WHITEPAPER · 2026

OPTIMAAI

# From Data to Decision

## An architecture for Operating Intelligence in established enterprises

Why most enterprise AI projects fail to compound, what we believe instead, and the technical architecture we deploy to turn structured data and tribal knowledge into an organisation that thinks faster than its inbox.

**OptimaAI · Dubai, UAE**

For corporate readers · Audience: CTOs, CIOs, Heads of Transformation, Directors of Operations

Reading time ≈ 18 minutes

# Contents

# 1. Executive summary

Most established companies do not have a data problem. They have a **latency problem**. The data exists; it is structured, governed, and often beautifully visualised. The gap is the time between something happening in the business and someone — usually a small number of senior people — knowing about it, deciding what to do, and getting that decision executed.

This paper sets out OptimaAI's thesis on enterprise AI, the architectural choices behind our solutions (Business MRI, GetMem, Memory Box), the reasoning behind those choices, and a specific chapter on deploying our stack into Microsoft-anchored environments — Dynamics 365, Power BI, Azure, and the M365 ecosystem. It is written for a technically literate reader: a CTO, CIO, Director of Transformation, or VP of Operations evaluating whether to invest in AI as an operating layer rather than as a feature.

Our core argument is that the next decade of enterprise value will not be captured by replacing systems of record. It will be captured by building a **continuous reasoning layer** on top of them — one that has durable memory of what the business knows, can be queried in plain language by anyone in the organisation, and acts on its own conclusions inside the bounds the business defines. We call this layer **Operating Intelligence**.

> **"We do not replace your ERP, your BI, or your CRM. We turn the gap between them — the place where decisions actually live — into software."**
>
> — OptimaAI design principle

# 2. The thesis

A typical mid-to-large company has spent the last decade buying excellent point solutions: an ERP for transactions, a CRM for relationships, a data warehouse for analytics, a BI tool for reporting, a helpdesk system for tickets, an HRIS for people. Each one works. Each one is governed. And yet decisions still take days, not minutes.

### 2.1 Three structural failure modes

Across hundreds of conversations and dozens of structured diagnostics, we observe the same three patterns:

#### FAILURE MODE A — THE DASHBOARD CEILING

BI tools are exceptional at structured, predefined reports. They are very poor at the long tail of ad-hoc questions that actually drive operations: "Which suppliers had quality incidents in Q1 in region two?" "Which customers haven't been touched in 21 days but had a spike in support tickets?" These questions are unbounded, and dashboards are bounded. So the questions don't get asked, or they get asked through a 30-minute Slack thread to a single analyst who becomes the bottleneck.

#### FAILURE MODE B — THE INSTITUTIONAL MEMORY DRAIN

The most valuable knowledge in any company sits in the heads of about 20 people: the senior PM who remembers why a process exists, the Ops director who knows which supplier is unreliable in monsoon season, the founder who recalls the off-the-books reason a customer left in 2021. When those people are unavailable — on holiday, in another meeting, or simply gone — the organisation forgets. New hires reconstruct knowledge by ambient osmosis over 12–18 months. By then they're senior, and the cycle repeats.

#### FAILURE MODE C — INSIGHT WITHOUT ACTION

Even when an insight surfaces (manually or through analytics) it rarely results in action automatically. A human reads a number, thinks about it, sends an email, schedules a meeting, opens a ticket. The decision-to-execution chain is human-mediated at every link. AI experiments that produce another dashboard add to the pile rather than collapse the chain.

### 2.2 Why "just add an LLM" doesn't fix this

Bolting a chat interface onto enterprise data, in the way many vendors now do, addresses only the surface symptom (accessibility) without touching the root issues (memory and action). A naive deployment looks impressive in a demo and degrades in production for predictable reasons: it hallucinates over edge cases, has no durable memory of past conversations, cannot be audited, and cannot take meaningful action without human intermediation. After 90 days, usage curves flatten and the project quietly becomes shelfware.

The right architecture, we argue, treats AI not as a chat feature but as an infrastructure layer with three first-class concerns: **memory**, **reasoning**, and **action**.

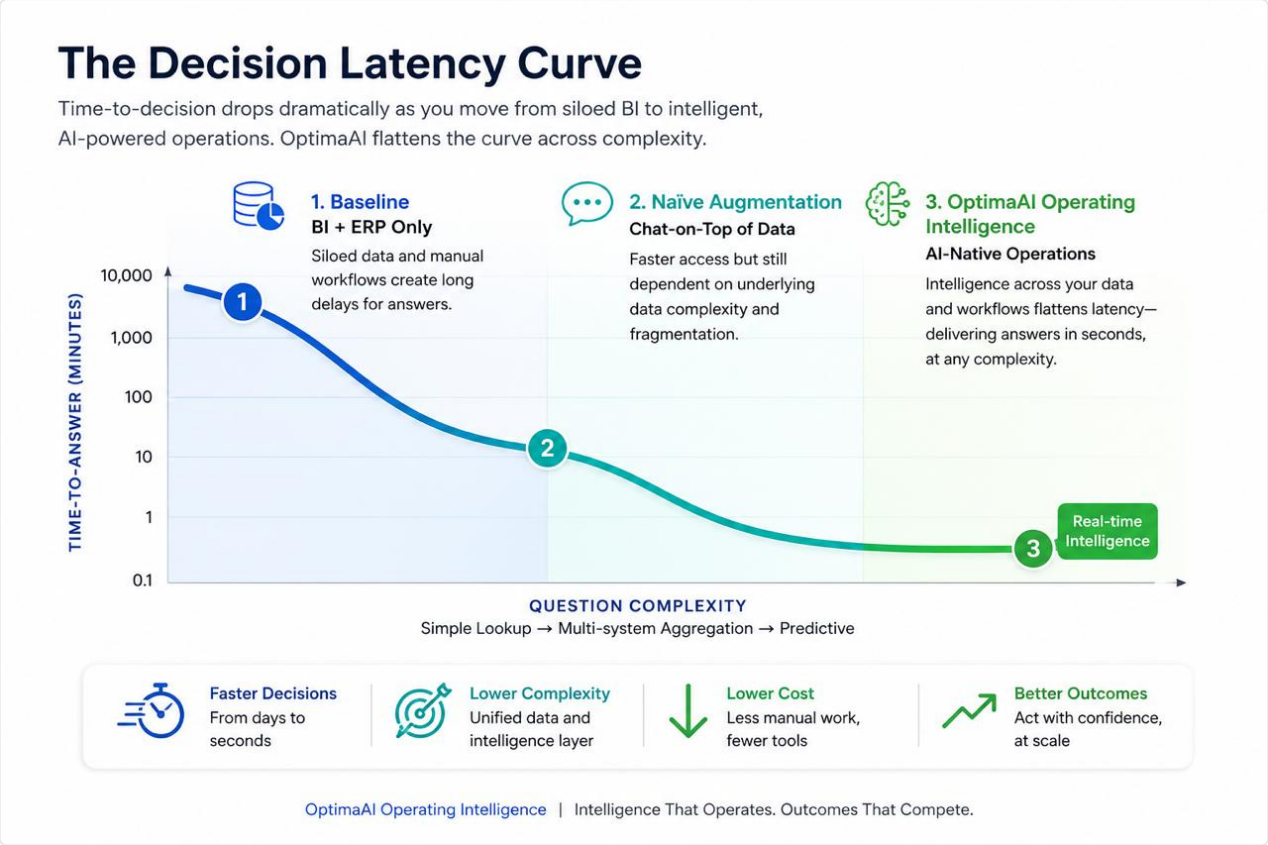


FIGURE 1

The Decision Latency Curve — time-to-answer in minutes plotted against question complexity for three operating states: BI + ERP only, naïve chat-on-top-of-data, and OptimaAI Operating Intelligence. Naive augmentation collapses only the leftmost segment; OptimaAI flattens the whole curve.

# 3. Operating Intelligence

We define Operating Intelligence as the property of an organisation in which the time between an event occurring, the right people understanding its implications, and the appropriate action being initiated, approaches the speed of the underlying systems rather than the speed of the slowest human in the chain.

An organisation has Operating Intelligence if, and only if, three properties hold simultaneously:

1. **Total recall.** Every meaningful piece of context — past decisions, customer history, previous troubleshooting, why a process exists — is durably captured, semantically indexed, and retrievable. Memory is not a feature of one tool; it is a substrate that all tools draw on.
2. **Conversational surface area.** Anyone authorised — a store manager, a finance analyst, a procurement officer — can ask a question in their own language and receive a

grounded, traceable answer in seconds, with citations to underlying systems. The cognitive cost of "asking the data" approaches zero.

3. **Bounded autonomy.** The system can take pre-approved categories of action without human mediation: drafting a PO when stock crosses a reorder threshold, escalating a ticket when sentiment shifts, generating an exception report when a KPI drifts more than two standard deviations. Each action is logged, reversible, and bound by policies the business writes.

The order matters. Memory is foundational; reasoning depends on memory; action depends on grounded reasoning. Inverting this order is how AI projects produce confident-sounding wrong answers.

# 4. The Business MRI

Before any technology lands, we run a **Business MRI**. Most consultancies sell strategy by the pound; we sell diagnosis. The MRI is a structured, six-scan instrument that produces a quantitative picture of where the organisation is haemorrhaging speed.

### 4.1 The six scans

| SCAN | WHAT WE MEASURE | OUTPUT |
|---|---|---|
| **1. Approval Chains** | Where authority lives, how many hops a typical decision takes, where chains fork or stall. | Approval graph + median latency per chain. |
| **2. Knowledge Gaps** | Tribal knowledge concentration: how many critical processes depend on fewer than three named individuals. | Heat map of single-points-of-failure by department. |
| **3. Process Failures** | Where work is reworked: how often outputs return to earlier stages, and why. | Rework-rate per process, root-cause clusters. |
| **4. Communication Debt** | Volume of "ping-back" loops: messages whose only purpose is to chase a previous unanswered message. | Communication-overhead index per team. |
| **5. Escalation Patterns** | Which issues bubble to executives that shouldn't, and why frontline staff couldn't resolve them. | Escalation taxonomy + delegation gap report. |
| **6. Org Health Score** | Composite of after-hours activity, meeting-to-output ratio, voluntary tool usage, response-time decay. | Single 0–100 score with trend. |

The MRI is deliberately read-only. We do not change processes, redesign org charts, or recommend headcount actions during the diagnostic phase. We surface what is, with evidence. The findings are presented to the executive team in a structured walkthrough that — and this matters — quotes the organisation's own people back to themselves. When 69 individual employees, by name and department, have each pointed at the same approval chain as broken, no consultant's slide is needed to make the case.

### 4.2 Why diagnosis precedes prescription

Two structural reasons. First, every organisation is convinced its problems are unique; they are almost never as unique as believed, but the proof must come from their own data, not from our benchmarks. Second, the act of producing the diagnosis builds the data spine that the eventual Operating Intelligence layer runs on. The MRI is not just an audit — it is the first ingestion of durable organisational memory.

> **"We do not start with a solution looking for a problem. We start with a measurable picture of where minutes are being lost, and let the picture choose the solution."**

---

# 5. The technical stack

Our stack is opinionated by design. Each layer exists because we have seen what happens when it is absent or substituted with the obvious alternative. This section walks through the four layers and the reasoning behind each choice.

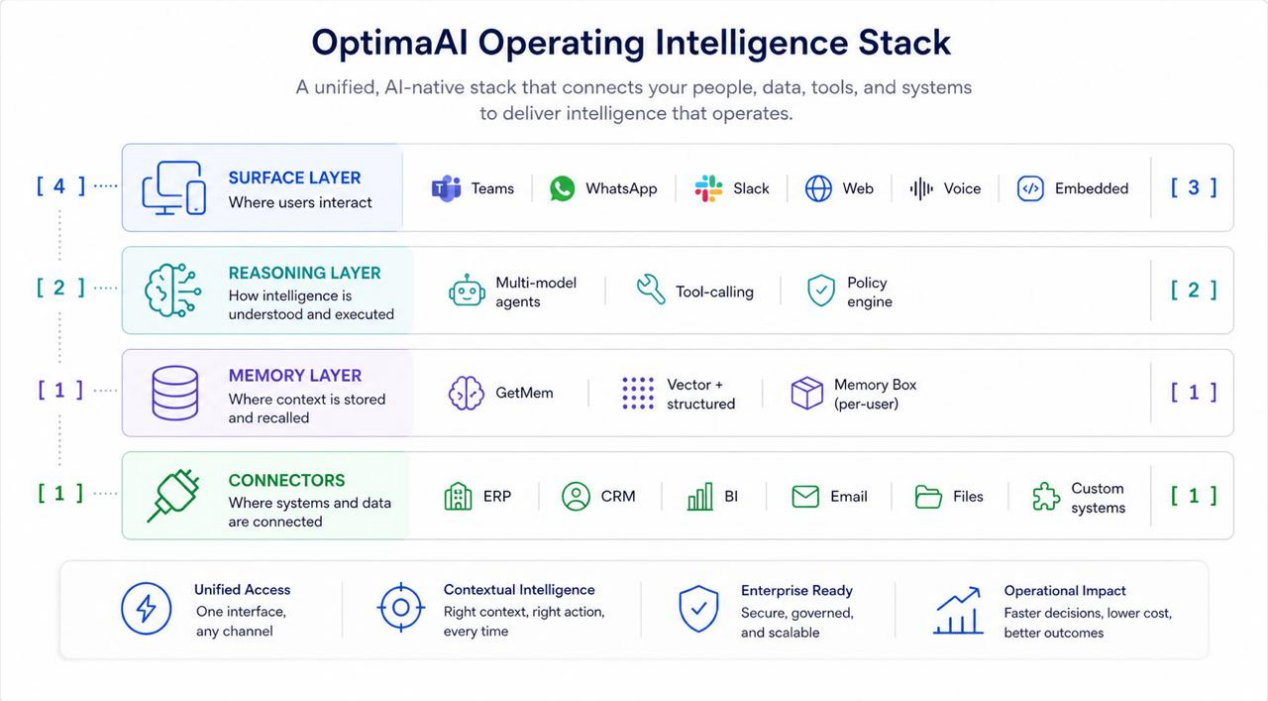


FIGURE 2

The OptimaAI Operating Intelligence Stack — four layers: Connectors read from systems of record; the Memory Layer (GetMem + Memory Box) holds episodic, semantic, and personal memory; the Reasoning Layer orchestrates multi-model agents under a policy engine; the Surface Layer meets users in Teams, WhatsApp, Slack, web, voice, and embedded contexts.

### 5.1 Layer 1 — Connectors

The connector layer reads from existing systems-of-record without altering them. We do not write back to source systems by default; that surface is opened explicitly per use case, with audit and rollback. Connectors are built around four primitives — entity, event, document, and metric — so that downstream layers see a consistent shape regardless of whether the source is Dynamics 365, SAP, a Salesforce instance, or a 1990s text-file batch drop. A connector is considered finished when it survives a week of production traffic with idempotent retry semantics, not when it produces its first successful read.

### 5.2 Layer 2 — Memory

Most enterprise AI stacks treat memory as an afterthought: a vector store bolted on for retrieval-augmented generation. We treat memory as a product. Our memory subsystem (GetMem) holds three categories of information distinct from each other and queried differently:

- **Episodic memory** — what happened: meetings, transactions, tickets, emails. Time-stamped, source-cited.
- **Semantic memory** — what is true, in general: definitions, processes, policies, product specs. Versioned.
- **Personal memory** — per-user context: preferences, ongoing projects, names, the shape of someone's working life.

The reasoning layer above does not query "the database"; it queries the appropriate memory type for the task. This single architectural decision is responsible for the largest gap in answer quality between naïve and well-built systems.

### 5.3 Layer 3 — Reasoning

Reasoning is performed by an orchestrated set of agents, each with a narrow remit (an outreach agent, a research agent, an analysis agent, a meeting-summariser agent), all coordinated by a controller that selects the right model for each step. We are model-agnostic and model-pluralistic by deliberate choice: Claude for nuance and long-form reasoning, GPT for breadth and tool-calling reliability, smaller open models for high-volume classification. Routing is policy-driven, not vendor-driven, and the routing rules themselves are an editable artefact the customer owns.

Above the agents sits a **policy engine**: declarative rules describing what kinds of action require human approval, which roles can ask which questions, what data may leave which boundary, and which actions are reversible. This is what allows bounded autonomy without surrendering control.

### 5.4 Layer 4 — Surface

Users meet the system where they already work. For most of our customers that means Microsoft Teams, WhatsApp Business, or a Slack workspace — not yet another web app to log into. The surface layer handles authentication pass-through, conversation threading, attachment ingestion, and graceful failure. It also enforces the same policies the reasoning layer enforces, so a question that would be denied in the API is denied at the chat surface as well.

# 6. Memory as a first-class primitive

GetMem is our memory infrastructure: an API and storage layer that any agent in the system, and any integration the customer builds in future, can write to and read from with consistent semantics.

### 6.1 Design properties

- **Source-cited.** Every memory carries provenance: where it came from, when, and under what circumstances. An agent answering a question shows the user the originating document, message, or transaction.
- **Time-aware.** Memories have validity windows. A pricing rule from 2023 is not retrieved as current; a customer's stated preference six months ago is weighted lower than one stated last week.
- **Hybrid retrieval.** Vector similarity for fuzzy semantic recall, structured filters for precise constraint satisfaction, and graph traversal for relationship questions ("which clients introduced which other clients?"). Each query type is served by the appropriate index.
- **Boundary-aware.** A memory written by the procurement agent is, by default, not visible to the marketing agent unless an explicit policy permits it. Memory leakage is the most common cause of embarrassing AI failures; the data model prevents it by construction rather than by hope.

### 6.2 Why we built our own memory layer

We evaluated existing offerings carefully. Hosted vector databases solve indexing but not provenance, time, or boundaries. Open-source memory frameworks model conversation history but not enterprise semantics. Building our own layer was, in 2025–2026, the only way to deliver memory with the four properties above as primitives rather than as customer-built epicycles. We expect this to commoditise over time; until it does, we ship the layer ourselves.

### 6.3 The compounding effect

A well-designed memory layer compounds: every interaction improves the next one. After three months of operation, the system not only knows the answer to "what did we agree with this customer in Q1?", it can pre-empt the question. After twelve months, it functions as the institutional memory the company always wished it had documented. The practical implication for the buyer: the value of Operating Intelligence is not realised in week one. It is realised when the memory layer has been written into for long enough to begin paying back compound interest. Our pricing model reflects this.

# 7. The reasoning loop and Memory Box

Memory Box is the per-customer reasoning environment in which agents think. It is the orchestration layer above GetMem: a controlled execution environment where agents can plan, call tools, query memory, draft outputs, and submit actions for approval.

### 7.1 The loop

Each agent operates in a deterministic loop:

1. **Receive** a request — from a user, a scheduled trigger, or an upstream agent.
2. **Recall** relevant context from GetMem (episodic, semantic, personal as appropriate).
3. **Plan** the steps needed to satisfy the request, choosing tools from a permitted set.
4. **Execute** the plan, tool-call by tool-call, accumulating intermediate evidence.
5. **Verify** the result against a checking rubric (consistency, evidence, policy compliance).
6. **Persist** the conclusion and its derivation back to memory; raise an action proposal if one is warranted.

Steps 5 and 6 are where most enterprise deployments fall short. Verification is treated as a hopeful afterthought ("the model usually gets this right"); persistence as an afterthought to verification ("we'll log it somewhere"). Both are first-class steps in our architecture, with explicit interfaces.

### 7.2 Tool-calling and the permission model

Tools are the verbs the system can use: `send_email`, `create_po_draft`, `query_dynamics`, `post_teams_message`, `book_meeting`, etc. Every tool is registered with a typed schema, an idempotency key, and a permission descriptor that defines which agents may call it under which conditions. Permissions are evaluated at the policy engine, not at the agent. An agent cannot grant itself access; it can only request access, which the policy engine grants or denies on the basis of declarative rules.

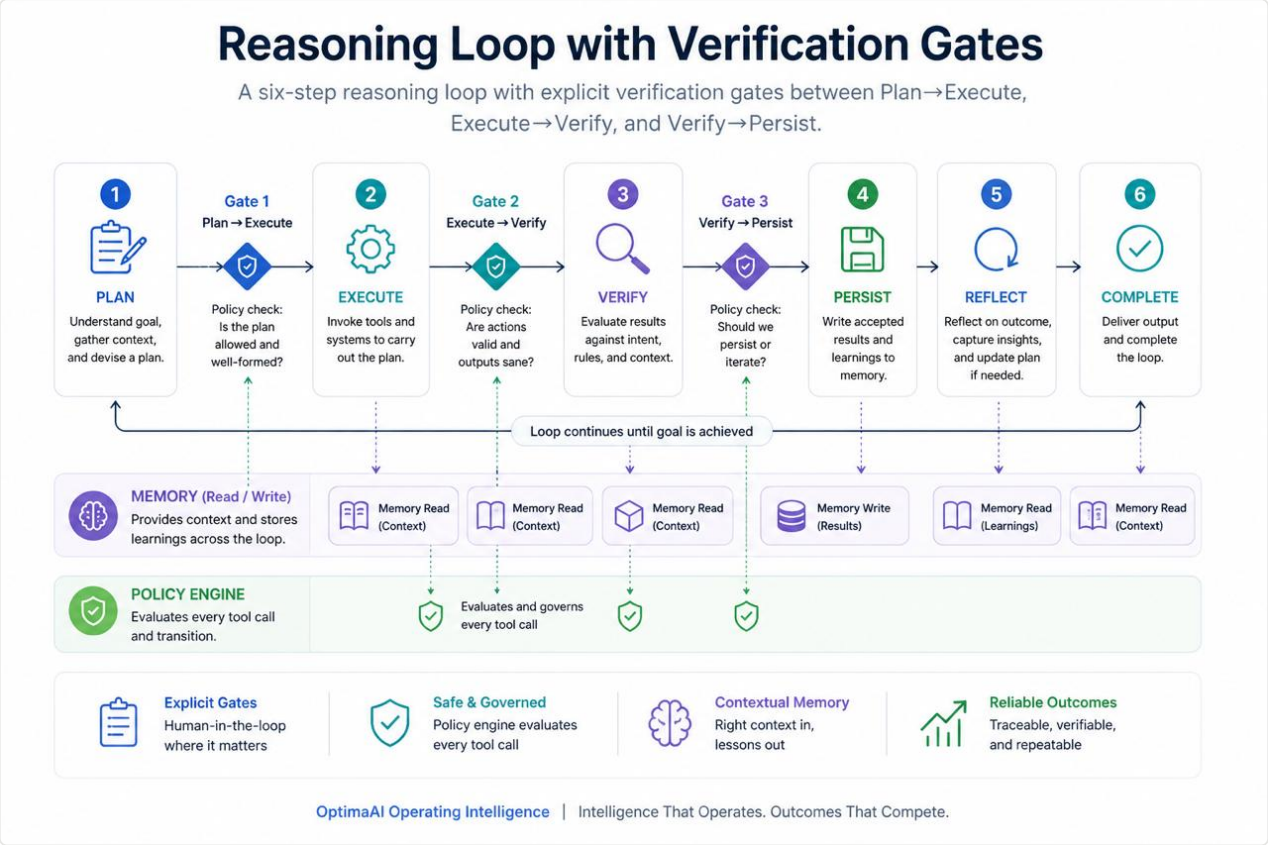


FIGURE 3

The Reasoning Loop with Verification Gates — a six-step deterministic loop (Plan → Execute → Verify → Persist → Reflect → Complete) with three explicit policy-engine gates. Memory reads supply context at every stage; memory writes persist accepted results and learnings.

# 8. For Microsoft-anchored enterprises

A meaningful proportion of the established companies we work with run on Microsoft. Dynamics 365 ERP, Power BI for analytics, Azure for compute, M365 for collaboration. Our position to such companies is not "rip and replace". It is the opposite: **your foundation is good. We sit on top of it.**

### 8.1 Where the Microsoft stack is already strong

Dynamics 365 holds a clean transactional record. Power BI produces governed, board-ready reports. Azure offers compliant compute with Active Directory integration. M365 is the universal collaboration surface. A Microsoft-shop CTO has done the difficult work of choosing and integrating a coherent stack. We do not pretend otherwise.

### 8.2 Where the structural gap is

The gap, with rare exceptions, is the same one we see in every enterprise — but expressed in Microsoft idiom:

- **Power BI is excellent for the questions you knew to ask.** It is poor for the long tail of unforeseen questions that operations actually depend on. A store manager wondering "which SKUs in my store have sat for over 90 days and have a regional comparator that's selling well?" cannot self-serve that question through Power BI without analyst help.
- **Dynamics tells you what happened.** It does not tell you what is about to happen — except through reports a human has to remember to run. Predictive flags ("this batch is on a trajectory to expire before sell-through at this store") require an intelligent layer above Dynamics, not inside it.
- **M365 has the data but not the action.** Teams, Outlook, SharePoint, and OneDrive between them contain most of the institutional memory a company has. They retrieve it on demand, but they do not act on it. A meeting transcript that mentions a deadline does not, by itself, create the calendar entry.

### 8.3 How OptimaAI integrates with the Microsoft stack

We meet the customer at the seams of their existing investments:

#### CONNECTORS

- **Dynamics 365 (F&O / BC / CE)** via the OData and Dataverse APIs, with read-only entity ingestion to GetMem and write-back limited to clearly-bounded tools (PO drafts, case notes).
- **Power BI** via the REST API and dataset XMLA endpoints — we read measures and tables, but we do not replace dashboards. Where appropriate, we publish back as a tile that summarises the OptimaAI conversation in board-ready form.

- **Azure SQL / Synapse** for direct warehouse access where the customer has already centralised reporting data.
- **M365 — SharePoint, OneDrive, Outlook, Teams** via Microsoft Graph for documents, mail, calendar, and Teams messages, governed by the customer's existing Conditional Access and DLP policies.

**SURFACE**

- Native **Teams app** as the primary user-facing channel — single sign-on through Entra ID, conversations indexed into GetMem under the customer's tenancy, no external chat UI to roll out.
- Optional **Outlook add-in** for inline drafting against memory.
- **Power BI custom visual** for embedding OptimaAI-grounded narrative directly into existing dashboards.

**INFRASTRUCTURE**

- Deployable into the customer's **Azure tenancy** as a private workload — AKS or App Service — so memory and reasoning never leave the customer's compliance boundary.
- Identity through **Entra ID**; row-level security in memory queries inherits Dynamics and SharePoint permissions rather than reimplementing them.
- Audit trail emitted to the customer's existing **Microsoft Sentinel** or **Log Analytics** workspace.

> **"You do not need a parallel data platform. You need an intelligence layer that respects the one you've built."**

### 8.4 A representative scenario

A Dubai-headquartered group with twelve operating companies runs Dynamics 365 and Power BI across the portfolio. The CFO's office produces an excellent monthly consolidation pack. But a regional MD, sitting in a meeting on a Tuesday afternoon, cannot answer the question "which of our Q2 supplier consolidations are tracking below the savings we forecast in the Power BI deck?" without sending a Teams message to the FP&A team and waiting until Thursday.

With the OptimaAI layer in place: the MD asks the same question to the OptimaAI Teams app. The reasoning layer queries GetMem (which has indexed the Q2 forecast deck and the live Dynamics actuals), cross-references the variance, returns a one-paragraph answer with a citation back to the specific Power BI tile and the source rows in Dynamics, and offers — as a follow-up — to schedule a 15-minute review with the relevant supplier owners. Question to grounded answer: under fifteen seconds. Question to action: a single approval click.

The CIO's risk posture is unchanged: data did not leave Azure, identity stayed in Entra, and the audit trail landed in the existing SIEM workspace. That is what we mean by an intelligence layer.

# 9. Deployment topologies

OptimaAI ships in three deployment shapes. The choice is governed by data sensitivity, compliance posture, and the customer's appetite for operational ownership.

| TOPOLOGY | WHERE IT RUNS | BEST FOR | TRADE-OFF |
|---|---|---|---|
| **In-house** | Customer-controlled hardware or private datacentre. | Regulated industries, defence-adjacent work, customers with strict data-residency obligations. | Highest control; longest deployment timeline (8–12 weeks); customer assumes operational responsibility. |
| **Private cloud** | Customer's own Azure / AWS / GCP tenancy. | Microsoft-anchored enterprises with mature cloud governance. | Excellent compliance posture; shared operational responsibility (we run the workload, customer owns the boundary). |
| **Managed** | OptimaAI-operated multi-tenant infrastructure with isolated per-customer memory. | Mid-sized companies, faster pilots, customers prioritising time-to-value. | Fastest to deploy (4–6 weeks); requires data-processing agreement; isolation is logical, not physical. |

All three topologies share the same software, the same memory architecture, and the same agent framework. The difference is where the workload runs and who holds operational responsibility. The intent is that a customer can begin in managed for speed, and migrate to private cloud once value is proven and procurement has caught up.

### 9.1 The framework path

For customers who prefer not to depend on us in the long term, we offer a **framework handoff** after the first production phase. Source code, infrastructure-as-code, the policy engine, and the memory schemas are transferred to the customer's engineering organisation, with a structured handover programme. This option exists because we believe the right relationship between an intelligence vendor and an enterprise should not be a per-

manent dependency. The companies who choose this path tend to be those with mature internal platform teams; the companies who do not tend to be those who would rather keep their engineering attention on their core product.

# 10. The economic argument

Operating Intelligence is not bought as a feature. It is bought as an investment whose returns compound over the first 12 to 24 months. This shapes how we price and how we measure return.

### 10.1 Where the value lands

The MRI typically surfaces value in three categories, in order of how quickly each can be realised:

1. **Decision latency reduction (months 1–3).** Median time-to-answer for ad-hoc operational questions falls from hours to seconds. The accumulated saving at executive level alone is usually visible inside the first quarter.
2. **Process recovery (months 3–9).** Quietly broken processes — approval chains that fork and don't reconverge, escalations that should have been resolved at frontline — are surfaced and fixed. The MRI tells us which to prioritise; the operating layer enforces the fix.
3. **Predictive value (months 6–18).** Once memory has accumulated enough operational history, the system begins to forecast: stock-outs before they happen, customers about to churn, processes drifting toward failure. This is the largest of the three categories and the slowest to mature.

### 10.2 How we price

Our default engagement shape mirrors how the value lands:

- **Phase 1 — Business MRI.** A fixed-fee diagnostic, usually completed in 4–6 weeks. The fee is recoverable against any subsequent build phase.
- **Phase 2 — Demo and MVP build.** Two short cycles: a tangible demo within four weeks, and a production-grade MVP within ten. Each cycle ends with a live walkthrough.
- **Phase 3 — Production and managed retainer.** Either a managed monthly engagement or, for companies who choose the framework path, a one-time handoff with a structured support tail.

We publish indicative ranges per stage; final scope is agreed against the MRI's findings rather than against a list price for vapourware.

# 11. What we don’t do

A coherent thesis is defined as much by its exclusions as by its inclusions. The following are deliberate choices, not capability gaps.

- **We do not replace ERPs, CRMs, or BI tools.** Every replacement project we have ever seen, including those we were asked to lead, has cost more than projected and has produced less value than building an intelligence layer above what existed. We respect the foundation, even when it is imperfect.
- **We do not deploy AI as employee surveillance.** We are explicit with customers in our diagnostic phase: we measure processes, not people. Individual performance scoring is not produced; content of communications is not read; CEO dashboards see patterns and aggregates, not transcripts. This boundary exists for ethical reasons and for practical ones — surveillance-flavoured AI projects collapse from internal resistance long before they realise value.
- **We do not promise headcount reductions.** Our objective is reduced decision latency, not reduced headcount. In practice, customers re-deploy capacity rather than remove it. Pitching otherwise produces a politically unviable project.
- **We do not lock customers into our infrastructure.** The framework handoff path exists precisely to make this credible. A customer who chooses managed today should be able to migrate in 18 months without rebuilding.
- **We do not train models on customer data.** Memory is per-customer; reasoning is per-tenancy; no cross-customer model fine-tuning happens. This is policy, not aspiration.

# 12. From reports to reflexes

The companies that will compound value through this decade are not those that buy the most AI. They are those that turn AI into the connective tissue between their existing systems — the layer that remembers, reasons, and acts within bounds.

The architectural choices we have made — memory as a first-class primitive, model-pluralistic reasoning, declarative policies, surface-where-the-user-is, three honest deployment topologies — reflect a decade of seeing what does and does not survive contact with real organisations. Nothing in this paper is novel in isolation. The novelty, such as it is, lies in refusing to ship the layer until each of the four parts is in place. A demo without memory looks magical; a production system without it is a liability.

For the Microsoft-anchored enterprise, our message is straightforward: your stack is good, and we are not here to replace it. We are here to do the one thing it is structurally incapable of doing on its own — turn the gap between systems, where decisions actually live, into software that thinks at the speed of the question being asked.

## Where to go from here

The natural next step is a **Business MRI**: a four-to-six-week diagnostic that produces a quantitative picture of where minutes are leaking from your organisation, with recommendations ranked by recoverable value. It is fixed-fee, recoverable against subsequent phases, and binding on neither side beyond it.

OptimaAI · Dubai, UAE · +971 58 510 2699