DoCoreAI

Autonomous AI Cost Governance

For Applications Using LLM APIs

Saji John · Founder & Builder · DoCoreAI

"Reduce AI costs in real time — without logging a single prompt"

AI Ecosystem: Where DoCoreAI Belongs — and Where It Doesn't



LLM API Apps: The Problem Landscape

Where enterprises lack control over LLM spending.

01. Cost Intelligence & Efficiency
(4 problems)

01 — Cost Optimization
Limited ability to predict and optimize token consumption before each LLM API call.

02 — Token Waste
Static max_tokens limits don't predict actual response length, leading to inefficient token allocation.

03 — Forecasting
Reactive spend reports after the fact; no predictive token intelligence to anticipate usage.

04 — Agentic Cost Runaway
One agentic task can trigger dozens of LLM calls, multiplying token consumption without per-call or per-workflow cost controls.

02. Governance & Accountability
(3 problems)

05 — Visibility
Observability tools show what happened, but data is disconnected from enforcement.

06 — Attribution
No native attribution layer to map AI spend to teams, projects or environments.

07 — Cost Governance
No centralized budget policies or enforcement; teams and apps spend without limits.

03. Architecture & Trust
(2 problems)

08 — Privacy & Compliance
Prompt-level observability can expose sensitive data, creating privacy and compliance concerns.

09 — Gateway Limitations
Proxy-based solutions add network hops, latency, failure points and data-routing concerns.

DoCoreAI: Autonomous AI Cost Governance

Three architectural principles. One new category.

Privacy-First Architecture

- No prompt or response content logged
- Metadata-only telemetry (cost, tokens, latency)
- PII detection at the edge before API calls
- No sensitive data leaves your network

No-Gateway Architecture

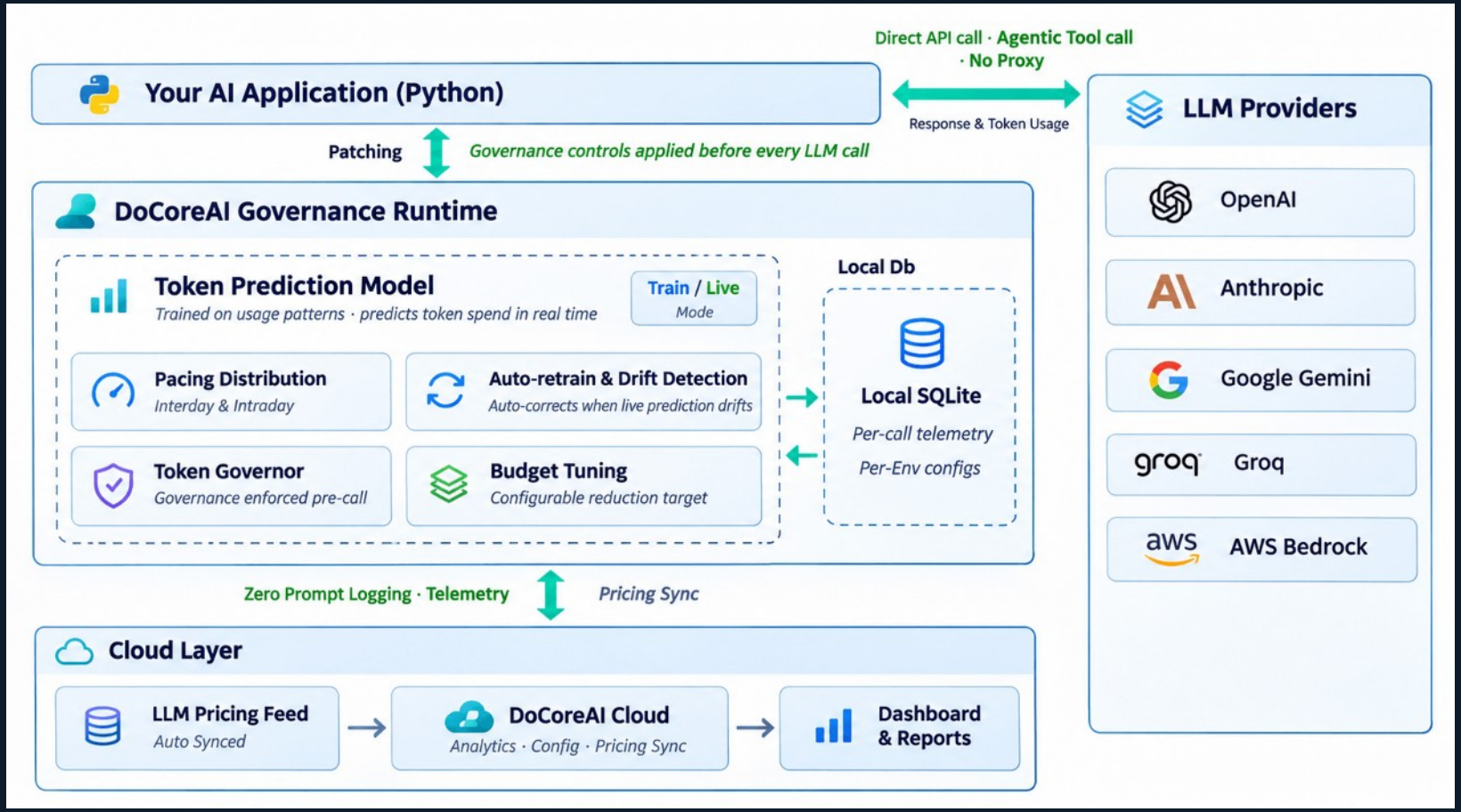
- No external proxy sits in your LLM call path
- No additional network hop latency
- pip install — no application code changes
- Fails open: LLM calls never blocked
- Works alongside your app, not in front

AI Cost Governance & AI FinOps

- Predictive token budgeting
- Dynamic rolling 24-hour budget pacing
- Per-team, project & env attribution
- Automated anomaly detection & alerts
- Governance reports & proof of cost savings

· No Forced Truncation · No Model Downgrade · AI token waste reduced

DoCoreAI Architecture: Privacy-First, No-Gateway AI Cost Governance



DoCoreAI Integration: How DoCoreAI Embeds Into Your Python App

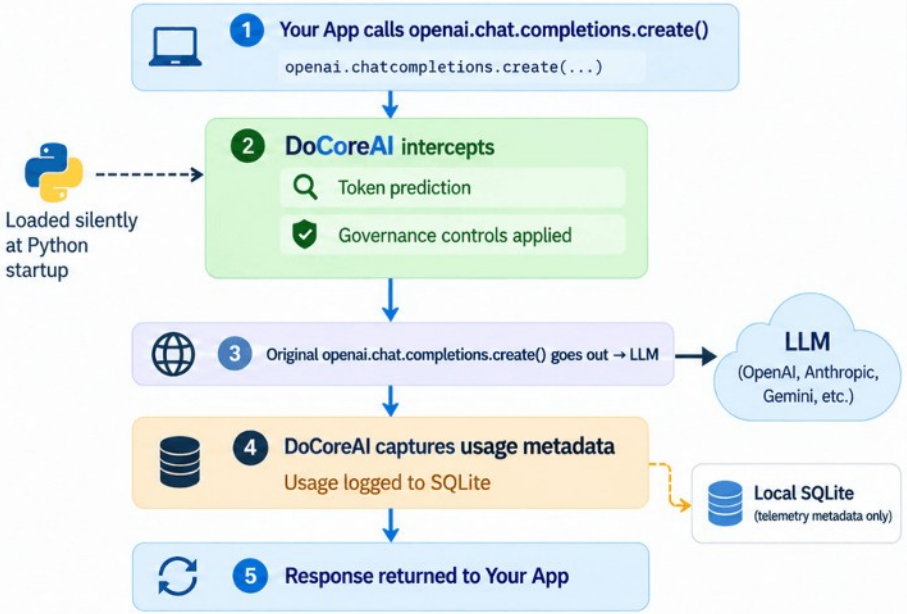
Like a toll booth on a highway

Your car still reaches the same destination, but every vehicle is recorded and governed as it passes through.



You never change your driving route.

How DoCoreAI Intercepts and Governs LLM Calls



DoCoreAI governs every LLM call along the way.

DoCoreAI: The Autonomous Learning & Governance Loop

DoCoreAI doesn't just monitor — it acts autonomously on your behalf

01 LEARN (Every LLM call observed locally)

Learns from metadata-only LLM usage patterns across each environment.

02 MODEL (Train)

Builds a token prediction model specific to your usage — not a generic baseline

03 ALLOCATE (Predict token usage)

Allocates budget across predicted hourly & daily predicted traffic patterns.

04 MANAGE

Dynamically paces token budgets against real-time spend and predicted usage.

05 CORRECT

Detects prediction drift and automatically retrains as usage patterns evolve.

EXAMPLE — Autonomous Budget Intervention

LEARN	→	MODEL	→	ALLOCATE	→	MANAGE	→	CORRECT
Observe 14 days of usage patterns		Build spend prediction per hour/day/feature		Distribute monthly budget intelligently		Pace requests to stay within allocation		Auto-adjust when deviations detected

How DoCoreAI Governs Every LLM Call — Three Cost Levers

Three Techniques for Real-Time LLM Token Reduction

Soft Limit Injection



Set the limit before generation, so the LLM self-constrains its response — not cut off.

Agentic Tool Pruning



Select and pass only the relevant tools for each step, while keeping the full tool set available.

Semantic Prompt Compression



Preserve the meaning, remove redundancy, and send the same intent to the LLM in fewer tokens.

How DoCoreAI Reduces LLM API Spend — Mechanism & Evidence

INTERVENTION MECHANISMS

01 Output Token Control

Predicts required output tokens per call; sets calibrated ceilings and provides conciseness guidance to reduce actual generated output.

02 Input Prompt Compression

Optionally compresses prompt content before the API call — reduces input tokens without truncation. Applied selectively; not to tool calls.

03 Agentic Tool Pruning

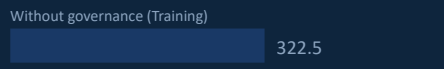
Removes irrelevant tool definitions before each agentic step; reduces repeated per-step context overhead.

04 Rolling Budget Pacing

Dynamically adjusts token budgets against predicted usage and rolling consumption — governs spend across the full session.

MEASURED IMPACT & SCENARIO PROJECTIONS

SINGLE-SHOT CALLS — MEASURED



▼ 33.8%

token reduction per call

15.1% lower cost per call

2,369 training · 2,074 live calls

MULTI-TURN CHAT — PROJECTION

15–20%

output-token reduction
per eligible response

Example scenario

6-turn customer support
conversation

Based on internal test observations.

AGENTIC TOOL PRUNING — PROJECTION

15–25%

input-token reduction
per eligible agentic step

Example scenario

10-tool code agent
2–3 tools relevant per step

Based on internal test observations. Not validated at production scale.

RAG + COMPRESSION — PROJECTION

6–10.5%

input-token reduction
(compression contribution only)

Example scenario

Knowledge base Q&A
65% context from retrieved docs

Output token savings measured separately. Keep rate 85–90%.

Projected ranges are illustrative engineering assumptions, not validated customer savings. Results depend on workload, model, prompt composition and response quality.

Pilot instrumentation live · First enterprise cost-reduction measurement: Q1 2027 | docoreai.com

DoCoreAI: Enterprise Auditability & Policy Lifecycle

Audit & Accountability:

Call-level audit log

Every AI call recorded locally: model, tokens, cost, session, timing, call type

Governance event log

Every blocked or flagged prompt logged with severity, confidence, and action taken — no prompt text stored

Policy & budget change log

Who changed what, old value → new value, timestamp.

Export-ready · docoreai export

Export audit records in JSON Lines format for integration with Splunk or Microsoft Sentinel

Enterprise road-map

H1 2027

End-user identity · Tamper evidence · Configurable retention · Identity provider integration

Policy Lifecycle:

01 LEARN

Learning Window

› Observes actual LLM call patterns.

› Builds a P75 token baseline — learns what each prompt type actually needs

› Activates governance after sufficient usage history.

02 ENFORCE

Per-Call Token Ceiling
ML-predicted token allocation.

Daily Budget Pacing
Controls spend against budget.

Session Budget Throttle
Controls agentic session spending.

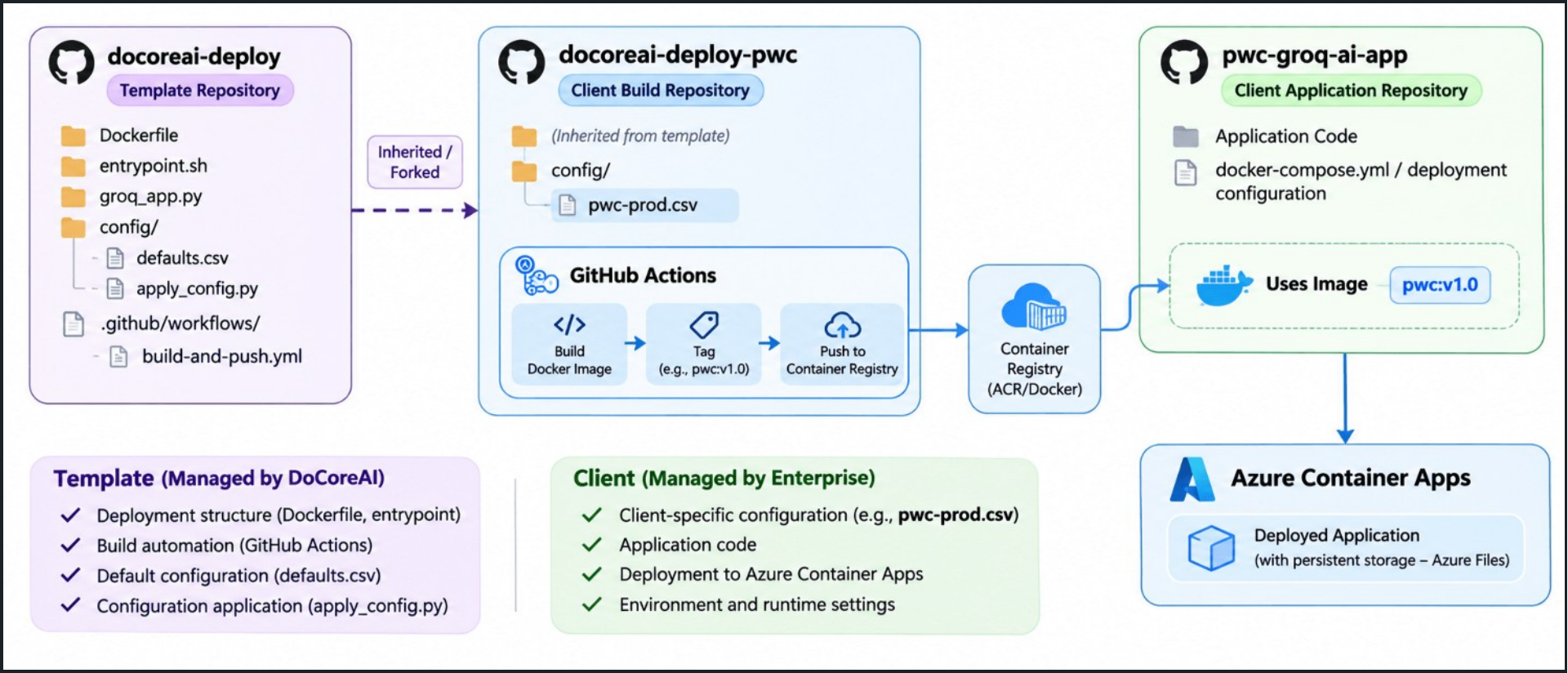
03 OPTIMIZE

Soft Limit Injection
Encourages concise responses.

Tool Pruning
Removes unnecessary tool definitions.

Prompt Compression
Reduces input tokens through optional compression.

Template-Driven & Client-Specific Configuration Management



DoCoreAI: Closing the Enterprise AI Cost Governance Gap

01. Cost Intelligence & Efficiency
4 challenges → 4 capabilities

01 Cost Optimization
Predictive token intelligence and per-call optimization.
✓ Lower cost per call

02 Token Waste
ML-based token ceilings and optional prompt compression.
✓ Reduce unused tokens

03 Forecasting
Spend forecasting and predictive budget pacing.
✓ Predictable spend

04 Agentic Cost Runaway
Agentic loop tracking, tool pruning and session budget controls.
✓ Controlled agentic costs

02. Governance & Accountability
3 challenges → 3 capabilities

05 Visibility
SDK-level telemetry, governance reports and anomaly detection.
✓ Real-time AI cost insight

06 Attribution
Org → Account → Project → Environment cost attribution.
✓ Clear accountability

07 Cost Governance
Autonomous token governor, budget enforcement and policy engine.
✓ Spend within limits

03. Architecture & Trust
2 challenges → 2 capabilities

08 Privacy & Compliance
Privacy-preserving telemetry, local audit records and edge-level PII detection.
✓ Data privacy & compliance

09 Gateway Limitations
SDK-based sidecar architecture; direct application-to-LLM communication without a proxy.
✓ No gateway, no added latency

Server Dependency Audit

DoCoreAI SDK — 13 Server Calls, Client-Side Feasibility Assessment - (**Total assessed effort: ~4 weeks**)

Dependency Group	Server Calls (endpoint · trigger)	Move to Client
Hard Blockers	patch-targets — SDK startup gate (SDK is no-op if empty) model-registry — pricing & routing cache (DEPLOYMENT_MAP_CACHE)	Bundle default patch targets in SDK; ship base model registry with SDK; use LiteLLM public JSON for auto-pricing sync
⚠ Auth & Identity	validate-token — CLI login & SDK init get_member_info — per-call daily limit enforcement	Sign offline JWT license at purchase; cache plan tier & expiry in local SQLite; validate signature locally
⚠ Budget / T→L	budget-recommendation — Training→Live mode switch trigger learning-state — post-retraining email notification	Port P90 budget logic to Python client; server call becomes optional async notification
Telemetry ×5	telemetry · telemetry-failure · org-token-info · accuracy-snapshot · governance-event	Gate all 5 with enable_remote = False — single config flag, zero code changes required
Infrastructure	PyPI version check — CLI startup HuggingFace download — LLMLingua weights (~500 MB, first use only)	Skip PyPI check when DOCOREAI_OFFLINE=1; pre-download & bundle model weights in Docker image
⍰ LLM APIs	OpenAI · Anthropic · Groq · Bedrock · Gemini	N/A by design — Ollama is the air-gap path; no changes needed

Hard — SDK broken without server fix first ⚠ Medium — moderate redesign needed Easy — single config flag gates it ℹ N/A — by design

Product Maturity & Readiness

21mo	5	6+	3
Active research & development since Jan 2025	Agentic Frameworks Supported	LLM providers supported	Commands to integrate DoCoreAI

Key Milestones

Jan 2025	Research begins — identifying key gaps: privacy risks, limited cost visibility, and lack of real-time cost governance.
Mar 2025	First PyPI release (v0.1.0) — LLM cost, token & latency tracking
Aug 2025	v1.0.1 — cloud dashboard
Feb - Sep 2026	v2.4.0 — Predictive LightGBM model, drift detection, auto-retraining, Budget pacing, soft limits, PII detection, multi-provider expanded
Oct 2026	PwC evaluating DoCoreAI partnership

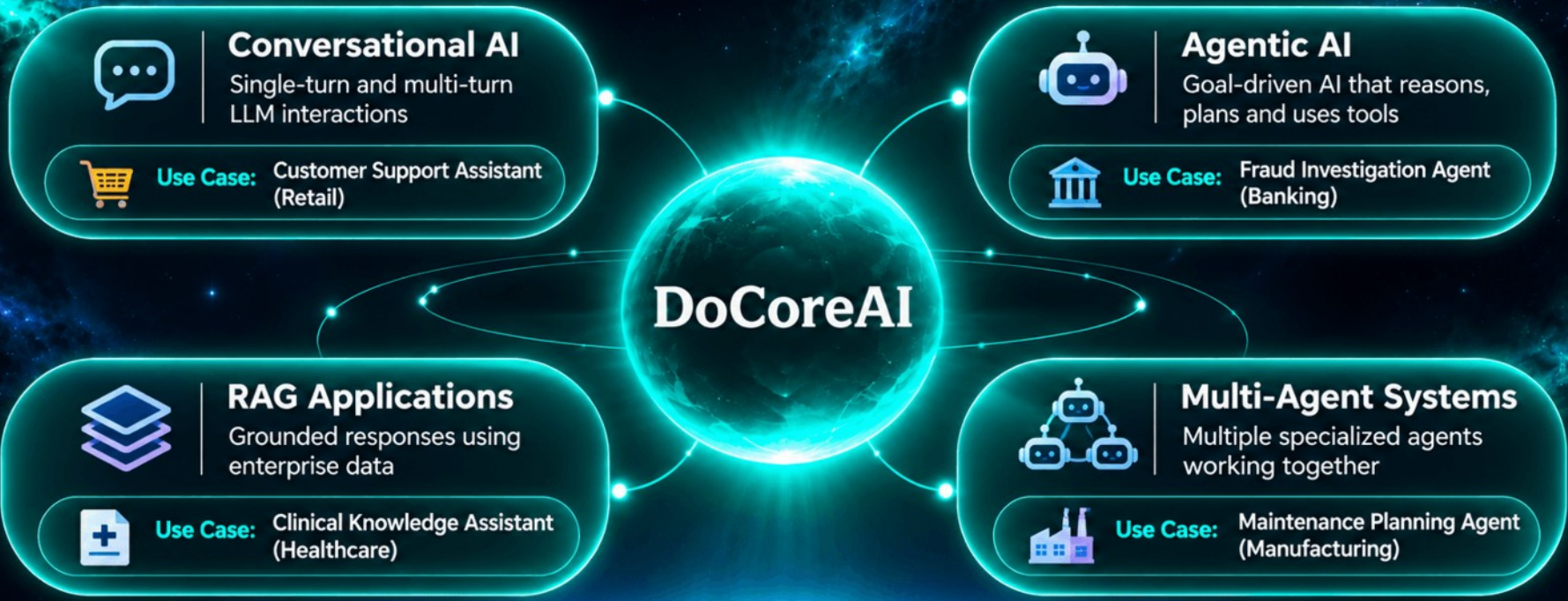
Whitepaper - Research Publication

Metadata-Sufficiency for Privacy-Preserving LLM Cost Governance
Zenodo DOI 10.5281/zenodo.20774715 · Published June 2026

LLM Integrations Supported

Provider	Single-shot	Agentic	Multi-turn
OpenAI	✓	✓	✓
Anthropic	✓	✓	✓
Google Gemini	✓	✓	✓
Groq	✓	✓	✓
AWS Bedrock & Azure	✓	✓	⟳ Deployed
Ollama	✓	✓	⟳ In Development

Appendix: DoCoreAI Across Enterprise AI Application Patterns



Real-Time AI Cost Reduction

saji.john@docoreai.com | +91 9108857655 | linkedin.com/in/saji-john-docoreai | docoreai.com