

AI in stewardship

A strategic framework for asset managers

Introduction

This paper is written across three chapters for the stewardship leader who has to navigate the AI shift, and the strategy that goes with it.

Contents

Chapter one

The strategic argument. How AI changes the stewardship function, and how that fits within the wider AI strategy of your firm.

1. Why now? The forces reshaping stewardship	3
1.1 AI Promises to solve the conflict of interest inherent in proxy advice	5
1.2 The capability shift in AI models	7
1.3 The core idea Stewardship infrastructure, a central LLM, and the outputs it generates	9
1.4 The opportunity	17

Chapter two

The architectural argument. It sets out the rules engine, the cited data layer, the vote records, and the submission pipeline that the stewardship function actually runs on. We set out what good looks like with use cases, why it matters, and how it connects to the strategy in Chapter one.

2. The new architecture of stewardship infrastructure	19
2.1 Stewardship data and the trust stack	21
2.2 The rules engine	25

Chapter three

The operating model that brings together the strategic and architectural arguments, setting out how stewardship teams should organise themselves, make build-or-buy decisions, work with vendors, and deploy AI in practice.

3. Implementing this within your firm	31
3.2 A roadmap for stewardship leaders	33

Closing

The AI transformation	36
-----------------------------	----

Why now? The forces reshaping stewardship

Every manager now has to decide how to embed AI in stewardship: what to build, what to buy, where to let it run, and where to keep a person in the loop.

The decisions are forced by an environment that has become harder to manage with the tools and team sizes most stewardship functions still operate with.

The pressure is partly internal — more disclosure, more client reporting, more demand on a fixed analyst headcount. But the sharper change is in the proxy-advisory market itself. Regulatory and political pressure on the established advisors has intensified over the past two years. An executive order, an open SEC investigation, and the reclassification of proxy-advisor reliance as an operational risk on the asset manager's own register have together turned a procurement decision that used to be routine into a strategic one.

The supplier relationship most functions have built around for two decades is being questioned and in some cases displaced. The pressure is coming from regulators and clients alike.

AI promises to solve the conflict of interest inherent in proxy advice

For two decades, the proxy advisory industry has operated as a duopoly.

A small number of firms produce the voting recommendations that more than 90% of asset managers rely on, and the custom voting policies those managers implement, and critically, the consulting services sold to the very issuers whose proposals are being voted on.

The firms advising funds on how to vote are also earning money from the same companies being voted on.

In fact, it's their profit centre. Annual changes to voting policies forced issuers to keep buying the formulas.

- 

Regulators have stopped treating this conflict of interest as a structural quirk, and started treating it as an enforcement target.
- 

In-house counsel at large asset managers are now classifying proxy advisor reliance as a material risk.
- 

The exposure that used to sit on the proxy advisors now sits, at least partially, on the asset managers who depend on them.

We've heard from the market that the proxy advisors' historic defence has been an economic one

Producing high-quality custom voting policies at scale — applying a bespoke policy across thousands of meetings, with the issuer-by-issuer judgement that genuine stewardship requires — was expensive.

That expense was subsidised by the consulting revenue from the issuer side whilst also relying mostly on benchmark policies. Custom work for the long tail of issuers was never very profitable on its own; the issuer-side consulting revenue is what made it viable.

Take that revenue away, the argument runs, and the custom service managers actually want would not exist.

AI changes this

The cost of delivering a custom policy collapses by an order of magnitude. The economic argument that justified the conflict of interest disappears.

This is relevant for stewardship leaders in two ways:

- 1 It reframes the AI conversation: this is not only an efficiency story, it is the way out of an increasingly untenable dependency on conflicted vendors.
- 2 The asset managers who build their internal capability now — clean voting infrastructure, addressable data, constrained retrieval over filings, evaluation infrastructure — will be operating a stewardship function that is both cheaper, more effective, and more defensible than one held together by proxy advisor subscriptions.

The capability shift in AI models

AI has developed significantly since the launch of ChatGPT in 2023. The models have reached a stage where meaningful work can now be accomplished by them.

Three capability shifts within AI matter for stewardship strategy:

1 Raw intelligence

Models have become dramatically smarter over a small space of time and are predicted to continue doing so. An “AI strategy” should hold this concept front and centre. Models are only going to improve and what can’t be done today will likely be possible tomorrow.

2 Agentic execution

Models are no longer limited to producing a single answer to a single prompt. They can be given a goal, the tools to pursue it, and the discretion to call those tools as needed — reading data, calling APIs, drafting outputs, checking their own work. Tasks that took multiple human steps and multiple handoffs can now be completed end-to-end by an agent operating on the right infrastructure. This is the capability that turns AI from assistant into operator, and it is the one the architecture in this paper is built around.

3 Retrieval and verification engineering

This is the shift that matters most, and the one most easily misread. The accuracy that makes AI usable for stewardship does not come from a frontier model on its own. ChatGPT or Claude out of the box will not reliably answer a multi-document stewardship question, and treating “point an LLM at it” as the strategy is how firms get it wrong. It comes from the system built around the model: constrained retrieval, multi-stage retrieval, automated evaluations and AI-driven source verification, working together. That is the difference between a raw model and an engineered AI stack. With the stack in place, accuracy on the kind of multi-document analytical questions stewardship analysts ask rises from the low-to-mid 80s into the 99-plus range, and the output is auditable rather than a black box. Without it, you are relying on the model’s best guess.

Most asset managers are in the middle of their journey. Most are still using language models to accelerate basic tasks, such as drafting emails, summarising documents, doing a piece of research. The leverage available from a properly built stewardship-AI stack that is connected to the right data, with the right boundaries and the right human-in-the-loop discipline is far beyond that.

The rest of this paper is about closing that gap.

The core idea: Stewardship infrastructure, a central LLM, and the outputs it generates

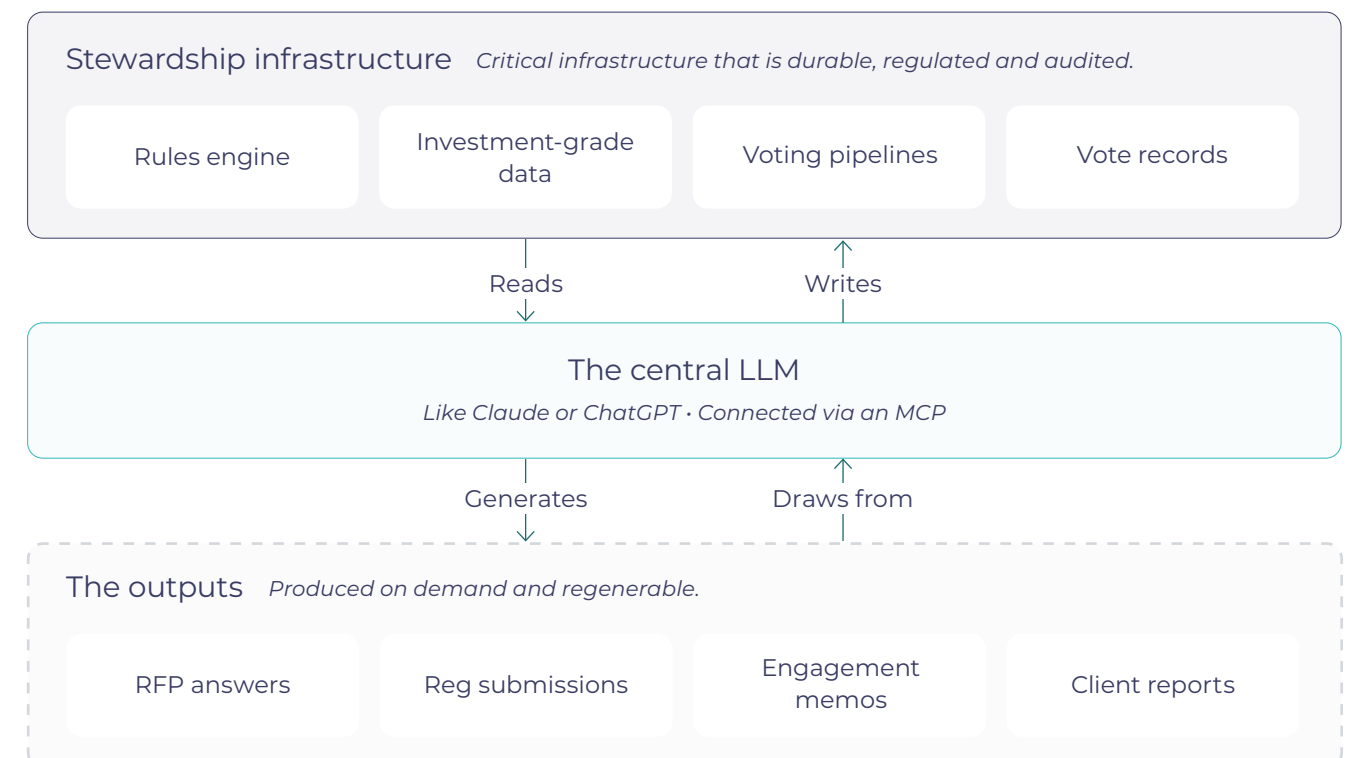
As explained in the previous section, frontier AI models keep improving and show no sign of plateauing. In order to benefit from this continued improvement, asset managers can deploy a central LLM, such as Claude or ChatGPT.

However, in order to be effective at work, the LLM is only as good as the context it can reach, so it needs access — through a Model Context Protocol (MCP) — to the systems holding data and carrying out decisions. This set up, a central LLM with access to data/actions through MCP, forms the basis for an AI strategy within the asset manager.

For stewardship, this means connecting Claude/ChatGPT/Copilot (hereafter referred to as “the central LLM”) to the critical stewardship infrastructure that enables you to decide, execute and report on votes.

In this next section, we go into detail around:

- Stewardship infrastructure: the durable, regulated systems the function runs on: Rules engine, investment-grade data, voting pipelines, vote records. You build this once and keep it. This is the layer Tumelo provides.
- The LLM orchestration layer: the central LLM itself, connected through MCP to that infrastructure, deciding what to retrieve, reasoning over it, and doing the work.
- Outputs: the RFP responses, regulatory submissions, engagement notes and vote rationales the LLM generates on the fly from that infrastructure. Nothing here is built and stored; it's generated when asked for, and regenerated when the data moves.

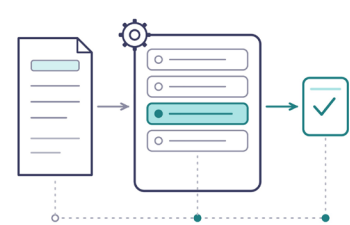


The stewardship infrastructure

what you build once and keep

There are four components that make up the infrastructure. They are built once, robustly, and then owned and maintained, not regenerated:

1 The voting policy rules engine



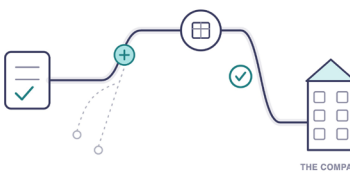
It must be flexible enough to mirror your policy and robust enough to reproduce the same answer every time. Every decision must be reconstructable: which policy applied, which rule fired, which data point was used, and which source that data came from. This is non-negotiable for Stewardship Code disclosure, SRD II reporting, and the rising bar of client scrutiny. An audit trail bolted on afterwards is not acceptable; it has to be a property of the system.

2 Investment-grade data



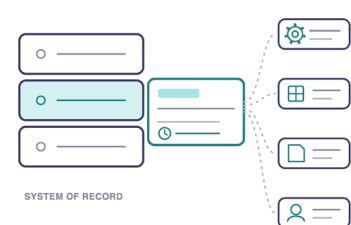
The information you need to decide how to vote. It must be accurate, with every figure traceable to the source; timely, refreshed at a cadence that matches the work and its deadlines; and carry enough metadata, including stable identifiers, to be joined across systems and reconstructed years later. Data that fails any of these tests becomes a liability for every workflow that draws on it.

3 The voting pipelines



The route a vote travels to the company. Global vote submission requires integration with custodians whose systems vary by market, jurisdiction and client, and once connected they must be monitored and maintained. None of this is work an LLM should be doing ad hoc. It is infrastructure you build, test rigorously, and trust to keep working.

4 The vote records database



The system of record for every ballot cast and the rationale behind it. Each entry is date-stamped and linked to the rule that fired, the data that fed it, the source filing, and the person who submitted or edited it. This is what makes the rest of the infrastructure defensible; regulatory and client scrutiny both depend on it.

The central LLM, and MCP

We believe the key strategy most firms will deploy will be a central LLM connected via the Model Context Protocol (MCP) to all required data and systems.

One interface over many systems

For years, the goal inside asset management has been a unified data and workflow platform, a single place where everything is structured, joined and accessible. A capable LLM with the right connections now largely is that interface. It decides which system to query for which question, retrieves the data, joins it across sources, and returns the answer with citations. The integration problem that used to need a data warehouse can increasingly be solved at the application layer instead.

The implication is that the firm's job is no longer to build a single unified platform. Its job is to make each underlying system *accessible* to the LLM — reachable through a clean interface, governed by sensible permissions, and producing output the LLM can reason over. Unification happens at the LLM, not at the data layer.

Model Context Protocol (MCP) as the boundary

Unique challenges present themselves when connecting LLMs to your data.

Unrestricted API access is too permissive: it is the equivalent of handing the LLM a developer's login, and most legal teams will rightly refuse it. Screen-driving, which is letting the LLM click through a browser or desktop app like a person, is too fragile: it impresses in a demo and then breaks the first time a layout changes or an authentication prompt appears, and it inherits everything a logged-in user can do.

MCP is the standard that has emerged for connecting LLMs to internal systems.

The LLM is given a defined list of permitted actions: `read filings` `read voting records` `write engagement notes` and can do those and nothing else. Every action is logged and every permission can be revoked. The firm can show, after the fact, exactly what the LLM was allowed to do and exactly what it did. The shift is from *the LLM can do anything a developer could* to *the LLM can do these named things, and we have a record of each*. That is a position regulated firms can live with.

Data, and the contracts that govern it

The data the LLM reaches comes in three categories, each with different contractual obligations.

1 Internal proprietary data

Meeting notes, internal research, voting history, engagement records.

The contractual question here belongs first in any AI-strategy discussion: the LLM provider must not ingest this into a training set or cache it beyond the session.

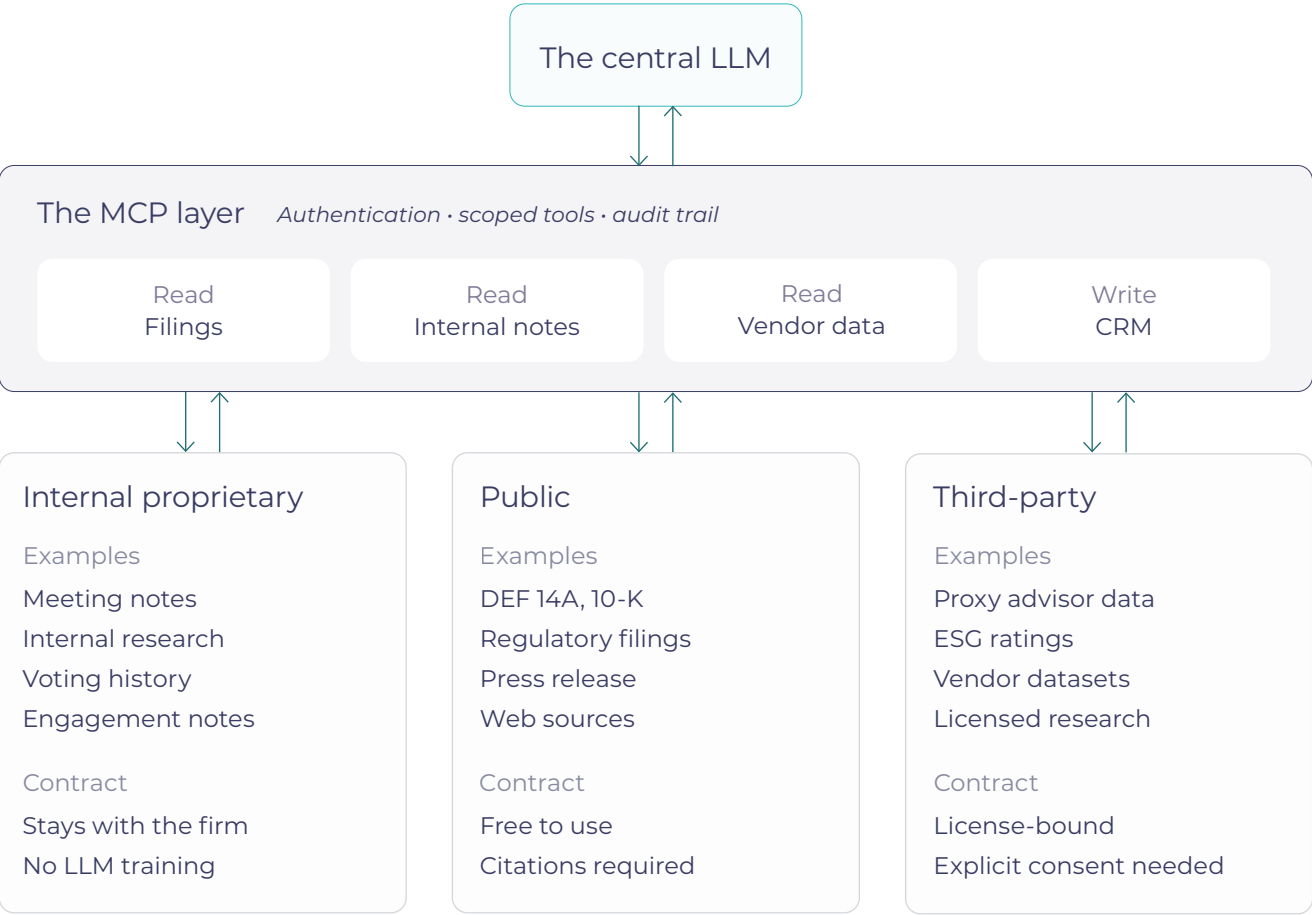
Most frontier providers will agree to these terms in enterprise contracts; the point is to make the terms explicit and reviewed.

2 Public data

Filings, regulatory submissions, press releases, web sources. Free to retrieve and use, with one requirement: every output drawn from it carries a citation, both for audit and because an uncited answer is unverifiable.

3 Third-party data

Proxy-advisor data, ESG ratings, vendor research, licensed datasets. The most contested category, and the one most managers underestimate. Licence terms typically restrict downstream use, and many never anticipated being made available to a customer's LLM. The work is contractual and explicit: vendors must agree, in writing, that their data may reach the firm's LLM. Some will agree, some will agree with conditions, and some will refuse.



Working with compliance

There exists a natural tension between the IT/compliance team and the stewardship team. In order for the LLM to be useful, it needs access to the right systems. But this also creates risk. The answer is to work on more granular access and go to the teams with a specific list of criteria in order to accomplish the workflows you have in mind.

Specifically, enabling the LLM access to read research, write/edit rules in the voting policy, run tests of the voting policy and read the voting records is a must in order to enable most workflows.

The outputs | What the LLM generates

A stewardship team is responsible for producing a large amount of outputs: an RFP answer, a Stewardship Code submission, a quarterly client report, a rationale for a contested vote. These have to be accurate but they are not infrastructure. Each is produced once, for one audience, and superseded by the next version.

The historic way to handle this was to do each by hand, or to buy a dedicated tool for each — an RFP tool, a reporting product — each one with its own subscription, integration and data feed.

With the stewardship infrastructure built and a central LLM connected to it, you no longer need any of that. Once the infrastructure is solid and the LLM can access it and take action on your behalf, producing outputs is a request, not a project: the analyst asks '*answer these RFP questions*', and the LLM accesses the relevant parts of the infrastructure, assembles the document, and cites every figure back to its source. Next quarter, when the numbers have moved, they regenerate it rather than rebuild it.

Stewardship teams can set up skills for larger, more complex tasks that can trigger manually or automatically. Building engagement agendas or preparing a stewardship report are two tasks that would sit as skills ready to be triggered by the team.

Huge unlocks occur because the extremely capable LLM now has access to all the data and actions it needs. Outputs cost almost nothing to produce, because the expensive part — trustworthy, cited, current data — was already built once, in the stewardship infrastructure.

What this looks like in practice

Two concrete examples show the output layer at work for things that stewardship teams do today the hard way.

Answering stewardship questions for clients and reporting

A wide range of recurring work — regulatory and Stewardship Code reporting, ad hoc client requests, and RFP questionnaires — comes down to the same task: how many ballots did we vote, on what share did we vote against management, how does that differ for this client's holdings versus the firm-wide book, how many engagements did we hold and with whom.

Historically, an analyst would export voting records into Excel, rebuild pivot tables, and would answer slightly differently each time because the work is redone from scratch. The new way: the analyst points the firm's LLM at the stewardship infrastructure — rules engine, voting records, engagement data — and asks it to answer; the LLM pulls the data, applies the right cuts (firm-wide vs client, full-year vs year-to-date), and returns each answer with citations back to source. The analyst reviews and edits where judgement is needed.

The gain is consistency as much as speed: answers trace back to the records and don't drift between analysts or need re-defending under scrutiny.

Refining a voting policy with a backtest

Teams iterate on policy every off-season — adjusting thresholds, tightening guidance, responding to client preferences. The question that should sit at the heart of every change — *what would happen if we treated any director with more than four other public-company boards as overboarded, instead of five?* — has historically been very hard to answer with evidence; a real backtest meant a request to the proxy advisor, weeks of waiting, and a bill.

With the rules engine and voting records connected to the LLM, the analyst describes the change in plain language and the system generates two or three variants, runs each against last season, and returns a short comparison against the firm's actual votes, broken down by sector, geography and issuer size. A backtest that used to take months and an external supplier now takes an afternoon and an internal decision.

The opportunity

The pay-off from getting this distinction right is substantial.

The time cost of producing outputs like an RFP response or Stewardship Code submission falls sharply. More importantly, the cost of asking a what-if question such as “*What would tightening director independence to four boards do to our voting outcomes?*” approaches zero.

Our framework makes the strategic decisions for asset managers straightforward. Build or buy decisions all flow from one test: *is this part of the stewardship infrastructure, or is it an output?* Stewardship infrastructure is where you buy the best provider you can find and integrate it deeply. Outputs sit on top of the infrastructure — generated on demand by the LLM with the right access. Answer that question for each part of the function and the operating model largely writes itself.

Get the distinction wrong and organisations tend to fail in one of two ways:

1 Paralysis

Treating every output as if it were a regulated system, with lengthy governance processes and little delivered.

2 Fragility

Asking the LLM to own capabilities it should never control, such as the voting database, vote-execution logic, or audit trail. The distinction between infrastructure and outputs is what avoids both.

The new architecture of stewardship infrastructure

Chapter one made the strategic case, this chapter describes what the stewardship infrastructure actually is, so a buyer can tell a serious system from an “AI-powered” label.

A proxy advisor sells research and recommendations to a human reader. Someone else’s analyst reads the filings, forms a view, and delivers it as a PDF.

Stewardship infrastructure is built differently — a rules engine, a cited data layer, a vote records database, and a submission pipeline, all reachable by the firm's own LLM.

The manager no longer needs a vendor's analyst reading filings on its behalf; its own analysts are freed to do the parts of stewardship that need human judgement.

Stewardship infrastructure has four components. Vote pipelines and the vote records database were covered in Chapter one and follow the same architectural pattern as the two described next: the data layer and trust stack, and the rules engine.

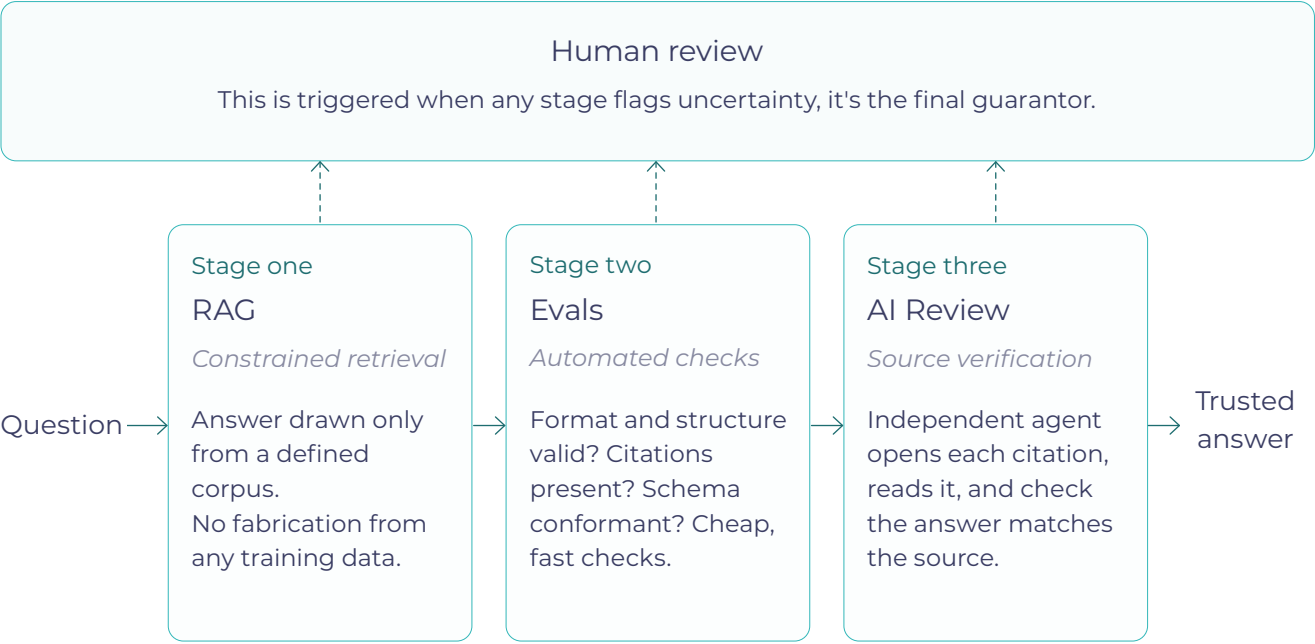
Stewardship data and the trust stack

Producing investment-grade stewardship data has always been labour-intensive: teams of analysts reading filings, extracting numbers, building comparisons, writing summaries, every season, across dozens of markets. The headcount that took is what the proxy advisors recovered through issuer-side consulting. A system built on AI-extracted data can build the same dataset with a fraction of the headcount, more accurately, and at roughly an order of magnitude lower cost.

The catch is trust: AI-collected data is only useful if managers can rely on it, and reliance comes from the infrastructure, where each data point is traceable to source, each output is checkable against the filing, each error is caught and escalated to a person. Four layers provide it.

The diagram below shows the four layers in sequence: constrained retrieval, automated evaluations, independent AI review, and human escalation. Each layer is more expensive than the one before it. Each is independently auditable.

Trust by construction | The validation pipeline



1 Stage one | Constrained retrieval (RAG)

A modern stewardship data system builds its own evidence base. Rather than asking the language model to answer from its training data, the system ingests the relevant filings — DEF 14As, 10-Ks, 8-Ks and their equivalents in other jurisdictions — breaks them down into structured form (text, tables, charts and images all converted into something the model can reason over), and stores them in a database that retrieves by meaning rather than by keyword match.

When a user asks a question, the system pulls the relevant material out of that database and hands only that material to the language model, with instructions to answer from what it has been given and to flag where any evidence is absent. The pattern is called retrieval-augmented generation, or RAG. It changes what the model is doing in a fundamental way: it is no longer producing answers from training data the user cannot see, it is interpreting documents the user can read for themselves.

Every output is cited back to the underlying filing. If the source material does not contain the answer, the system says so rather than guessing. Both behaviours are what make RAG-based research defensible.

Multi-stage retrieval

Constrained retrieval works well for precise questions where the answer sits in a specific place inside a single document — what engineers sometimes call needle-in-a-haystack questions. It works less well for the kind of questions stewardship analysts actually ask: multi-year remuneration comparisons, director overboarding across overlapping mandates, peer-relative analysis on metrics like CEO pay against total shareholder return. These require the system to pull information from several documents at once and synthesise across them, and a standard semantic query usually returns too much irrelevant context for the model to find the relevant signal.

The fix is to break retrieval into stages. Each question is classified first, a tailored retrieval plan is built for it, and the results are ranked and filtered before any of it reaches the language model, so that only the most relevant evidence ends up in the prompt.

This is the area of AI engineering that has matured most rapidly in the past 18 months, and it is where the gap between the best systems and the weaker ones is widest. A system without sophisticated retrieval engineering will produce accuracy figures in the low-to-mid 80s on multi-document analytical questions; a system with it can produce accuracy figures above 99%. For a buyer of stewardship data, the implication is that “AI-powered” is now too broad a label to be useful, two providers using the same marketing language can be 15 accuracy points apart.

2 Stage two | Automated evaluations

A RAG system has to be tested continuously. The second layer of the trust stack is evaluation infrastructure — known as "evals" by engineers — a set of automated tests that run on every output the system produces. They check that the format is right, the citations are there, the data conforms to the expected structure, and the answer is internally consistent. The tests are cheap to run, catch the obvious failures, and give the engineering team the metrics they need to spot regressions before they reach production. Evals are not a project that is done once and signed off. They are continuous infrastructure: expanded as new failure modes appear, version-controlled with the system itself, and reviewed at the same cadence as any other production system.

3 Stage three | AI review

The third, heaviest, and priciest layer of trust is independent AI review. A second agent — separate from the one that produced the original answer — opens each citation, reads it, and checks whether the answer is genuinely supported by the source. This is the closest thing to having a human reviewer in the loop. Most answers do not need to reach it; they pass cleanly through retrieval and the evaluation layer. But for any output that will drive a decision with real consequences, independent review is what takes the system from “probably right” to “definitely right” and verifiable.



Human-in-the-loop

A person sits above all three stages rather than at the end of them: any stage that flags uncertainty escalates to a stewardship analyst, who reads the citation and decides.

The economics work because the automated stages do most of the catching, allowing the expensive resource and analyst time to be reserved for the small fraction of outputs that genuinely need a person, and their corrections feed back into the evaluation set so the system improves.

The rules engine

A voting policy is a set of rules.

Vote against director X under condition Y

Withhold support from compensation plan Z if metric M exceeds threshold T

Apply jurisdiction-specific defaults where local norms differ from the firm's house view

This is what allows a stewardship team to vote across thousands of meetings a year without each meeting being adjudicated from scratch. The policy is the institutional knowledge; the rules engine is the mechanism that applies it at scale.

A rules engine architecture that AI can leverage

A robust rules engine has to model the full range of policy detail a stewardship team uses in practice. This includes, but is not limited to:

- Custom director independence rules with carve-outs for tenure, committee composition, and board diversity.
- Custom pay for performance models
- Watchlist functionality that can run different rules depending on different company criteria
- All the different edge cases that emerge each proxy season

The design principle that makes this work, and makes it compatible with regulated infrastructure, is the separation of objective from subjective. Custom policies are expressed as objective rules. When a meeting comes up, the system does not ask the LLM *how should we vote on this pay proposal*; it applies the rule. The rule is deterministic, testable and version-controlled, so two analysts running the same policy across the same meetings get the same result.

Three things follow from this design:

1 Model risk

The decision logic is deterministic, so what auditors and regulators care about — what the system decides, and why — is fully inspectable.

2 Defensibility

A contested vote can be traced end to end: the rule that fired, the data that triggered it, the filing the data came from.

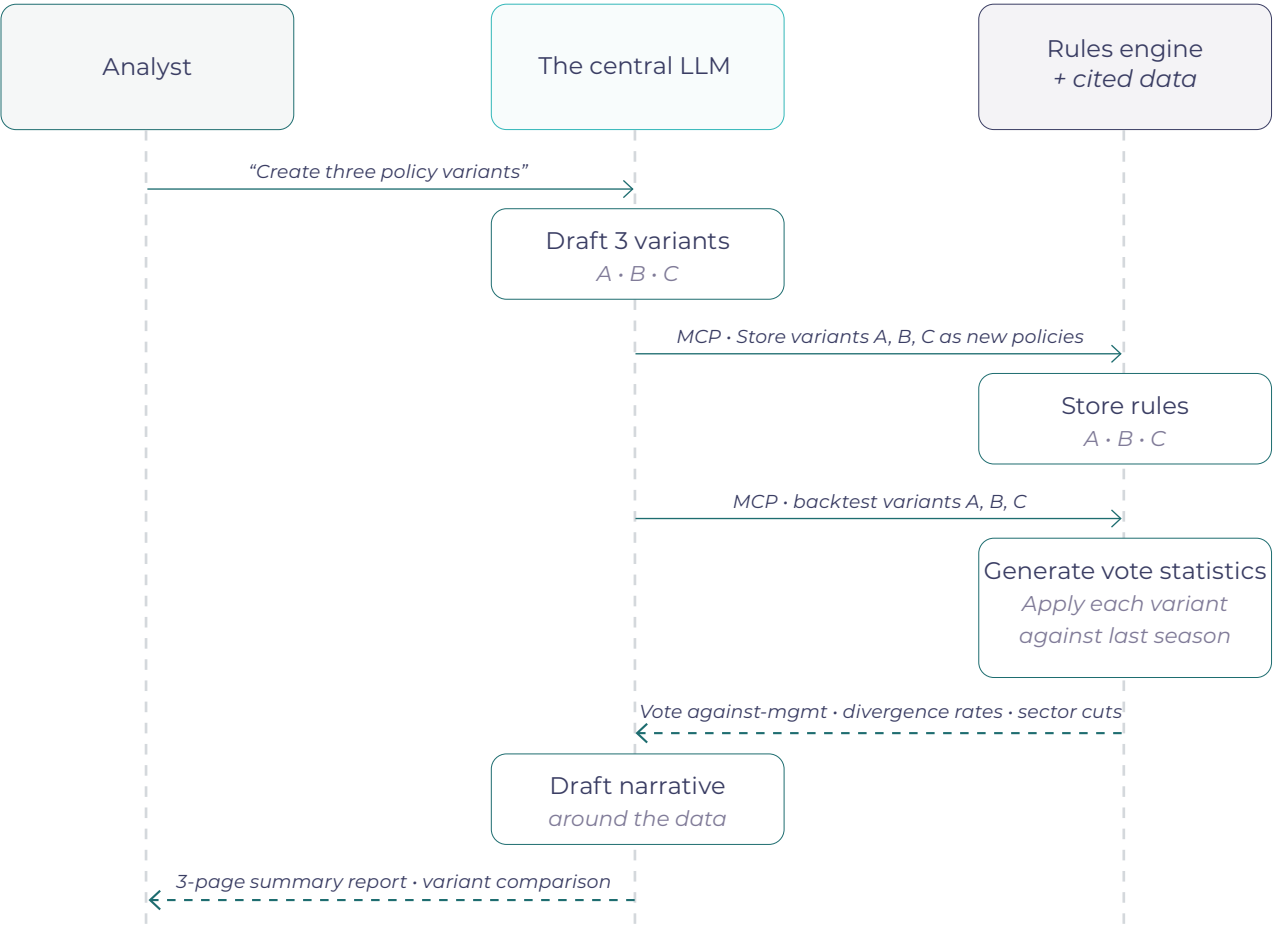
3 Change management

When a policy needs to evolve, the change is to the rule, not to a prompt. It can be tested, reviewed, and versioned in the way any other piece of regulated infrastructure is.

The rules engine does the deciding. When a meeting comes up, the engine retrieves the data the rules require and applies them. The decision logic itself is deterministic, testable, and version-controlled.

Backtesting | The capability the industry has lacked

This is also what makes empirical policy refinement possible — the backtesting shown in Chapter one. With data already structured and cited and the engine able to apply any policy specification against the existing dataset, a research project that used to take months becomes an afternoon's work. It is the single biggest capability the industry has lacked.



Other workflows that follow the same pattern

Once the LLM-plus-clean-infrastructure pattern is in place, a long tail of stewardship workflows unlocks. Each is the same shape; the LLM reads, retrieves, and composes; the critical infrastructure provides the source-of-truth data; humans focus on judgement rather than labour.

1 Stewardship Code and SRD II Reporting

Annual disclosure documents largely assembled from the same voting and engagement records, with the LLM drafting narrative around figures it has retrieved with citations. MCP connection across multiple systems enables it to pull the data together for your submission.

2 Client customisation

With a voting policy built as clear, objective rules it becomes easy to personalise this for clients in segregated mandates or pass-through voting programs.

3 Contradiction detection

Flagging issuers whose board statements or proxy materials conflict with previous engagement commitments or peer-disclosure norms.

The economic argument

The technical architecture also explains why the proxy advisors' cross-subsidy described in Chapter one stops being necessary.

The defence the incumbents have historically offered is economic: producing genuinely custom voting policies at scale is expensive, the fees paid by asset managers do not cover that cost on their own, and the issuer-side consulting revenue is what makes the asset-manager-side product viable. Without the cross-subsidy, the argument runs, the custom service that asset managers actually want would not exist.

A provider running a rules engine over AI-extracted data has a different cost structure. The marginal cost of supporting an additional custom policy is not a team of analysts; it is a configuration of the engine. The fixed cost of building the engine is spread across the entire client base. The economics of custom policy provision change accordingly.

Custom work becomes affordable for a wider range of clients, including the asset managers that have historically used benchmark research because the bespoke price was hard to justify at their AUM. The market for custom policy work grows at the same time the cost of serving each policy comes down.



Ask yourselves how important each one of these things is to the way you want to run your stewardship function:

Can the architecture of the rules engine actually codify my voting policy?

A codified policy runs the same way every time, can be backtested, and produces a complete audit trail.

Can I backtest a candidate policy change against last year's data, in less time than it takes me to write the proposal for that change?

If a backtest requires a project, a scope document and a bill at the end, the engine is not built for the kind of policy iteration this paper describes.

Can I modify my policy myself, or does every change have to be implemented by the proxy advisor?

A rules engine that requires vendor intervention for every change is exactly how you end up spending months trying to refine your policy and is not something AI can leverage.

Implementing this within your firm

The architecture described in this paper does not run itself. It needs an operating model and a small number of new roles, with a clear framework for build or buy, and an honest re-think of what the existing stewardship analyst's day looks like.

Operating model and people

Workflow and prompt ownership

Someone in the team has to own the library of prompts, agents, and workflows everyone uses. This is the equivalent of the senior analyst who, traditionally, knew how to pull the right Excel template for each RFP. The new version of that role curates the patterns the central LLM uses across the function and ensures they are accessible across the team. Work with your stewardship infrastructure provider in order to get out of the box skills and prompts that best leverage the system.

Better stewardship

The work that becomes possible is the work that analyst capacity has historically been gated against.

- Building voting policies from the ground up that represent your firm's views. You can now support multiple policies, be that for specific fund types or for individual clients.
- More engagement, with more issuers, on more contested topics.
- Deeper preparation for contested AGMs. Genuine peer analysis rather than generic comparisons.
- Stewardship reporting that reflects what the team has actually done, with more nuance drawing on the full data set with ease rather than having to manually stitch it together yourself.

These are things that teams have wanted to do for years, not blocked by the lack of ambition, but by lack of analyst hours.

Build or buy | The decision framework

Build or buy decisions all flow from the distinction between stewardship infrastructure and what the LLM can create independently.



LLMs enable firms to achieve more in-house than ever before — shifting the build vs buy equation. Historically, firms would have bought dedicated software such as RFP tools, reporting products, etc. Each with its own vendor relationship, and integration cost. LLMs now allow firms to accomplish these tasks in-house, leveraging the stewardship infrastructure described earlier (rules engine, data, vote pipeline, vote records).



A good, AI-enabled stewardship infrastructure provider will help you refine the workflows to generate the outputs — writing skills for Claude or ChatGPT (your central LLM) to use and building automated workflows.

The test of any vendor in your stewardship stack is whether they have built their product to be consumable by your central LLM. Vendors that fit into this architecture become more valuable as models improve.

A roadmap for stewardship leaders

The work to operationalise this architecture inside a stewardship team breaks naturally into two years. The first year is about securing the stewardship infrastructure. The second is about scaling on top of it. The temptation in both years is to invert the order — which is the most common path to expensive failure.

Year one | Secure the stewardship infrastructure

The first year is about putting the right foundations in place.



Secure the stewardship infrastructure

Replace your current arrangement with a provider that satisfies the rules-engine architectural tests in Chapter two — AI-native rules engine, MCP exposure, self-serve policy creation, rigorous evals. The procurement window described in Chapter one means this is the moment when buy-side leverage is highest; firms that move now will set their stewardship stack for the next decade.



Expose the firm's stewardship data via MCP

Voting records, engagement logs, internal research notes, and the rules engine itself all need to be accessible through scoped MCP tools. This is the single most leveraged piece of infrastructure work the firm will do in the AI era. Without it, every workflow described in Chapter one remains a manual process. External providers should help with this, but internal data will require some technical work.



Pilot two or three “generated-output” workflows

RFP response is the natural first pilot, it's a high volume, painful incumbent process, that could bring immediate analyst leverage. Stewardship Code reporting is the second. Pre-AGM briefing packs is a strong third. The point of the pilot is not to deliver every workflow; it is to prove the pattern works inside the firm so year-two scaling is not a leap of faith.

Year two | Scale

Year two is when you see the benefits of the work done in year one.



Scale agentic workflows across the analyst function

The patterns proven in year-one pilots get applied across the long tail of analyst work.



Expand the MCP catalogue

The first MCP integrations were the firm's own systems plus the new stewardship infrastructure. Year two extends this to engagement CRMs, regulatory filings databases, client portals, ESG data providers, and any internal system the LLM should be reasoning over. Each new MCP integration multiplies the number of workflows it enables.

What to avoid

Three failure modes are common enough to warn about.

1 Letting an LLM build the durable infrastructure

The temptation is to ask the central LLM to maintain a voting database, produce its own data/research, or implement the decision logic — because in a demo, it can. It always fails in production.

2 Blanket bans on AI

The opposite failure. Compliance or legal teams react to model-risk concerns by withholding access to the data the LLM needs, on the assumption that less access is safer. Granular access is safer than less access — less access produces tools that cannot do the work, and the roadmap dies under blanket bans.

3 Treating AI strategy as a technical project rather than an operating-model project

The most common and most expensive failure. A central technology team is given a budget, deploys an LLM, deploys some MCP infrastructure, and discovers that the stewardship team is not using it because the operating model has not changed. AI strategy is operating-model strategy. The technology is necessary but never sufficient.

Closing

The AI transformation

AI is transforming how an asset manager operates. Strategies to take advantage of this involve deploying a central LLM connected across your firm's data and providers, enabling efficiencies and removing the drudgery. Ultimately, elevating teams to deliver better outcomes for clients.

With the shift in regulation happening at the same time, stewardship teams now need to think about how to position themselves for success over the next five years.

This paper lays out the practical pieces of that answer and the key components to achieve this success.

- 1 Stewardship infrastructure (the rules engine, data layer, voting pipelines and voting records).
- 2 The central LLM, and how this connects via MCP enabling a central interface that can analyse and complete tasks for you.
- 3 The outputs and workflows that follow, including voting policy refinements in afternoons not months, and RFP answers in minutes not days.

The asset managers who draw the line in the right place will run stewardship functions that are cheaper, faster, more bespoke and deliver better outcomes for clients. The architecture is already being built, by providers who can see what the next decade of stewardship will look like.



Written by Will Goodwin

Tumelo Co-Founder

will@tumelo.com

[LinkedIn](#)

