

A proposal for the AI-agent era

Trust nothing. Delegate everything.

How to build a workplace for AI —
and why the evolved headless CMS is the answer

Kuroco by Diverta Inc.

Internal trusted zone

Core systems & confidential data



Data DMZ = the AI workplace

Inside permissions, audit, approval



Untrusted zone

AI agents / LLMs

HOW TO READ

How to read — shared basics first, then by role

The first half shares why AI needs a dedicated workplace; the second half splits by your role.

Common part

Why AI needs a workplace / The data-DMZ concept / Why a headless CMS is the vessel / Kuroco's three pillars (governance, execution, AI-driven development) / Security, operations, pricing



For operating companies

From “having it built” to “building it yourselves.” How to keep improving operations without engineers: reasons to adopt, the first step, and how to proceed.



For development companies

From competing on build volume to being chosen for governance and design. Proposal patterns, revenue points, and redefined roles in the AI-driven era.

Common closing: answers to the objections you will hear, plus a next action you can decide today

Executive summary

This deck organizes the decision around three points.

1

AI became a worker — but has nowhere to work

Generative AI is everywhere, yet handing over real work fails: there is no framework for permissions, records, or approval.

2

You can't grant it human-level trust

AI can't be a contracting party and bears no liability. Don't wire it into core systems — put a buffer zone in between.

3

The vessel: headless CMS. And AI builds on it

The device for safely handing internal data outward was always the headless CMS — and apps on top are built via AI-driven development.

Conclusion: Kuroco is the evolved headless CMS where AI works safely on business apps

Japan-built, 6,000+ deployments; on July 15, 2026 it became an MCP-native AI platform — governed environment and apps built via dialogue.

The problem

They want to hand work to AI — but can't

AI adoption moved fast. What stalls is not adoption but the handover of real work.

Access

No way to decide which AI may touch what

It ends up all-or-nothing: full data access, or a total ban

Audit

No record of what was read or changed

When something breaks, the blast radius can't be traced

Control

No human approval on critical steps

Nothing prevents “AI finalized it on its own” — so it never ships



AI can't be treated on the same terms as an employee

It's not about capability but liability and explainability. Skip this and you won't get buy-in.

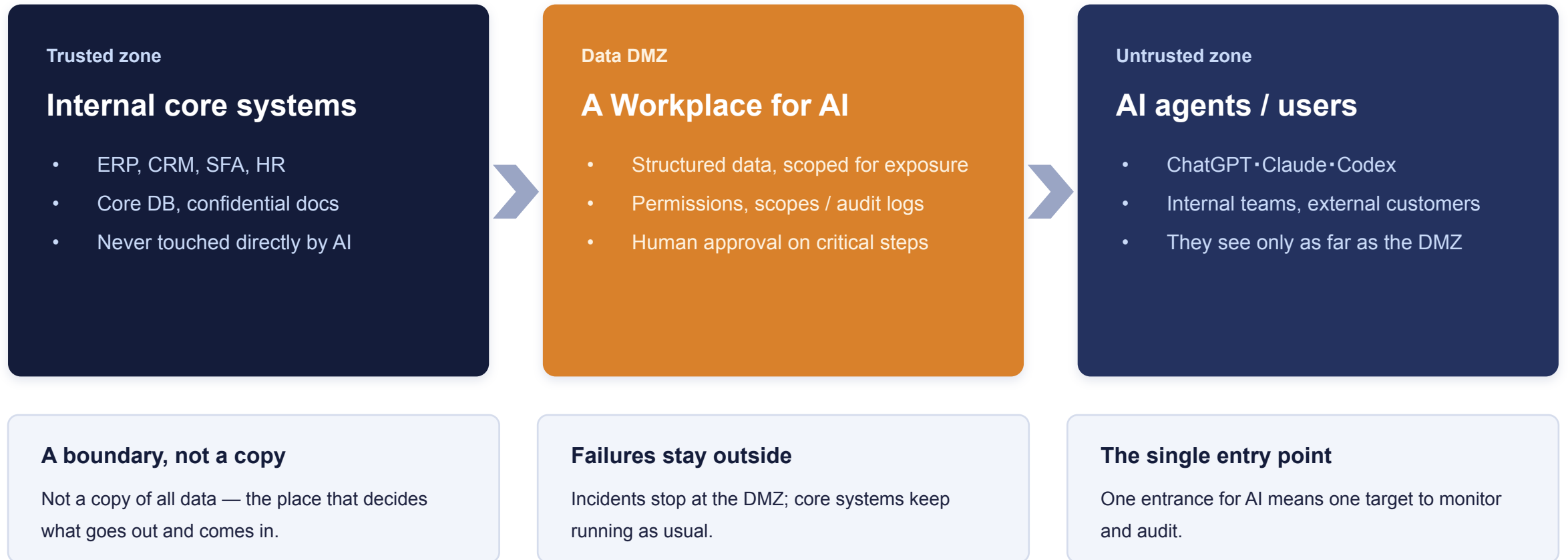
Aspect	Employee (human)	AI agent
Contract	Bound by employment contracts, work rules, NDAs	Can't be a contracting party; you bind the user and the settings
Liability	Recourse exists: reviews, discipline, legal liability	Bears no liability; the cost of errors stays with you
Explanation	Can explain why they did something	Judgments are probabilistic; evidence must be logged externally
Resilience	Notices odd instructions from context	Vulnerable to prompt injection and poisoned data
Scope	Permissions shift naturally with roles	Grows without limit unless scope is designed explicitly

So what AI needs is not “the same keys as employees” but a purpose-built workplace with permissions that stop at its walls.

The answer

A “data DMZ” — a buffer between your core and AI

Standard practice in network design: never connect an untrusted party directly to what must be protected.



The headless CMS was already a “data DMZ”

Editing inside, delivery outside, API as the boundary — built precisely to hand internal data outward, safely.

Structured content

Then (humans): Data with schemas, not pages for human eyes

For AI: Easy for AI to interpret; fewer accidents than loose documents

Exposure & permission control

Then (humans): Decide exactly which data is shown to whom

For AI: Maps directly onto “which AI, how far”

Approval workflow

Then (humans): Human review before publishing, as standard

For AI: Puts a human in before AI's output is finalized

Versioning & logs

Then (humans): Built on recording who changed what, when

For AI: AI's operation trail becomes your audit evidence

API boundary

Then (humans): Keeps internal systems loosely coupled to the outside

For AI: AI-side changes never drag core systems along

Only the reader changes — from human browsers to AI agents. The vessel already exists.

The market agrees — Salesforce is buying Contentful

Placing a governed content layer next to AI agents is becoming the industry consensus.

The facts

- On June 1, 2026, Salesforce announced a definitive agreement to acquire headless/composable CMS Contentful
- Contentful serves 4,800+ brands, including roughly 30% of the Fortune 500
- Closing expected in Q3 of Salesforce's fiscal 2027, subject to regulatory approval
- Reported deal size is in the \$1.0–1.5B range

Why they bought it

- 1 To feed AI agents the raw material**
Lets Agentforce query, assemble, and deliver structured content
- 2 To join customer data with content**
Tied to Data 360; assembled dynamically by context, channel, language
- 3 Because the web's readers changed**
Contentful's founder notes AI agents are overtaking humans on the web

Implication: any company serious about AI in operations will buy — or build — a governed content layer.

Sources: Salesforce news release (June 1, 2026), CMSWire, Salesforce Ben, The Next Web, and other coverage

So how do you bring this shift in-house?

The platform giants create a tailwind — but their answer doesn't fit every company.

1

“AI platform” is being redefined

CRM and CMS are being recast as “where AI works.” Start now and it's a platform project, not a one-off site build.

2

A platform-locked answer isn't always available

If you don't run that platform — or want data and control on your side — you need another option.

3

Domestic & independent: the practical answer

In finance, manufacturing, and the public sector, keeping data and governance in-house is often a hard requirement.

That option is Kuroco — Japan's evolved enterprise headless CMS, 6,000+ deployments

Since 2021. Security, permissions, and approvals honed across finance, manufacturing, and the public sector — now opened as a workplace for AI.

KUROCO

From headless CMS to a workplace for AI

On July 15, 2026, Kuroco moved from API-first to AI-first design.

MCP support and autonomous agents provide a safe, no-development backend —
and the apps on top are built through dialogue with AI.

6,000+ companies

Track record as a headless CMS

2 MCP servers

Client API MCP and Admin MCP

AI-driven

Business apps built via dialogue with AI

Governance — who did what, and how far

Give AI scoped permissions, not keys. Enterprise-grade access control is built in from the start.

1 Nearly every operation is MCP-ready

Client API MCP (~200 operation patterns) and Admin MCP (~150 tools) cover almost the entire admin surface — operations run without opening the admin panel.

2 Connects directly from major LLM clients

Remote connection from Claude, ChatGPT, and IDE assistants.
Authentication via OAuth 2.0 / Bearer tokens.

3 Scope control and read-only mode

Cut scopes per exposed module and start with read-only access, widening permissions step by step.

4 Logging

Records which tools AI called and what changed — audit-ready material as-is.

Key point: instead of wiring AI to all company data, scope what AI sees, module by module.

Execution — AI actually moves the work

Not just search-and-answer: it acts inside business processes, with human approval built in.

Autonomous AI agents

Executes tasks on Kuroco data. A dedicated agent email address lets form submissions or notification emails trigger runs automatically.

Built into approval workflows

Delegate from inquiry handling to approve/reject decisions. AI makes the first pass while logs keep every judgment auditable.

RAG and vector search built in

Auto-ingests and vectorizes content, email, web pages, and Slack/LINE/Teams messages. Knowledge search and answers work with no extra development.

AI processing that protects data quality

“AI post-processing” generates, translates, and transforms on save; “AI validation” checks integrity — fail-closed, every verdict logged.

The admin panel: from a place to operate to a place to supervise

With no training needed, day-to-day updates increasingly go through AI. The panel stays best for a cross-cutting view of the data — humans shift from operating to reviewing and supervising (the on-screen AI assistant takes requests too).

AI-driven development — apps emerge from dialogue

AI writes the code; you describe the requirement in plain language — data design through workflow, via dialogue.

Before: building from scratch

- Write requirement and design documents
- Build DB, APIs, permissions, admin screens
- Test and review; re-implement on every spec change
- Work stalls waiting for engineer availability

Kuroco × AI: built through dialogue

- Say “I want deal management” in plain language
- AI designs and creates the data structures
- APIs, batches, triggers, permissions — prepared together
- Spec changes land via dialogue; prototype and production share one platform

Official skill package: Kuroco Skills for Claude Code

Five skills — docs search, API/content ops, frontend, server processing, admin API — teach Claude Code the Kuroco way up front.

AI-driven, still governed

Permissions, scopes, approvals, and audit logs come from Kuroco itself — in-house building stays inside governance, never shadow IT.

Why AI-driven development here ships — instead of dying at PoC

Most AI-generated apps stall before production because no one can verify them. Kuroco keeps them verifiable by structure.

WHAT AI MAINLY BUILDS

Frontend (screens & display logic)

API =
demarcation

THE BACKEND IS CONFIGURATION, NOT CODE

Data models, APIs, permissions, approvals, auth = Kuroco settings

1 A small review surface

AI-written code is confined to the frontend; the backend is a matter of checking settings. Security review fits a realistic effort budget — so production sign-off actually happens.

2 The API draws the demarcation line

The frontend (AI-built) and backend (configured) are loosely coupled over APIs. Tasks stay small — easy for AI to build, easy for humans to verify.

3 Humans verify in the admin panel

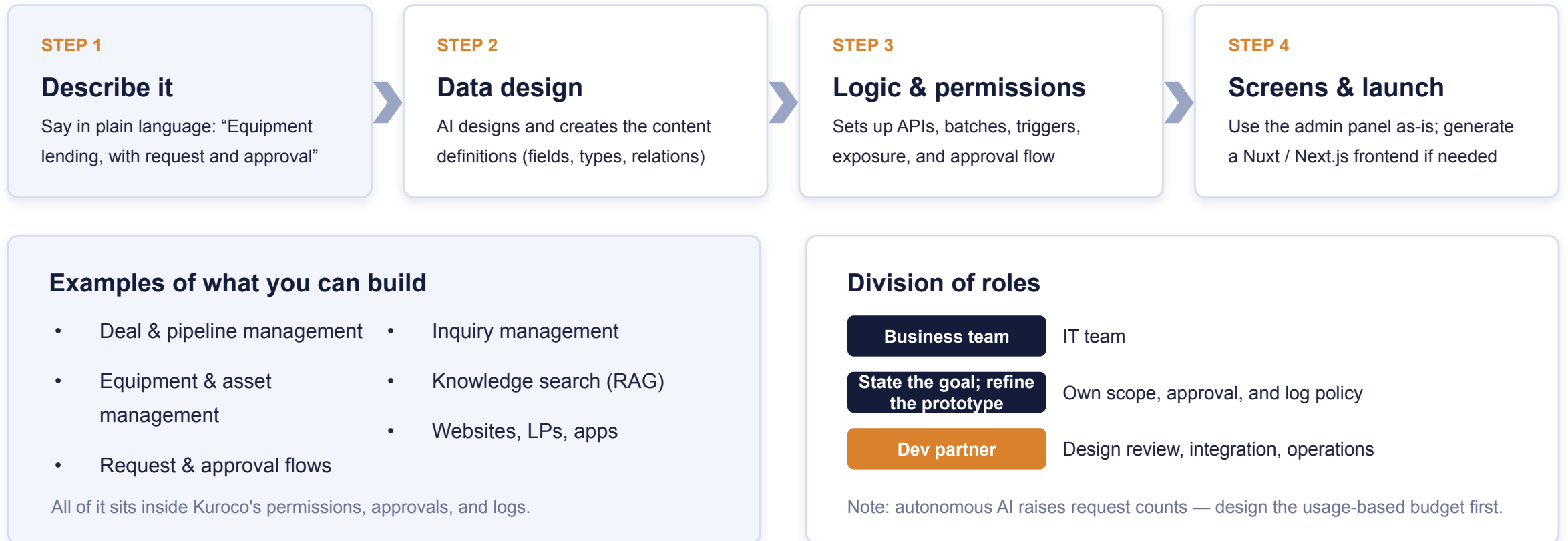
Settings, data, and permissions are visible cross-sectionally in the admin panel. You can supervise what runs without reading code.

4 Auth is never written by AI

OAuth, SAML, 2FA — the layer where mistakes cost most — comes from the platform as standard. Generated code never contains auth logic.

How it works, and what you can build

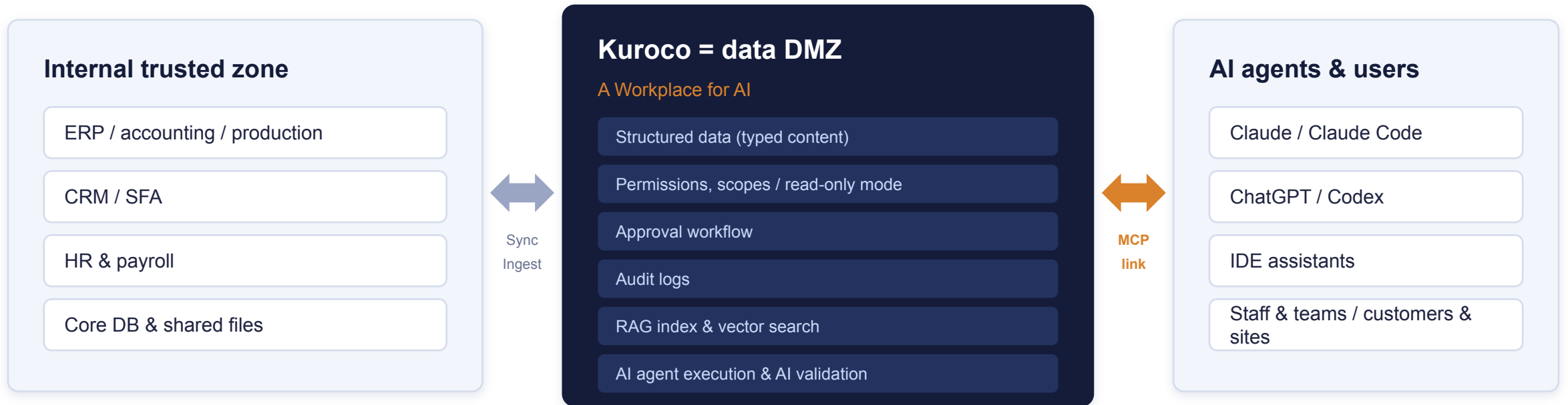
Teams without engineers can start; design review and production stay with the governance side.



The big picture

Kuroco as the data DMZ

Never wire AI to core systems — make Kuroco the single point of contact, and there is one thing to monitor and audit.



AI sees only inside Kuroco; the route is MCP (typed tools) alone, and core systems stay out of reach.

Enterprise-proven security, applied to AI as-is

Not a mechanism invented for AI — AI governance layered onto a platform run by 6,000+ companies.

Platform security (long-standing)

- ISMS (ISO 27001) certified
- WAF, DDoS protection, TLS encryption as standard
- IP restriction, 2FA, SSO via SAML/OAuth
- Automated vulnerability scanning in place
- Developed and operated per IPA secure-website guidelines
- Security checklist available for procurement review

AI-era governance (July 2026—)

- MCP connections use OAuth 2.0 / Bearer tokens
- Module-level scopes and read-only mode (/readonly)
- Every AI operation is logged as audit evidence
- AI validation is fail-closed (no verdict, no pass)
- Human approval built in for critical steps

AI-driven security checks, too

Ask Claude Code (with Kuroco Skills loaded) to inspect settings — auth methods, CORS, exposure, permissions. The AI that builds and the AI that audits share the same platform knowledge.

Sources: Kuroco official site, docs, and public materials (as of 2026). Confirm current certification and scan coverage in the latest official information.

Production operations support, built in

Not just build-and-done: the platform ships with what you need to keep it running.

Environment separation

Run production, staging, and development apart — and confine AI to a sandboxed environment when validating.

Approval, permissions, logs

Approval workflow, fine-grained permissions, and operation logs as standard — trace who (or which AI) did what.

Backup & monitoring

Periodic backups; service status visible on a public status page, with incident history.

Automated deployment

CI/CD via GitHub integration (KurocoFront): no manual pushes to production, fewer update accidents.

Support

Docs, Slack community, and a contact desk — plus optional paid technical support billed per 30 days.

Operating in the AI era

Admin API / Admin MCP requests are billable.
Decide read-only scoping and rate design up front.

Two delivery models — from free tier to dedicated

No per-seat pricing: license cost doesn't jump as business teams and AI agents multiply.

Public SaaS — pure usage-based

- No setup fees — up to \$12 of usage free every month
- Fee = sum of API requests, transfer, storage, email, AI
- No per-user or per-account charges
- Fixed-price plans available; discounts above \$800/mo

Private SaaS — from ¥473,000/month (JPY)

Dedicated environment on ISMAP-compliant cloud (AWS); infrastructure provided by us or prepared by you.

Scale reference (published examples)

Trials & small internal APIs	Within the \$12 free tier
------------------------------	----------------------------------

Active membership site, 1,000 users	≈ \$180 / month
-------------------------------------	------------------------

Media site, 1M page views/month	≈ \$275 / month
---------------------------------	------------------------

Sample plans from the official pricing page; actuals vary with assumptions.

Budgeting for AI: autonomous runs increase request counts

Measure real request volume in a free-tier PoC before estimating. Pair read-only scopes with rate design and costs stay predictable.

Figures from the official pricing page (<https://kuroco.app/pricing/>, as of 2026). Private SaaS pricing is quoted in JPY.

PART A

For operating companies

From “having it built” to “building it yourselves.”

- Can we keep improving operations without engineers?
- Can in-house building coexist with governance (no shadow IT)?
- Where do we start if we want to start small?

This chapter's answer

Business teams build in plain language; IT holds governance. One platform makes that division of labor work.

Six reasons to start AI-driven development on Kuroco

1

Use your existing AI stack as-is

Just connect from the ChatGPT, Claude, or Codex you already use — no tool migration, no company-wide sign-off to wait for.

2

Radically less code to write

Auth, permissions, audit logs, approvals, RAG — none hand-built. Even the app itself is built through dialogue. The cost structure changes.

3

Business teams use it — and build it

Plain language is enough to update and to build. No admin-panel training, no waiting for engineers.

4

Easier internal approval

Japan-built, offered since 2021, 6,000+ companies including finance, manufacturing, and public sector — solid review material.

5

Avoids lock-in

Built on open-standard MCP; works with both Anthropic and OpenAI stacks — no waiting for the company-wide AI decision.

6

Start small, then scale

Begin as a free-tier PoC on usage-based pricing, then expand by department and company-wide as results come in.

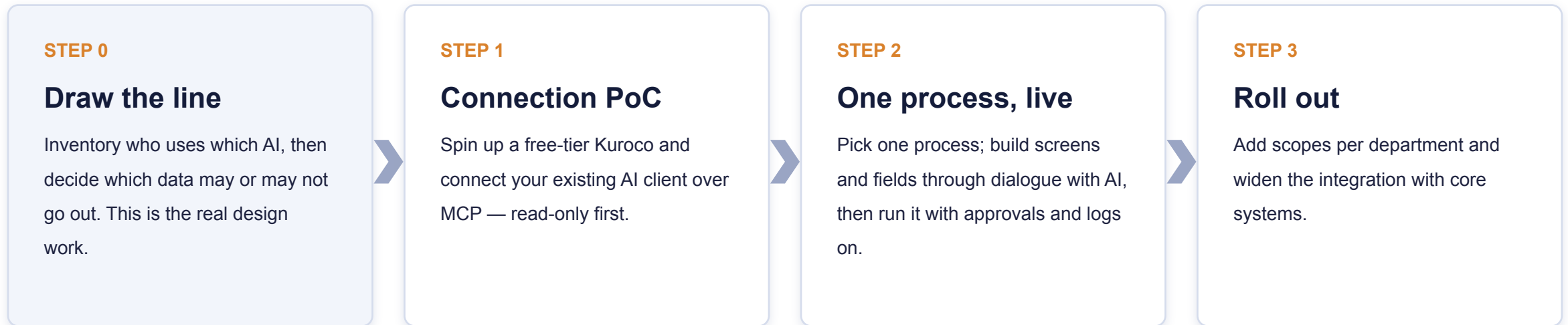
Four easy entry points

Scenario	Typical pain	Where Kuroco helps
Site renewal + AI migration	Auditing and migrating legacy content takes people and time	AI extracts, structures, and migrates content from the old site, compressing effort and cost
Automated inquiry handling	Not enough people to answer the inquiry volume	An AI agent parses each form, consults internal knowledge via RAG, and drafts or sends the reply
First-pass screening of requests	Approvals pile up with individual approvers; judgments vary	Delegate the first pass to AI, with reasoning logged and auditable
In-house app building support	Deal, asset, and request workflows still live in Excel and email	AI-driven build via Claude Code dialogue; AI prepares data design and workflow while IT focuses on review and operations

Everything runs on the same platform as classic headless-CMS use (websites, apps) — wider coverage without another platform.

Start small, and scale with the governance

Don't design the whole company first. Get logs and approvals working in one process, then repeat the pattern.



STEP 1 uses your existing AI environment, so getting started is light

Companies already using ChatGPT, Claude, Claude Code, or Codex connect to Kuroco as-is — no tool swap, no waiting for a rollout decision.

PART B

For development companies

Compete on governance and design, not build volume.

- When AI absorbs implementation, what do we sell to stay chosen?
- How do we grow site-build deals into AI business-platform deals?
- Is AI-driven in-housing a threat — or an opportunity?

This chapter's answer

Engineers move from writing code to directing and verifying AI. The weight shifts to governance design, review, integration, and operations — where recurring revenue lives.

Six benefits of proposing Kuroco to clients

1 Bigger deals to propose

Site-build and CMS-update projects become AI business-platform projects; budgets move from one-off spend to DX investment.

2 Less to build

Auth, permissions, audit logs, approval workflow, RAG — none hand-built. Lower effort than scratch means sharper quotes.

3 The in-housing-support position

Precisely because clients can build, you stay engaged long-term on requirements, governance design, and review.

4 Easier client approvals

Japan-built, offered since 2021, 6,000+ companies including finance, manufacturing, and public sector — strong review material.

5 No waiting on the client's AI selection

Open-standard MCP at the core: runs on both Anthropic and OpenAI stacks, avoiding after-the-AI-policy delays.

6 Recurring revenue in operations

From free-tier PoC to department and company-wide rollout: added scopes, integrations, and cost optimization keep orders coming.

Deal patterns, and where the revenue lives

AI-driven development thins implementation revenue — and turns what AI can't do into the product.

Deal pattern	Client situation	Where the revenue lives
Site renewal + AI migration	Content audit and migration take manual effort	Migration design and verification; AI-driven extraction compresses effort — turn the difference into proposal power
AI business platform build	Wants AI, but can't ship without governance	Governance design (scope, approval, audit policy) and core-system integration — the part AI can't be trusted with
In-housing support	Business teams want to build, but shadow IT is the fear	Design review, naming and data-design standards, training — easy to package as a fixed-fee engagement
Operations & growth	Autonomous AI makes request counts and costs hard to read	Monitoring, rate design, cost optimization, added scopes — managing usage-based billing becomes the product itself

AI-driven development doesn't shrink the work: weight shifts from building to design, governance, and operations, and one-off fees become recurring revenue.

Four objections you'll hear — and the answers

Q Why not just add MCP to our existing systems?

Then auth, permissions, scopes, audit logs, and approvals must be built into the core from scratch — and failures land there too. A separate boundary is faster and contains the blast radius.

Q Why not Salesforce or Contentful?

Same direction — a strong option if that platform is your baseline. Without it, or when data and control must stay in-house, a domestic independent platform is the practical answer.

Q What if AI makes a wrong call?

Design for it: read-only mode, module-level scopes, fail-closed AI validation, human approval on critical steps, full operation logs. Close every path an error could take.

Q What if we can't predict the cost?

Usage-based by default with fixed-fee contracts available. Measure in the free tier before deciding — and for a fixed budget there's the dedicated Private SaaS (from ¥473,000/mo).

Summary

- 01** AI became a worker — but can't be trusted like an employee.
- 02** So place a “data DMZ” between AI and your core systems.
- 03** The vessel: the evolved headless CMS — Kuroco.
- 04** And the apps on top are built through dialogue alone.

Next action: connect your AI stack to Kuroco and build one small app by dialogue (free tier)

Load one process's data in read-only mode and let AI query it. Once it works, stand up one small app — equipment lending or a request form — through AI-driven development.

Pricing: Public SaaS is pure usage-based (fixed-price optional; up to \$12/month of usage free). Dedicated Private SaaS from ¥473,000/month. / Support: docs, Slack community, technical support desk.

Start by trying everything, as-is

Free trial

- Not a limited demo — the full feature set, as-is
- Start with just an account signup; no setup fees — up to \$12 worth of usage per month free
- MCP connection from your existing AI clients (Claude / ChatGPT etc.) works in the trial too

https://kuroco.app/ja/free_trial/

Contact

Diverta Inc. — Kuroco team

kuroco@diverta.co.jp

Contacts: Yoshio / Kitano

From trial setup and MCP connection to AI-driven development and cost estimates — we take questions from the earliest stage. Demo requests welcome.