figure 1: The efficient frontier in financial services repricing. *Left*: two to three scenarios evaluated per cycle; the frontier is invisible and the selected scenario may be far from the best achievable position. *Right*: with an agentic repricing system, hundreds of strategies are evaluated, the frontier emerges as a concave arc bowing toward the upper-right, and the gap between current position and the best achievable result is measurable and closeable. Same book. Same repricing engine. The only difference is whether the frontier can be seen.

ing space where cache keys are factor vectors and values are outputs of an already-validated production engine.

The critical distinction from recent financial ML is architectural: prior approaches approximate a continuous pricing function using a neural network [Huge and Savine, 2020, pp. 1–4]. The repricing cache stores exact outputs of a discrete approved engine. No model is trained. No approximation is introduced. The governance implication follows directly: there is no model to validate because there is no model.[1]

Existing commercial repricing platforms—Earnix, Radar Live, and the Willis Towers Watson pricing suite—expand scenario throughput through parallelization, but each evaluation invokes the rating engine from scratch. None apply cache memoization to regulatory-filed factor vectors as an architectural primitive.[2] The governance-free property follows directly from that architecture: no new computation means no new model to validate. It cannot be retrofitted onto a recompute-based design without rebuilding the engine.

5. The Single-Agent Architecture

The architecture has three operations and one rule.

5.1. The Cache Primitive

- **Lookup**—given a factor vector, check the cache for a previously computed outcome.
- **Compute**—if no cache entry exists, run the full repricing computation against the existing engine. Exact. Unchanged.
- **Store**—write the result to the cache, keyed by the factor vector.
- **The rule**—if search is empty, compute.

No approximation is introduced. No centroid is substituted. No fuzzy matching is performed. The cached

[1] Huge and Savine's differential ML trains networks to "simultaneously learn a function and its derivatives," using "pathwise differentials as training labels" [Huge and Savine, 2020, p. 2]. This produces fast sensitivity analysis across a continuous surface—a different problem from ours, solved with a different tool. Their approach introduces a trained neural network as an intermediary in the computation and therefore triggers model risk governance requirements: SR 11-7/OCC 2011-12 in banking; NAIC Model Audit Rule #205, ORSA model governance requirements, and applicable state DOI model oversight frameworks in insurance. Our architecture introduces no such intermediary and triggers none of these requirements.

[2] This characterization is based on publicly available product architecture documentation, technical whitepapers, and configuration guides for each platform as of 2024. Earnix's published architecture describes scenario evaluation as a rating-engine invocation per policy per scenario (Earnix Platform Architecture Overview, 2023); Radar Live similarly describes per-scenario full rerate pipelines (Verisk Radar Live Technical Guide, 2023); WTW's Radar and Arius documentation describes batch scenario processing without a memoization layer (WTW Radar Product Documentation, 2024). None of the vendor documentation describes a factor-vector cache keyed to filed regulatory outputs. An operator with direct platform access should verify current product capabilities, as vendor architectures evolve.

result is the result the existing repricing engine would produce, because it was produced by that engine and stored verbatim.

5.2. Self-Improving Behavior

The cache is self-improving within each scenario evaluation. Every distinct factor-vector profile that triggers a compute enriches the cache for that run. In early evaluations, the cache is sparse and compute fires frequently. As the evaluation proceeds, it densifies. Lookups increasingly dominate. The within-scenario speedup grows as the run progresses.

Cross-scenario reuse depends on the nature of the rate plan change. When a scenario changes factor table multipliers without restructuring the underlying factor categories, the factor vectors of existing policies are unchanged and the cache carries over. When a scenario restructures the factor index itself, affected policies produce new vectors and must be recomputed. The within-scenario speedup applies universally. Cross-scenario reuse applies where factor structure is held constant.

5.3. Human Interaction: Generate, Analyze, Select

The agent operates in two modes. In autonomous mode, it continuously explores the repricing surface without a human prompt. In human-triggered mode, any decision-maker issues an objective in plain language rather than a specific strategy configuration.

Stopping rules are human-defined. Examples include: achieve the highest growth above a minimum profitability threshold; run N strategies, whichever comes first; or stop when marginal improvement across successive strategies falls below a threshold.

The human experience is a three-step loop. *Generate*: the platform produces scenarios autonomously, each a point on the frontier. *Analyze*: the decision-maker reviews the frontier visualization as it accumulates. *Select*: the decision-maker chooses a frontier point; the Orchestrator returns the corresponding repricing strategy ready for deployment.

This is what makes the platform genuinely agentic rather than merely automated. The human does not configure parameters. The human states a business objective and navigates a frontier. The agent does everything in between.

5.4. Why Governance Cannot Touch It

The governance argument is structural, not procedural. The agent produces mathematically identical output to the existing repricing engine because it *is* the existing repricing engine, with a cache layer in front of it.

There is nothing to validate that is not already validated. Nothing to deploy that has not already been approved. Nothing to explain that is not already explainable. The factor tables, business rules, and regulatory filings that define the pricing logic are unchanged. The cache does not modify them; it stores their outputs verbatim.

A governance committee reviewing this system has nothing to review. Not by oversight. Structurally.

6. The Multi-Agent Game-Theoretic Frontier

The single agent reveals the frontier. The pipeline makes it deployable.

6.1. The Blind Spot of the Single Agent

The single agent does one thing precisely: it maps the institution's own repricing surface faster than any prior architecture. That precision is also its limit.

The single agent reveals the repricing surface of the institution's own book. But it operates in isolation: it cannot model what the market will do in response, whether the projected portfolio will actually materialize under competitive pressure, or whether the resulting position violates institutional constraints.

These are not minor gaps. They are the difference between a frontier point that looks good on paper and one that holds in practice.

The multi-agent pipeline closes all three gaps. Nash's foundational theorem on non-cooperative games establishes that in any finite game, at least one equilibrium exists in which each player's strategy is a best response to the strategies of all other players [Nash, 1951, p. 286]. As Dixit and Nalebuff establish in the applied context, in strategic interaction your choice of action depends on what you expect your rivals to do [Dixit and Nalebuff, 1991, ch. 3, pp. 63–87]. The pipeline operationalizes this reasoning at the individual policy level, not as a macro market assumption, but as a policy-by-policy competitive score derived from each competitor's filed factor tables.

6.2. The Five-Agent Pipeline

Figure 2 illustrates the architecture as a sequential pipeline. Each agent is deterministic and auditable. No new model risk is introduced anywhere in the stack.

Explorer takes a repricing strategy and reprices the book via the cache. Output: the repriced book.

Shaper takes the repriced book and applies demand models at the individual policy level. For each policy, it estimates the probability of conversion on new business quotes and the probability of retention on renewals at the new price, incorporating the competitive price for that specific risk from each major competitor. The output is a policy-by-policy projected portfolio composition under this strategy, not a macro market assumption.

Evaluator takes the projected portfolio and scores it on both axes of the frontier. Profitability is computed by running the projected portfolio through the loss and expense models. Growth is the total premium volume of

the projected portfolio, accumulated across converting new business and retaining renewals. Output: a single (growth, profitability) coordinate placing this strategy as a point in frontier space.

Guardian screens the Evaluator's output against regulatory and institutional constraints before it reaches the Orchestrator. In insurance, these include NAIC ORSA concentration limits and state regulatory floors as codified in the NAIC Model Audit Rule and associated state filings [NAIC, 2010]. In banking, they include Basel III minimum capital adequacy ratios and the countercyclical capital buffer requirements [Basel, 2017, pp. 2–5, 128–130] and CECL forward-looking loss provisioning requirements under FASB ASU 2016-13 [FASB, 2016, pp. 1–4]. The Guardian enforces constraints that already exist and are already validated; it introduces no new ones. Output: admissible coordinate, or flagged with the violated constraint.

Orchestrator receives the screened coordinate, applies the stopping rule, and either directs the Explorer to the next strategy or terminates. On termination, it returns the full ranked frontier to the decision-maker: every admissible strategy scored on profitability, growth, and the durability of each position under the competitive response already embedded in the Shaper's demand models.

6.3. What Changes

The single agent answers: *where is the frontier?* The pipeline answers: *here is the frontier, constructed from demand, loss, and expense models at the policy level. Which point do you want to stand on?*

6.4. A Worked Example

A carrier's Orchestrator fires the Explorer with a Texas personal lines auto strategy: a tier-rebalancing that raises rates 4% on preferred risks (approximately 55% of the Texas book) and holds standard rates flat. The Explorer reprices all 340,000 Texas policies via cache in 11 minutes (warm-cache conditions; the cache was pre-populated by prior scenario runs within the same repricing cycle, covering the majority of distinct factor profiles in the Texas book).

The Shaper runs each repriced preferred-risk policy through the demand model. The new rate is compared against competitor quotes for that specific risk profile. Where the carrier's new rate exceeds competitor rates by more than the estimated switching friction for that risk segment, the model projects lapse. Where it remains within retention tolerance, it projects renewal.

Output: of the 187,000 preferred-risk policies, 174,000 are projected to retain (7% lapse rate on the affected segment); 153,000 standard-risk policies retain at the base rate. The model projects approximately 7,000 new business policies entering the book across both segments during the projection window, based on the current pipeline conversion rate applied to the competitive index output. Net projected portfolio: approximately 334,000 policies (174,000 preferred retained + 153,000 standard retained + 7,000 new business).

The Evaluator runs the projected portfolio through the loss and expense models. Projected combined ratio: 94.6%. Projected DWP growth: -1.8%. Additional KFIs: projected retention rate 94.7%, new business conversion rate 6% below current on preferred segments, implied return on surplus improvement of 2.4 percentage points. This strategy trades modest volume for meaningful margin improvement.

The Guardian screens: no concentration limit violations, no regulatory floor breaches. Admissible.

The Orchestrator logs the coordinate (DWP growth $= -1.8\%$, combined ratio $= 94.6\%$) and fires the next strategy. After 120 strategies, it terminates and returns the ranked frontier. The decision-maker sees, for the first time, the full achievable surface and where each strategy sits on it.

7. Generalization Across Financial Services

7.1. P&C Insurance

The architecture applies wherever factor tables define the repricing space: personal lines auto, small commercial, workers compensation, commercial auto. This covers approximately 70–75% of US P&C premium volume.

For large commercial accounts, the unit of composition shifts. Rather than repricing individual policies, the agent assembles and evaluates coverage tower configurations and scores account-level profitability across combinations. The cache stores rated tower components; the agent assembles what-if scenarios by combining them.

7.2. Banking

In credit decisioning, the repricing space is defined by score bands, LTV buckets, DTI tiers, product categories, and channel. A credit strategy is a mapping from this space to approval decisions, credit limits, and pricing tiers. The cache stores outcomes for evaluated strategy configurations; the agent explores the strategy surface in exactly the same way.

Applications include unsecured credit, mortgage origination, deposit pricing, and limit management. The Shaper's demand models incorporate publicly disclosed competitor rate sheets and market surveys at the individual account level. The Guardian applies Basel III/IV constraints, CECL provisioning requirements, and concentration risk limits.

7.3. The Universal Discriminator

The architecture applies wherever: (i) the repricing space is defined by a finite set of factor categories with discrete

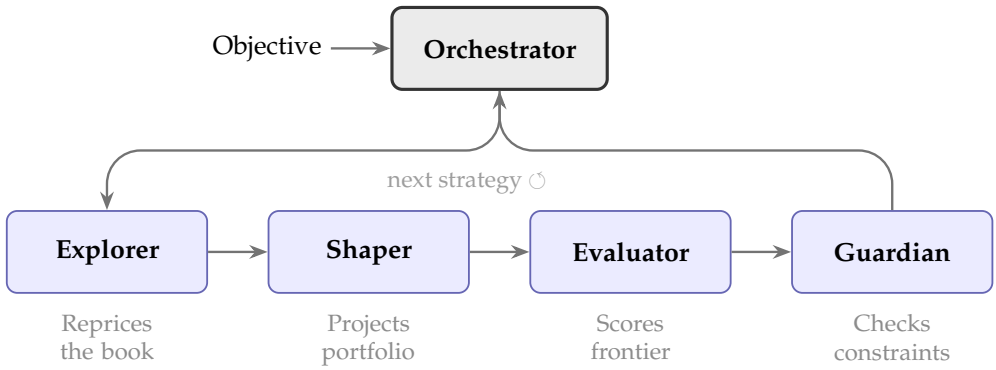

Figure 2: Five-agent repricing pipeline. The Orchestrator fires the Explorer with a strategy; the pipeline flows left to right; the Guardian returns the scored, screened result to the Orchestrator, which either loops back to the Explorer with the next strategy or terminates and returns the ranked frontier to the decision-maker.

values; (ii) within a fixed factor structure, identical factor vectors always produce identical outcomes; and (iii) the repricing computation is expensive enough relative to lookup that caching produces material speedup. These conditions hold across the vast majority of P&C insurance and banking repricing contexts.

8. The Compounding Strategic Effect

One repricing cycle with a frontier navigator is better than one without. That is not the interesting result.

The interesting result is what happens in year three.

The institution that operates near the efficient frontier in cycle one prices its book correctly. Correct pricing runs adverse selection in reverse: the risks that stay are the risks worth keeping. Better risk profiles produce more stable loss experience. Stability builds deployment confidence. Confidence enables more frequent repricing. More frequent repricing closes the competitive response window before rivals can correct. Each cycle is faster, better-informed, and more consequential than the last.

The competitor running three scenarios per cycle is not just behind. They are falling further behind with every cycle, and they cannot see it, because they have never seen the frontier either.

An institution that moves first does not merely price better. It sets the competitive tempo for every market it operates in. Rate adequacy becomes a structural property of the book rather than a periodic correction. That is a different kind of company.

9. The Phased Reality

The commercial path is deliberately narrow at the start. A proof of concept on a single product in a single market demonstrates the frontier concept, begins populating the cache, and quantifies the gap between current position and the best available strategy on a representative slice of the book. That gap funds the full program. It has never failed to.

Weeks to first value. A proof of concept on a single product in a single market begins populating the cache and quantifies the frontier gap on a representative slice of the book.

Months to production scale. A proper engineering program delivers the full architecture: cache layer, search index, the five-agent pipeline, frontier visualization, and audit infrastructure. The repricing engine is untouched. The policy administration or origination system is untouched. The integration surface is minimal.

Permanent compounding advantage. Each repricing cycle produces intelligence about which frontier positions hold under competitive response and which erode. That intelligence accumulates. The competitive advantage compounds indefinitely across deployment cycles.

10. Conclusions

The efficient frontier in financial services repricing has always existed. No carrier and no bank has ever seen it. The constraint was never analytical capacity. It was compute architecture: specifically, the compulsion to recompute from scratch what could instead be retrieved from a cache.

The repricing space is finite. Factor categories are discrete. Within a fixed factor structure, identical inputs produce identical outputs. These properties together make the repricing problem a search problem in disguise, and fast search infrastructure, matured to billion-vector scale in recent years [Johnson et al., 2021], finally makes the disguise removable.

The five-agent pipeline described here extends the single-agent cache into a complete scenario evaluation instrument: the Explorer reprices, the Shaper projects the portfolio at the individual policy level, the Evaluator

scores profitability and growth, the Guardian screens for constraint violations, and the Orchestrator drives the loop. The result is the first architecture in financial services that allows any decision-maker to state a business objective rather than a rate plan and receive a ranked frontier of deployable strategies in return.

It requires no new models. No regulatory approval. No governance committee.

Agentic fraud detection is a frequently cited high-ROI application of AI in financial services and deserves direct comparison. It faces two compounding problems. First, undetected fraud is systematically mislabeled as legitimate in training data; models can only learn what prior detection rules already knew and cannot outperform their own blind spots. Second, false positives consume investigator capacity that is never idle: the queue always refills, and net ROI after investigation costs is real but narrow.

The repricing pipeline faces analogous problems in materially milder form. Strategy prices are calculated exactly from the approved rating engine: no estimation, no approximation. Their effect on the portfolio flows through demand, loss, and expense models that already exist and are already validated. Pricing data is far less censored than fraud labels. A false positive is a scenario that does not reach the frontier; it costs compute, not human time. The comparison holds. The case is structural.

The efficient frontier has always existed. The compute to find it has existed for years. The governance argument for deploying it is airtight. The only question left is which institution decides that seeing the frontier, finally, is worth the ninety days it takes to build the cache.

That institution does not merely price better than its competitors. It sets the tempo. Everyone else responds.

11. Limitations

The most binding constraint on adoption may be organizational rather than technical. The platform shifts the analyst's role from building strategies to stating objectives. In mature actuarial and credit risk cultures, where expertise is defined by the ability to construct and defend specific rate plans, this transition requires active sponsorship at the CRO or Chief Actuary level. Without it, the platform risks being adopted as a scenario accelerator—a faster version of the existing three-scenario process—rather than as a frontier navigator. The efficiency gain is real in either case. The compounding strategic advantage requires the organizational transition.

The cache architecture delivers its primary speedup within each scenario evaluation: distinct factor-vector profiles are computed once and reused across all policies sharing that profile. Cross-scenario reuse applies when rate plan changes alter factor table multipliers without restructuring the underlying factor index, without reclassifying territories, redefining tier bands, or changing variable coding. When a scenario restructures the factor index, affected policies produce new vectors and the cache rebuilds for those profiles. The within-scenario speedup applies universally. Cross-scenario reuse applies selectively.

The accuracy of the frontier score depends on the quality of the underlying demand, loss, and expense models. Those models are inputs to the pipeline, not outputs of it. Improving them improves the frontier. The pipeline architecture does not constrain their sophistication. Loss models that adequately capture the heavy-tailed behavior of insurance and credit losses—whose mathematical foundations are developed in Embrechts, Klüppelberg, and Mikosch [Embrechts et al., 1997, chs. 1–3]—will produce frontier estimates robust to tail scenarios; simplified Gaussian loss approximations will not. The $\hat{L}_i$ term in the Evaluator's combined ratio computation inherits whatever precision the carrier's actuarial models already provide. The frontier score is only as credible as the loss model beneath it.

The Shaper's policy-level competitive scoring depends on competitor rate transparency. In markets with limited individual-level rate data, the competitive score will be less precise. The framework assumes competitor rates can be estimated from filed factor tables and observed market data; in thinly filed markets this assumption weakens.

The Guardian agent enforces regulatory constraints as static rules. Regulatory environments evolve. Keeping the Guardian's constraint library current requires ongoing maintenance—a compliance function, not an engineering one, but a real one.

Acknowledgements

This paper was drafted in collaboration with Claude (Anthropic), a large language model that served as primary thinking partner, co-writer, and editorial critic throughout. All analytical content, claims, numerical examples, and conclusions are the author's own. The author accepts full responsibility for any errors.

The views expressed are those of the author in a personal research capacity and do not represent the positions of Capgemini SE or any client organization.

References

Akerlof, G. A. (1970). The market for 'lemons': quality uncertainty and the market mechanism. *The Quarterly Journal of Economics*, 84(3):488–500.

Basel Committee on Banking Supervision (2017). *Basel III: Finalising Post-Crisis Reforms*. Bank for International Settlements, December 2017.

Bellman, R. (1957). *Dynamic Programming*. Princeton University Press.

Dixit, A. K. and Nalebuff, B. J. (1991). *Thinking Strategically: The Competitive Edge in Business, Politics, and Everyday Life*. W. W. Norton & Company.

Embrechts, P., Klüppelberg, C., and Mikosch, T. (1997). *Modelling Extremal Events for Insurance and Finance*. Springer.

Financial Accounting Standards Board (2016). *Accounting Standards Update 2016-13: Financial Instruments—Credit Losses (Topic 326): Measurement of Credit Losses on Financial Instruments*. FASB, June 2016.

Huge, B. and Savine, A. (2020). Differential machine learning. *Risk Magazine*. arXiv:2005.02347.

Johnson, J., Douze, M., and Jégou, H. (2021). Billion-scale similarity search with GPUs. *IEEE Transactions on Big Data*, 7(3):535–547.

Markowitz, H. (1952). Portfolio selection. *The Journal of Finance*, 7(1):77–91.

Michie, D. (1968). Memo functions and machine learning. *Nature*, 218:19–22.

National Association of Insurance Commissioners (2010). *Model Audit Rule: Annual Financial Reporting Model Regulation (#205)*. NAIC, adopted December 2010.

Nash, J. (1951). Non-cooperative games. *Annals of Mathematics*, 54(2):286–295.

Tirole, J. (1988). *The Theory of Industrial Organization*. MIT Press.

Appendix: The Shaper Demand Model

The Shaper operates as a three-step sequential process. Step 1 (Explorer) produces scenario prices. Step 2 (Shaper) runs those prices through demand models to determine which risks stay in or enter the portfolio. Step 3 (Evaluator) scores the resulting portfolio. The distinction matters for the governance argument: each component is a pre-existing validated model. The pipeline assembles them in sequence. Nothing new is introduced.

Step 1: Scenario Prices

Strategy s defines a new rate plan. The Explorer applies it to every risk i in the book via the cache, producing updated price p_i^s. For all risks not affected by the rate plan change, the cache returns the existing price verbatim.

Step 2: Demand Filter

Notation for Step 2. The following variables are used throughout:

- x_i — full rating vector for risk i (territory, symbol, age band, tier, and all filed factors)
- p_i^s — carrier's annualised premium for risk i under strategy s
- $\{p_{ik}^c\}_{k \in K}$ — filed-rate competitor premiums for risk i, one per competitor k
- k^* — index of cheapest competitor for risk i
- CI_i^s — competitive index: ratio of carrier rate to cheapest competitor rate
- $\mu(x_i)$ — segment-level switching friction (fraction above cheapest competitor at which 50% of policyholders lapse), estimated from observed lapse indicators
- $\tau(x_i)$ — friction dispersion (slope parameter of the logistic retention curve)
- $\alpha(x_i)$ — baseline new-business conversion rate for segment x_i
- $\beta(x_i)$ — price elasticity of new-business conversion for segment x_i
- $\sigma(\cdot)$ — logistic function: $\sigma(z) = 1/(1+e^{-z})$

For each risk i, the batch rater (e.g. Quadrant, Minuquote) accepts x_i and returns exact competitor premiums $\{p_{ik}^c\}_{k \in K}$, rated through each competitor's filed factor tables. The competitive index for risk i under strategy s is:

$$\mathrm{CI}_i^s = \frac{p_i^s}{p_{ik^*}^c}, \qquad k^* = \arg\min_{k \in K} p_{ik}^c \tag{1}$$

No distributional assumption is required. The batch rater returns exact filed-rate competitor premiums per policy.

Retention. An existing policyholder stays if the carrier's new price is within switching friction of the cheapest competitor. The retention probability is:

$$\Pr(\mathrm{stay}_i \mid s) = \sigma\left(\frac{-(\mathrm{CI}_i^s - 1 - \mu(x_i))}{\tau(x_i)}\right) \tag{2}$$

where σ is the logistic function, $\mu(x_i) > 0$ is the segment-level switching friction, and $\tau(x_i)$ is friction dispersion. Both are estimated by regressing observed lapse indicators on batch-rated CI_i at renewal date, stratified by risk segment. When $\mathrm{CI}_i^s = 1$ the carrier is at market; above 1 it is above the cheapest competitor by that ratio.

New business. A quoted risk enters the portfolio if the carrier is competitive enough for that segment:

$$\Pr(\mathrm{enter}_i \mid s) = \sigma\big(\alpha(x_i) - \beta(x_i) \cdot \mathrm{CI}_i^s\big) \tag{3}$$

where $\beta(x_i)$ is the segment-level price elasticity, estimated by regressing observed bind indicators on batch-rated CI_i at quote date. The intercept $\alpha(x_i)$ is the baseline conversion rate for segment x_i; multi-product relationships enter here—a risk already holding a home policy carries a different baseline conversion rate than a monoline prospect. The logistic link is used for tractability and interpretable odds ratios; the architecture is link-function agnostic.

The projected portfolio under strategy s is:

$$P^s = \{i \in R : \mathrm{stay}_i\} \cup \{j \in N : \mathrm{enter}_j\} \tag{4}$$

where R is the renewal book and N is the new business pipeline.

Step 3: Scenario Scoring

The Evaluator runs P^s through the existing loss and expense models to produce the scenario combined ratio and DWP, reported as change from base:

$$\text{DWP}^s = \textstyle\sum_{i \in P^s} p_i^s \cdot e_i, \qquad \text{CR}^s = \frac{\sum_{i \in P^s} (\hat{L}_i + \hat{E}_i)}{\text{DWP}^s} \tag{5}$$

$$\Delta\text{DWP}^s = \text{DWP}^s - \text{DWP}^0, \quad \Delta\text{CR}^s = \text{CR}^s - \text{CR}^0 \tag{6}$$

where p_i^s is the annualised written premium for risk i under strategy s, $e_i \in (0, 1]$ is the earned exposure fraction (policy in force for a full term has $e_i = 1$), $\hat{L}_i$ and $\hat{E}_i$ are the predicted loss and expense for risk i from the carrier's existing actuarial models, and DWP^0, CR^0 are the base-case values under the current deployed strategy.

The coordinate $(\Delta\text{DWP}^s, \Delta\text{CR}^s)$ is the point on the efficient frontier for strategy s. The Orchestrator accumulates these coordinates across all evaluated strategies until the stopping rule is met, then returns the ranked frontier to the decision-maker.

Numerical Trace: One Policy Through the Pipeline

The following trace uses a single representative preferred-risk policy from the Texas scenario in §6.4 to verify that the pipeline equations produce outputs consistent with the aggregate results reported there.

Policy characteristics. Preferred tier, Texas, personal auto. Current annualised premium $p_i^0 = \$1{,}200$. Full-term policy ($e_i = 1.0$). Predicted loss $\hat{L}_i = \$810$; predicted expense $\hat{E}_i = \$210$ (combined ratio at base: $(810 + 210)/1200 = 85.0\%$, consistent with a profitable preferred risk).

Step 1: Explorer. Strategy s raises preferred rates by 4%. Cache hit; returns $p_i^s = \$1{,}200 \times 1.04 = \$1{,}248$.

Step 2: Demand filter. Batch rater returns cheapest competitor premium for this risk: $p_{ik*}^c = \$1{,}180$.

$$\text{CI}_i^s = \frac{1{,}248}{1{,}180} = 1.058 \tag{7}$$

The carrier is 5.8% above the cheapest competitor. Segment-level switching friction for preferred Texas auto: $\mu(x_i) = 0.08$, dispersion $\tau(x_i) = 0.04$.

$$\begin{aligned} \Pr(\text{stay}_i \mid s) &= \sigma\left(\frac{-(1.058 - 1 - 0.08)}{0.04}\right) = \sigma\left(\frac{0.022}{0.04}\right) \\ &= \sigma(0.55) \approx 0.634 \end{aligned} \tag{8}$$

The carrier is below the friction threshold ($\text{CI}_i^s = 1.058 < 1 + \mu = 1.08$), so the retention probability exceeds 0.5.

Step 3: Scoring contribution. This policy is projected to be retained ($\text{stay}_i = 1$). Its contribution to the scenario score:

$$\text{DWP contribution} = p_i^s \cdot e_i = \$1{,}248 \tag{9}$$

$$\text{CR contribution} = \frac{\hat{L}_i + \hat{E}_i}{p_i^s} = \frac{810 + 210}{1{,}248} \approx 0.817 \tag{10}$$

At the aggregate level, preferred risks like this policy contribute below-average loss ratios to the portfolio—consistent with the scenario-level CR of 94.6% reported in §6.4, which reflects a mix of preferred (low CR) and standard (higher CR) risks under the strategy's selective premium adjustment.

All model inputs—loss model, expense model, retention elasticities, new business elasticities, and batch-rated competitor premiums—pre-exist in the carrier's validated infrastructure. The pipeline introduces no new epistemic liability.