



AI Transformation Blueprint: Readiness, Building Blocks & Deployment

2026-09-17 · Pavel Kolář

Draft — Version 0.1 · Contact: pavel_kolar@logicstream.eu

Table of Contents

[Introduction](#)

1. [Prerequisites for a Successful Transformation](#)
2. [Organizational and Process Readiness](#)
3. [Building Blocks of the AI Platform](#)
4. [Deployment Options by Use Case](#)
5. [EU Regulatory Compliance \(AI Act & GDPR\)](#)
6. [Security: Threats & Countermeasures](#)
7. [Business Value](#)
8. [Recommended Roadmap](#)

[Appendix: Dictionary of Terms](#)

Introduction

A successful AI transformation happens in three sequential stages, not a single model deployment:

- **Technical readiness** — clean data, documented processes, contained technical debt
- **Organizational and process readiness** — governance, roles, and risk appetite adjusted for autonomous systems
- **Architecture choice** — matched to each use case, chosen only once the first two stages are in place

Most pilots stall in production because the first two stages are skipped and teams go straight to "building agents."

This blueprint sets out, in order: the non-negotiable prerequisites for starting at all (Section 1); how a mid-size services company and a mid-size manufacturer must adjust their operating model and processes to safely absorb autonomous systems (Section 2); the building blocks of a production-grade agentic architecture — foundation, extensibility, orchestration, context isolation, governance, and enterprise RAG/data sovereignty (Section 3); how to match deployment harnesses and concrete vendor products to specific use cases while protecting maintainability, extensibility, traceability, and vendor independence (Section 4); what the EU AI Act and GDPR require and how the same architecture and process documentation already covers most of it (Section 5); the agent-specific security threats this architecture has to defend against and the countermeasures already built into it (Section 6); the business case — cost efficiency with proven savings ranges, error elimination, and the secondary benefits that in practice often outweigh the primary one (Section 7); and a phased roadmap for sequencing the whole transformation (Section 8).

This blueprint sets out a CTO/CIO-level framework for taking a mid-size services company or mid-size manufacturer from ad hoc AI experimentation to production-grade agentic systems: the technical and organizational prerequisites that have to be met first, the operating model and process changes that follow, the architectural building blocks involved, how to match a harness and vendor to a use case without locking the organization in, what EU regulation requires, the security threats that matter most, the business case, and a phased roadmap. It is written to be read in order — each section assumes the one before it — but a reader focused on a specific decision (for example, choosing a harness in Section 4) can jump there directly from the table of contents.

This framework is written from 25+ years of delivering enterprise architecture and process simplification on programmes including group-wide SAP S/4HANA transitions, core banking replacements, BSS modernization, and post-acquisition ERP rollouts. The same disciplined sequence — stabilize the systems and processes first, then choose the agent architecture — is applied here to agentic AI. Hands-on work with Claude Code, analysis of markdown-driven multi-agent systems, and LangGraph-based state-machine agents informs the engineering choices in Sections 3 and 4.

A note on scope. This is a high-level strategy document, not an implementation plan. It sets out the shape of the problem and the sequence of decisions that make an AI transformation durable rather than a one-off pilot; it does not replace a discovery engagement. Applying it to a specific organization requires analysis of that organization's actual systems, data, processes, risk appetite, and regulatory footprint — the prerequisites in Section 1, the operating model in Section 2, and the use-case mapping in Section 4 all need to be re-validated against that organization's reality, not adopted as written. The technology stack referenced throughout — the harnesses, platforms, and vendors named in Section 4 in particular — is illustrative rather than exhaustive: it reflects

a snapshot of a fast-moving market, new entrants and capabilities appear continually, and a procurement decision should be preceded by a current build-vs-buy and vendor evaluation rather than taken directly from this document. The technology recommendations and orchestration patterns are grounded in both large-scale delivery experience and hands-on analysis of production multi-agent codebases — markdown-skill agents and LangGraph state-machine systems.

Key Takeaway — This is a high-level strategic framework, not a finished implementation plan — read it in order, and validate every prerequisite and technology reference against your own organization before acting on it.

1. Prerequisites for a Successful Transformation

Three technical conditions must hold before an agent pilot can become production infrastructure — and they are not equally optional. Technical debt is the hard prerequisite: an organization cannot substitute well-documented processes or data availability for it, because every later step (writing an SOP, opening a data-access layer, adding a governance gate) assumes the underlying systems already expose something stable enough to describe, query, or gate. Model choice is irrelevant until this bar is cleared.

Prerequisite	Why it blocks agentic AI	Minimum bar before a pilot	Typical remediation
Technical debt	Agents call existing systems as tools; brittle integrations, undocumented APIs, and inconsistent data models break tool calls silently and agents fail without a clear error	Core systems expose stable, typed interfaces (API or DB view) for the data/actions a use case needs — not the whole estate	API layer or read-only views over legacy systems; retire or wrap point-to-point integrations before agentizing them — the same pattern used on multi-vendor BSS and core-banking cutovers

Prerequisite	Why it blocks agentic AI	Minimum bar before a pilot	Typical remediation
Well-documented processes	An agent (or the skill/command that drives it) can only encode a procedure that is already explicit; tribal knowledge cannot be “prompted into” a system	The target process exists as a written, current SOP with explicit decision rules and exception paths, not just a flowchart of the happy path	Process mining or structured SME interviews to convert tacit knowledge into a documented, versioned procedure before automating it — the identical artefact later reused for both agent encoding and ISO/IATF or EU AI Act evidence
Data source availability	Retrieval, grounding, and structured tool outputs all depend on data that is queryable, access-controlled, and current; agents amplify whatever the data already is — including its errors	Target data exists in a system of record with an access-controlled query path (API, warehouse, or document store with metadata), not only in email, spreadsheets, or paper	Data cataloguing, access-control mapping, and a minimum-viable retrieval index scoped to the use case, not an enterprise-wide data lake first

For a mid-size services company, the binding constraint is usually documentation: expertise lives in senior staff's heads, not in SOPs, so the first investment is often knowledge capture rather than IT. For a mid-size manufacturer, the binding constraint is more often data source availability: shop-floor, MES, and quality data sit in siloed, poorly integrated systems (or on paper), so a data-access layer over OT/IT systems has to come before any agent that touches production data.

These three gates are non-negotiable because, on past large-scale transformations in regulated environments — banking, energy, telecom — they were consistently the difference between a pilot that reached production and one that didn't.

Key Takeaway — Technical debt, undocumented processes, and unavailable data are not obstacles to work around — they are the prerequisite gate. No agentic AI initiative should move past a pilot until all three are cleared.

2. Organizational and Process Readiness

Once the technical prerequisites hold, the operating model has to change too: an autonomous agent needs an owner, a risk gate, and an escalation path in the same way a new employee needs a manager — and most mid-size organizations have none of these roles defined yet.

Dimension	What changes with agentic AI	Mid-size services company	Mid-size manufacturer
Governance & ownership	Someone has to own each agent's scope, permissions, and behavior the way a system owner owns an application	A cross-functional AI governance group (IT, delivery leads, legal) reviewing every use case before production, not a central "AI team" building everything	Governance shared between plant/quality leadership and IT/OT security — an agent touching production data or equipment needs sign-off from both
Roles	New accountability layer: process owner (defines the SOP), agent owner (maintains prompts/skills/tools), reviewer (approves risky actions) — in practice, the agent owner is often the same senior analyst or process owner who previously owned the requirements-traceability matrix and process-map library on an ERP or core-banking programme	Practice leads become agent owners for their domain (proposals, reporting, support) instead of a single central prompt engineer	Plant/quality engineers become agent owners for their process; IT provides the platform, not the domain judgment
Risk appetite & human-in-the-loop	Autonomy has to be assigned per action, not per agent — "allow" (fully autonomous), "ask" (propose, human approves), or "command-only" (never model-triggered)	Client-facing and billing actions default to "ask"; internal drafting/research defaults to "allow"	Anything touching safety, quality release, or equipment control defaults to "command-only" or "ask"; read-only monitoring/analysis can be "allow"