

MEMO

To: Fluint Team

From: Nate & Jon

Re: Olli's "Judgement:" How we win the market & defend our position

Part 1: The Market's Telling Us Something Important

Everything our customers' top reps do, but can't explain, is what we need to build around.

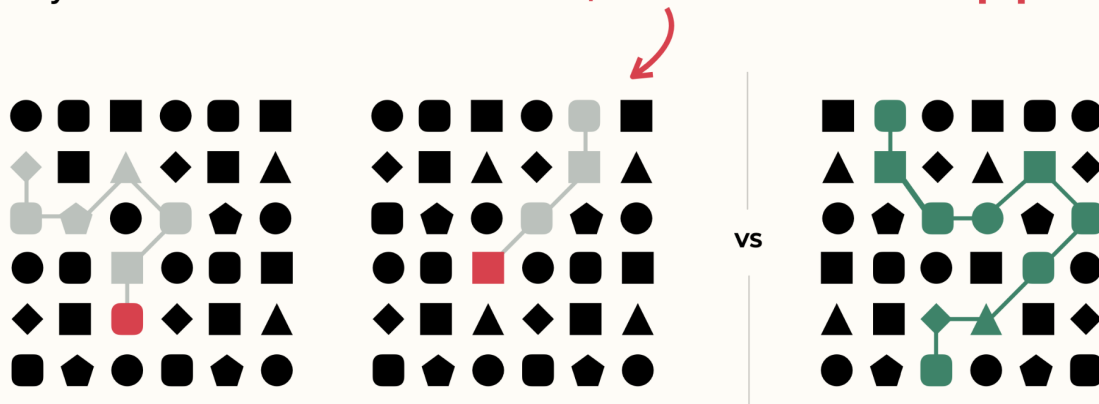
The stuff that will never live inside a sales process, playbook, CRM, knowledge base, etc. (e.g. what an AE infers from a silent Slack thread, how they know which contact in IT to engage upfront, even when they typically kill deals), it's the "tribal" knowledge. And it's the single, most valuable asset and competitive edge living inside a GTM org.

Here's why it matters for us:

In almost every enterprise team, a small number of reps (17% reps) drive the majority of the team's revenue (80%). The gap between those reps and the middle of the pack is almost always that type of "invisible" knowledge: accumulated experience, pattern recognition, and "judgement" that's built up from hundreds of cycles. The explicit stuff (decks, templates, emails, process) is basically the same across reps.

Which means the single, highest-leverage thing any enterprise GTM org can do is, (1) capture that tribal knowledge, (2) wrap every new rep, every deal, and every forecast review with it. It's the only way to scale what the few do differently, to the other 80%.

Enterprise deals are full of "gray" and nuanced decisions that take years to master and creates **\$ millions in stalled pipeline:**



One of our customers, Matt, calls it putting "lightning in a bottle."

And it's what we're building Olli to do.

For now, we're calling this new layer **GTM Judgment**. We think it's not only what our customers need, but the most defensible approach to build. The rest of this memo is about why, and the problems in our own offering and GTM it's addressing.

The Opacity Problem

Evan Armstrong from *The Leverage* has a good way of saying this: When you hire McKinsey, you can't reverse-engineer what they did into a bill of materials. The production process is "opaque," because you're paying for judgment and institutional knowledge that's bundled into a deliverable whose value is difficult to quantify.

- AI work is increasingly transparent, however. Sophisticated buyers can see the API calls, estimate token counts, and calculate the rough cost of what's delivered. If the work they're buying is a markup on some prompt engineering using their context, vs. a unique add from the vendor, that erodes trust. "Why pay for this?"
- This makes the production process "reproducible." When customers can call Claude with a system prompt, skills and retrieved context to feel they're getting "enough" through an internal build, value evaporates.

The Quality Problem

When I look at the most expensive resource in our company, defined by price per hour, it's our lawyer. We're willing to pay \$1,000/hour for quality counsel, because we're buying access to judgment, for nuanced and sticky scenarios.

- I "rent" his set of experiences with venture-backed companies, and his ability to deliver a highly-valuable outcome (board structure, negotiations, valuation, etc.) because we trust it, and can't access it elsewhere (like Claude). Plus, getting it wrong is expensive.
- So the work we're billed for isn't about a "document" that could be reproduced by my own prompting in Claude. Instead, we're billed for a series of documents that fit together to form a unique strategy, sequenced in the right order, for our specific situation, with the foresight of what's likely to happen in our Series C, D, etc... to keep us ahead.

And again, all based on judgement that's accumulated over time, been tested, and is highly-valuable in a complex, nuanced topic.

The same holds true for our customers:

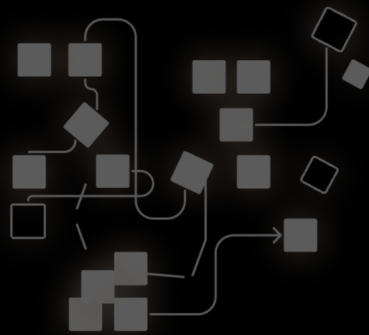
What's a quality exec brief? Who do you thread in next, and how? That quality and counsel is contextual, and can really only be defined by the outcome, determined over time in a sales cycle: did the customer win, or not? And did they win in a way that was non-obvious, counterintuitive, and the result of a judgement call? This is what a GTM team hires and pays for: what sales managers, enablement, RevOps, and leadership all collectively do, rooted in their "tribal" knowledge. They make judgment calls, to guide deals and GTM through gray, ambiguous scenarios.

Ultimately, that means value for our customers comes from the "thing" that makes tasks and tokens useful in the first place: accumulated context, driving nuanced and creative judgement, based on high-stakes outcomes.

GTM Drift: The Silent Cost of Not Capturing Judgement

Today's tech stack can't develop the kind of "tribal knowledge" that top reps use to proactively make those strategic choices:

(1) building siloed AI agents can't trace patterns across 1M+ GTM data points



(2) SaaS doesn't actually *do the work*



When "tribal" knowledge isn't captured, something predictable happens: the organization drifts off the known path to more wins, deal by deal, quarter by quarter. Just like a model "drifts" and constantly needs to learn and adapt, **GTM Drift** works the same way, at two levels:

- **Org Drift:** The company-wide version. Messaging that used to land stops landing. New hires trained on what *used to* win, but nobody notices it until revenue drops.
- **Deal Drift:** The individual-deal version. A deal that should close starts to wobble: the wrong stakeholder gets threaded in, a signal gets missed, and the deal quietly goes sideways.

The only real antidote to drift is a system that continually captures what's actually working, adapts, and course-corrects live. Which is why a new AI layer is needed:

The New GTM Stack

Based on all of this, the GTM software stack is splitting into new layers:

1. **Systems of record** (databases)
2. **A new middle layer: GTM Judgment.** Built on the tribal knowledge that tells AI agents what to do, in what order, for a specific situation, by tracing the path to outcomes over time.
3. **Systems of action** (execution)

Our "bill for labor" thesis was an to answer the monetization question ("how do we track usage and align pricing to work delivered?"), but the temptation has been to focus on the *outputs* as the thing customers buy.

When we're actually selling the *conditions* for good work, not just the work itself.

Those conditions are created by the judgement derived from a unique context layer, that only we create, and is scarce. Meanwhile, the pure "sell work" companies are commodities: e.g. Claude and RevOps teams building with their new superpowers.

What This Means for Us

Our positioning, "Hire Olli," is evolving from its start in the "sell work" camp. It's served us well in 2025: it's concrete, simple to understand, and an agentic launching point.

As customers get more sophisticated about AI costs, and horizontal tools like Claude get better at content generation, we're evolving to stay ahead and build on what we set ourselves up to do from early choices, and to get our customers what they truly want.

Which is the fun part for us:

What we've built over the years with the underlying model isn't "AI that writes your sales docs." By delivering that, we've built a unique, global dataset that enables us to deliver the judgement enterprise sales teams need. Our content and guidance is the *output* of judgement, not the product itself.

The judgement is now the product.

Part 2: How Olli Thinks & Operates

Elevator Pitch

Hire Olli: the mid-funnel operator for mid-market & enterprise sales

Olli encodes what actually works in your enterprise GTM, boosted by the patterns from 150K+ similar cycles, while learning from your specific won/loss data. Then, he executes what works — without you having to prompt him. Delivering creative, high-value sales plays that get stuck deals moving. While creating new GTM data, which compounds into a living playbook that adapts.

Wayfinding, vs. Turn-by-Turn Directions

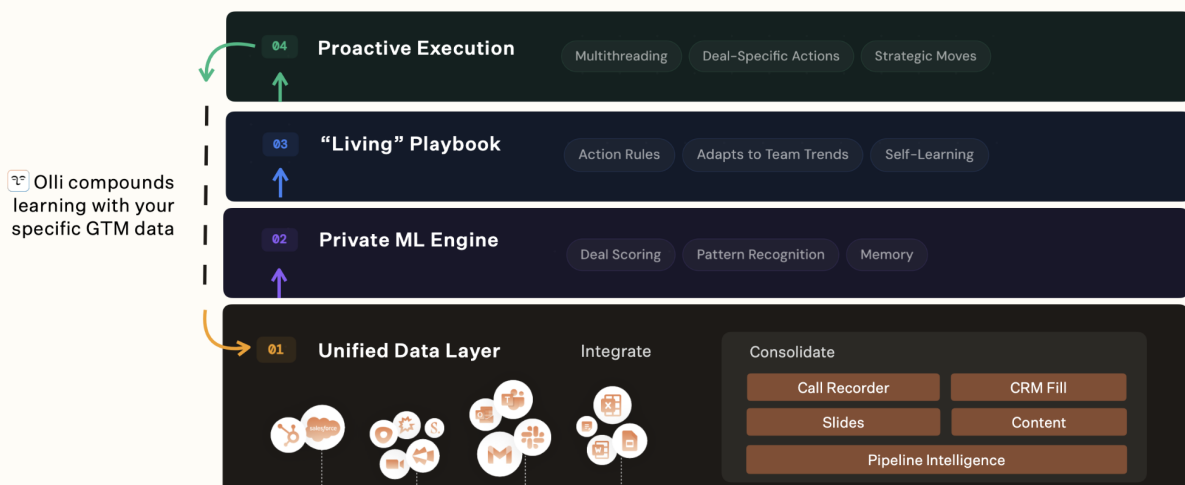
The reason this matters more as companies move upmarket is simple: the higher the stakes, the less a preset route is worth.

- SMB and low-end mid-market deals are like a suburban road. They're relatively predictable, relatively repeatable, and turn-by-turn playbooks do most of the work. "Send this email, book that meeting, run this demo." Rinse and repeat.
- Enterprise deals are open ocean, though. There's no preset route. Every cycle has its own weather, its own stakeholders, its own internal politics, its own triggers that can sink a deal out of nowhere. So what a rep needs isn't turn-by-turn directions, it's more **wayfinding**: the ability to judge and navigate what maps can't show.

Wayfinding is what tribal knowledge *is* translated as a verb. And it's what Olli is built to do.

The Four Layers That Make This Possible:

Olli is the only AI you can trust to make the right, high-stakes judgement calls to get your stuck deals moving:



Under the hood, Olli is a four-layer system designed to capture tribal knowledge, and deploy it as judgement in real time:

1. Unified Data Layer. Every signal that matters — calls, emails, CRM state changes, stakeholder shifts, document engagement, internal Slack threads, buyer-side behavior — pulled out of the customer's stack and normalized into a coherent time-series per deal. We own this pipeline end-to-end so the customer doesn't have to touch it.

2. Private ML Engine. The customer's own model, trained on their wins, losses, reps, and buyers, bootstrapped by patterns from 150K+ similar cycles. This is where we map "what's happening on this deal right now" against "what happened on deals that looked like this before, and how they ended up."

3. Living Playbook. The output of the model, expressed as the plays that actually work for this customer, in this segment, with this persona, at this stage. Not a static PDF. A playbook that adapts as the model learns.

4. Proactive Execution. The content, nudges, and sales plays — exec briefs, multithreading sequences, business cases, next-step recommendations — surfaced where reps already work (Slack, Teams, Claude, Email). The system proposes, the rep executes.

Every cycle Olli runs feeds new data back into the stack, which makes the next cycle sharper. That's the flywheel, where every deal makes the next one smarter.

All of which means Olli is the operator living inside a GTM system that's specifically designed to capture tribal knowledge, fight drift, and do the nuanced, creative execution in complex deals.

Part 3: It's a Data Problem, Not a Prompting Problem

Photos vs. Videos

Think about the difference between a photo and a video.

A photo is a single frame:

It tells you what's true *right now*. You can look at it, describe it, even write a caption. That's what an LLM does with a deal: you hand it a snapshot — the latest call transcript, the current opp record, a few emails — and it generates something plausible about that frame.

A video is thousands of frames in sequence:

The meaning lives in the *motion between them*. You can't understand a story from one still. You need to see the arc: what changed, when it changed, what came before it, and what came after. That's what an ML system does. It goes beyond the single frame to model the *trajectory over time*.

Complex enterprise deals are videos, not photos.

A procurement email on a Friday night at 7:30pm, because the exec can't wait, signals something entirely different during a negotiation, than that email coming the following Wednesday, during a workday.

An LLM alone can only ever see the photo. To actually guide a deal, you need the video — and to get the video, you need ML trained on the motion.

That's why prompt-based systems plateau.

They're trying to describe a story from a single still frame. No matter how clever the prompt is, the output will look “decent,” but it's blind. It can't tell you if a deal is quietly dying 3 weeks *before it dies*, because it's never watched all the other dying deals before.

The Caddy: Course Knowledge + Today's Conditions

One more: Olli is like a great caddy who's played the course hundreds of times. They know the breaks on this green, the hidden bunker on 14, the shots that look safe but aren't. That's tribal knowledge.

But a great caddy doesn't stop there. They read *today's* conditions: wind, humidity, pin position, how your swing is feeling on this particular morning. They combine the durable knowledge of the course with the specific variables of this round, and this shot.

That's the difference between a generic playbook and situational judgement: the course knowledge comes from 150K+ cycles. The conditions come from the live data on this

specific deal. An LLM only reads today's conditions, with no course knowledge. A static playbook has course knowledge but can't read conditions. Olli is built to do both.

Models 101: Why the Small Errors Compound

Quick primer for anyone who wants the mechanics: a model is just a way to capture the relationship between inputs and outputs so you can describe reality accurately. Inputs go in (deal signals, rep actions, buyer behavior), the model learns the mapping, and outputs come out (win/loss, cycle time, ACV).

When a model drifts, the outputs start to diverge from reality.

The same thing happens to GTM. A 2-degree deviation at the start of a cycle is invisible. Ten weeks later, that 2 degrees is a deal you lost and didn't know why. Stack that across a quarter of deals, and you're buried in drift before the number tells you anything is wrong.

The only real defense against that compounding, is a model that's continuously re-tuned by actual outcomes. Which, again, is a data problem, not a prompting one.

It's a Data Problem, Not a Model Problem

Everyone has access to the same frontier LLMs. What they don't have is the dataset, and the pipeline to turn that dataset into something a model can actually learn from.

For every account Olli works, we're ingesting and modeling **literal billions of tokens worth of signal** — calls, emails, CRM state changes, stakeholder shifts, document engagement, internal Slack threads, buyer-side behavior, every micro-interaction that leaves a trace. **Not as a bag of context you dump into a prompt window, but structured as time-series data, where the ordering, spacing, and velocity of events is first-class.**

That lets us do something a prompt can't: compare the trajectory of this deal, right now, against the trajectories of 150K+ similar cycles that already ran to a known outcome. We can see which motions preceded wins, which preceded stalls, and which preceded losses — across dozens of factors interacting at once (champion strength, stakeholder breadth, narrative alignment, timing, competitive pressure, exec coverage, and so on).

That's the GTM Judgment layer. It's not "the LLM is smart." It's "the ML has seen this movie before."

Why This is Hard to Copy

Three reasons the moat here is real:

1. **The ETL is the hard part, and we own it end-to-end.** Getting this data out of a customer's stack — Salesforce, HubSpot, Gong, Salesloft, Clari, Gmail, Outlook, Slack, calendars, document engagement, etc... — and normalizing it into a coherent time-series per deal is brutal, unglamorous, and expensive. We do the full pipeline for the customer, so they don't touch it. Their side is zero-lift. Our side is a system we've been hardening for years.
2. **The labels are earned, not bought.** The ground truth that makes this model useful is "what actually happened to this deal, and why." You can't scrape that. You earn it by sitting inside real cycles, watching them play out, and feeding the outcomes back. Every deal Olli works makes the next deal better, which is a data flywheel, not a prompting trick.
3. **Reproducibility ends at the data layer.** A RevOps team with Claude and a clever prompt can reproduce a *frame*. They can't reproduce a *video library* of 150K+ enterprise cycles with structured time-series signal and outcome labels. That's the part that takes years of product, pipeline, and customer trust to build, and it's the part that compounds.

The Mandate

This is worth sharing with customers, because it's why we believe in this direction:

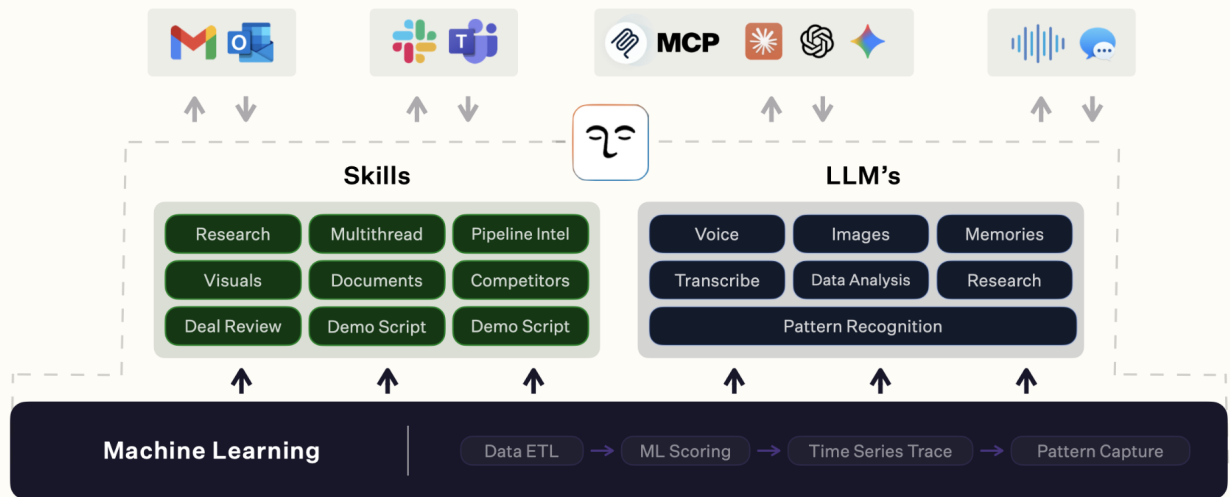
If someone's not running their own machine learning models on their specific GTM data, they're actively guiding their team in the wrong direction.

Competitors can copy a playbook, and they can clone prompts. They can all buy the same horizontal AI. But what they can't do is replicate a customer's own GTM model, because they don't have their data. That's the asset Olli created, protects and compounds for them.

And it's the reason this is genuinely hard to build in-house (and every quarter someone waits, is a quarter of invisible errors compounding into visible ones).

Part 4: Where We Go in Our GTM

Olli shows up **where you work with a single-agent** vs. siloed agent design, built on **deep ML guiding LLM output**



A GTM team's tribal knowledge: the accumulated judgement of top reps, GTM leadership, their best cycles, is their single biggest competitive advantage. It outweighs any one factor like product, pricing, brand, tech stack etc.

What this means for our GTM: this is not a "buy vs. build" world. It's a "come build with us" world.

A lot of smart RevOps teams are building with Claude right now: standing up agents, wiring up skills, automating pieces of the workflow. And it's great. That's their execution layer, and it only will get better over time. GTM Judgment is what makes sure *it works and follows the known path to revenue*.

Because execution without judgement is just fast guessing: an agent confidently running the wrong plays at the wrong moment, because it doesn't know any better. *Vs. LLMs backed by a judgement layer, telling those agents what to do, when, for a specific deal, based on what's actually worked.*

Put differently: if your internal builders are making the engine faster, we're making sure it's pointed in the right direction. Which is the piece that's hardest to build in-house, because it requires the cross-customer dataset and the ETL depth that's a grind.

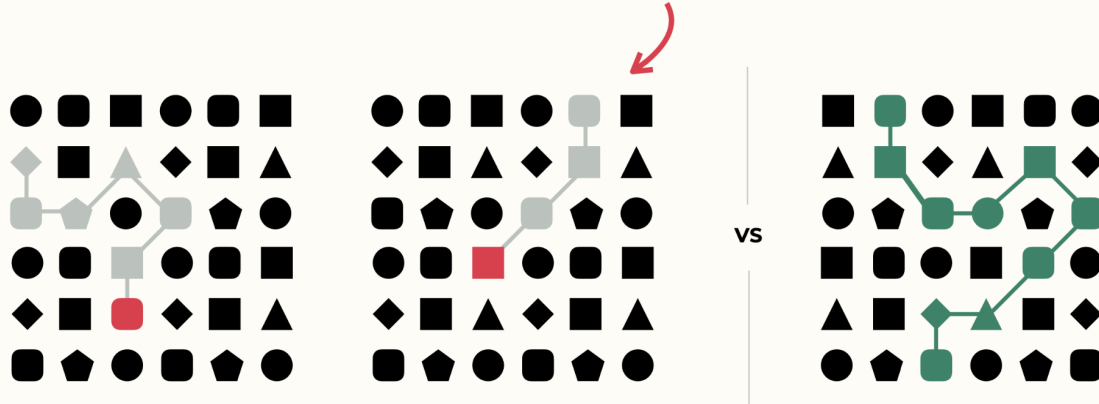
That's how our customers win. Tribal knowledge captured, drift eliminated, and "judgement" wrapped around every rep, every deal, every play. A living model that compounds with every cycle.

- Nate & Jon

Olli in 4 Visuals

1. The Problem:

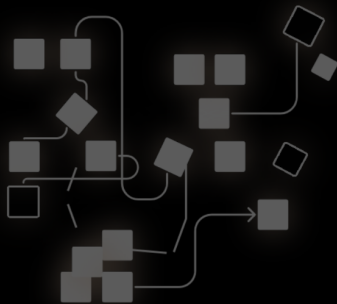
Enterprise deals are full of “gray” and nuanced decisions that take years to master and creates **\$ millions in stalled pipeline:**



2. The Gap:

Today's tech stack can't develop the kind of “tribal knowledge” that top reps use to proactively make those strategic choices:

(1) building siloed AI agents can't trace patterns across 1M+ GTM data points

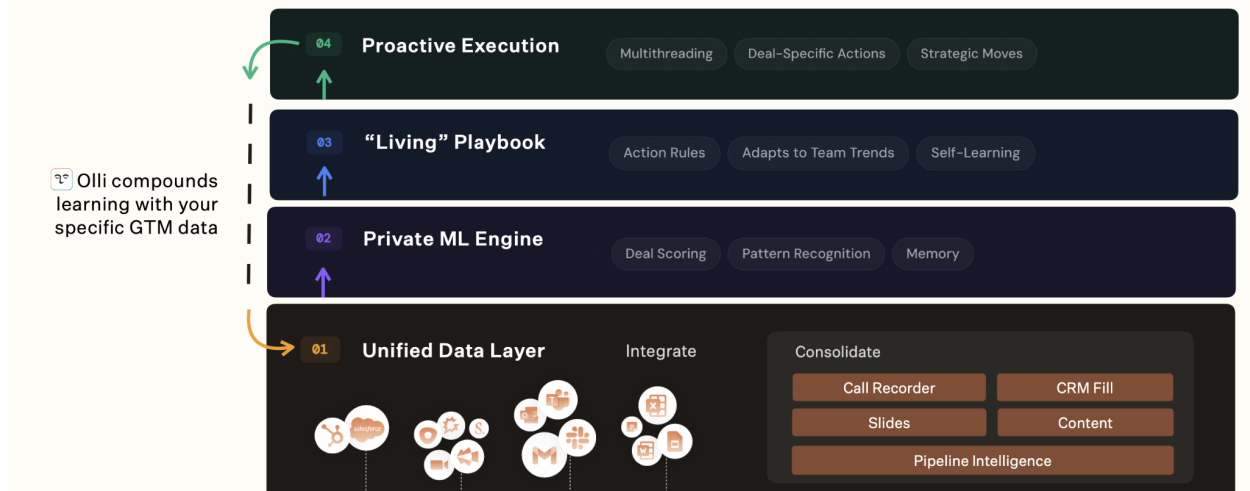


(2) SaaS doesn't actually *do the work*



3. The Difference:

Olli is the only AI you can trust to make the right, high-stakes judgement calls to get your stuck deals moving:



4. User's Experience:

Olli shows up **where you work with a single-agent** vs. siloed agent design, built on **deep ML guiding LLM output**

