INTEGRATION MAP

Reusable LPM Knowledge Object · Platform Structure / AI Amplification

Map systems -> data -> workflows -> controls -> AI action boundaries

Purpose

Use this template to map how systems, data, workflows, controls, and AI tools connect across the enterprise. The goal is not a technical diagram alone. The goal is to make every connection visible as an operating-model dependency: what moves, why it moves, who owns it, what can fail, what controls apply, and what AI is allowed to do through it.

Metadata	Value
Object type	LPM Knowledge Object
Primary LPM layers	Platform Structure, Information Ecology, Governance Architecture, AI Amplification
Connected layers	Ownership Map, Decision Architecture, Communication Architecture, Source-of-Truth Map, Data Lineage Map, Workflow Inventory
Primary use	Map how systems, data, workflows, controls, and AI tools connect across the enterprise so dependencies, owners, failure paths, and control boundaries are visible.
Website use	Downloadable template, workshop guide, AI readiness resource, JSON object for Lapemo ingestion, and future guided skill.
Version	1.0
Owner	LPM / Lapemo
Last reviewed	2026-06-24

1. Core principles

Principle	Meaning
Integration is an operating-model dependency	An integration is not just technical plumbing. It moves work, data, decisions, evidence, controls, and AI actions across boundaries.
Every connection needs an owner	The sending system, receiving system, data object, workflow, control, and failure path must have accountable owners.
Manual handoffs are integrations too	Spreadsheets, email, chat, file uploads, and swivel-chair work must be mapped because AI and automation will amplify those weak points.
Data movement changes decision quality	Dashboards, models, agents, and executives inherit the quality, freshness, transformation logic, and lineage of the integration path.
AI access must be bounded by the integration map	AI should not retrieve, update, trigger, or act across systems unless permission, source, lineage, audit, and human review rules are defined.
Failure handling is part of the design	Every critical integration needs observability, retries, exception routing, owner notification, escalation, and recovery evidence.
The map should drive simplification	The goal is not documentation theater. It should identify duplicate flows, brittle handoffs, shadow automation, governance gaps, and consolidation opportunities.

2. Required fields

Field	Definition	Required
Integration ID	Unique identifier for the integration, connector, automation, file transfer, API, event, RPA bot, data pipeline, or manual handoff	Yes
Integration name	Human-readable name used by business and technical teams	Yes
Integration type	API, webhook, event stream, ETL/ELT, file transfer, database replication, RPA, workflow automation, native connector, manual handoff, or AI tool action	Yes
Business purpose	The business capability, workflow, decision, report, control, or AI use case supported by the integration	Yes
Source system	Platform, data product, team, vendor, document, or human process sending the data or work item	Yes
Target system	Platform, workflow, dashboard, AI tool, data store, control system, or team receiving the data or work item	Yes
Source owner	Accountable owner for the source system, source data, or source workflow	Yes
Target owner	Accountable owner for the receiving system, receiving process, or downstream use	Yes
Data owner / steward	Owner of definitions, quality, freshness, and permitted usage for critical data moved through the integration	Required when data moves
Workflow owner	Owner of the business process or operating workflow affected by the integration	Required when workflow-critical
Control owner	Owner of access, audit, retention, approval, segregation, exception, or compliance rule	Required when controlled
Critical objects moved	Records, metrics, decisions, documents, tickets, approvals, model inputs, outputs, prompts, or evidence objects moved across the integration	Yes
Trigger	Schedule, event, user action, AI action, approval, status change, file drop, API call, or manual request that starts the integration	Yes
Frequency / latency	Real-time, near-real-time, hourly, daily, weekly, monthly, ad hoc, batch, or manual	Yes
Transformation logic	Mapping, enrichment, filtering, aggregation, validation, normalization, scoring, or prompt/model processing applied during movement	Yes
Validation checks	Completeness, schema, reconciliation, quality, access, approval, exception, and error-handling checks	Yes
Failure path	How failures are detected, routed, escalated, retried, recovered, and evidenced	Yes
Observability signal	Logs, SLA, freshness monitor, queue depth, error rate, reconciliation report, alert, or manual check	Yes
AI usage boundary	Whether AI can read, summarize, classify, recommend, trigger, update, approve, or act through this integration	Yes
Human review rule	When a human must review AI-triggered movement, exceptions, data updates, or downstream actions	Required when material
Risk tier	Low, moderate, high, critical, regulated, customer-impacting, financial-impacting, or AI-actioning	Yes
Evidence link	Log, ticket, control record, policy, runbook, data lineage reference, source-of-truth record, or decision log link	Yes
Lifecycle state	Discover, pilot, active, monitored, governed, constrained, duplicate, replace, retire, or exception	Yes
Review date	Next review of purpose, ownership, data movement, controls, AI boundary, and failure path	Yes
Version	Object version, owner, last reviewed date, and change history	Yes

3. Integration types

Type	Definition	Example	Common risk
API / service call	Synchronous system-to-system request and response	CRM calls pricing service	Timeouts, version drift, missing owner, weak auth
Webhook / event trigger	System pushes event when something changes	Contract signed triggers onboarding workflow	Silent failures, duplicate events, no retry path

Type	Definition	Example	Common risk
Event stream	Continuous publish/subscribe movement across systems	Kafka events feed data products and alerts	Schema changes, consumer sprawl, weak lineage
ETL / ELT pipeline	Batch or scheduled data extraction, transformation, and loading	ERP data into warehouse and BI model	Stale data, transformation opacity, dashboard trust gaps
File transfer	CSV, flat file, secure transfer, export/import, or vendor feed	Daily HR file to identity system	Manual errors, late files, no validation evidence
Database replication	Data copied or synchronized between stores	Operational DB to analytics replica	Latency, schema drift, ambiguous source of truth
RPA / bot	Automation interacts with a UI when no API exists	Bot moves cases between legacy systems	Brittle selectors, hidden credentials, weak audit trail
Workflow automation	Rules engine or orchestration tool moves work across systems	ServiceNow to Jira to Teams notification	No process owner, no exception routing
Native connector	Vendor-provided connector between platforms	Salesforce to Slack notification	Limited transparency, permission creep
Manual handoff	Human moves data or work through chat, email, spreadsheet, meeting, or upload	Analyst emails weekly report	Invisible dependency, stale evidence, AI hallucination risk
AI tool action	AI retrieves, transforms, summarizes, classifies, updates, or triggers work across systems	Agent updates CRM from meeting notes	No decision rights, source ambiguity, uncontrolled action

4. Integration map stages

Stage	What happens	Evidence	Key fields	Common risk
1. Identify	Catalog the integration, type, business purpose, source, target, trigger, and lifecycle state.	App inventory, architecture diagram, workflow inventory, data catalog	Integration ID, type, source, target	Hidden or undocumented connection
2. Own	Name source owner, target owner, data owner, workflow owner, technical owner, and control owner.	Ownership map, platform map, RACI, support model	Owners and review cadence	Ownerless integration or unclear accountability
3. Trace	Map critical objects, transformation logic, upstream dependencies, downstream consumers, and lineage.	Data lineage, source-of-truth map, pipeline config, runbook	Objects moved, mappings, lineage	Decision or AI output cannot be traced
4. Control	Attach access, security, approval, retention, privacy, audit, reconciliation, and exception rules.	GRC records, IAM policy, audit logs, control evidence	Controls and validation checks	Data moves without approval or evidence
5. Monitor	Define observability, failure detection, SLA, freshness, reconciliation, alerts, and support path.	Logs, SLA reports, monitoring, tickets, alerts	Health signal and failure path	Silent failure or brittle handoff
6. AI-bound	Define what AI can retrieve, summarize, transform, trigger, update, approve, or act on through the integration.	AI policy, model registry, permission model, agent logs	AI boundary and human review rule	AI action without authority or audit
7. Optimize	Decide whether to standardize, govern, automate, consolidate, replace, retire, or accept as an exception.	Roadmap, architecture review, risk register, value case	Lifecycle decision and next action	Integration sprawl and automation debt

5. Integration archetypes

Archetype	Definition	Examples	Governance requirement
Operational integration	Moves work between systems during execution	Case to ticket, lead to opportunity, order to fulfillment	Requires workflow owner, failure path, SLA, and escalation rule
Data integration	Moves data for reporting, analytics, models, or downstream systems	ERP to warehouse, CRM to lakehouse, HRIS to BI	Requires data owner, lineage, transformation logic, freshness, and reconciliation
Control integration	Feeds or enforces access, approvals, risk, policy, audit, or compliance evidence	IAM provisioning, GRC evidence feed, policy exception	Requires control owner, audit trail, evidence, and exception handling
Experience integration	Connects customer, employee, partner, or vendor experiences across tools	Support ticket to customer portal, HR request to knowledge base	Requires experience owner, privacy review, quality checks, and communication rule
Knowledge integration	Moves documents, policies, decisions, templates, playbooks, or knowledge objects	Confluence to AI knowledge base, SharePoint to search index	Requires freshness, source-of-truth, versioning, and AI retrieval boundary
AI integration	Allows AI tools or agents to retrieve, summarize, recommend, update, or act through connected systems	Copilot reads CRM, agent updates Jira, RAG over policy docs	Requires permission boundary, source lineage, human review, logging, and kill switch
Shadow integration	Unapproved or unmanaged movement of	Manual CSV exports, spreadsheet uploads,	Requires risk triage, unmet-need analysis,

Archetype	Definition	Examples	Governance requirement
	data or work	personal automations	and migration plan

6. Template: integration map

Map element	Entry	Guidance
Integration ID	[INT-001]	Unique connector, automation, pipeline, bot, or handoff ID
Integration name	[Name]	Common business and technical name
Integration type	[API / event / ETL / file / RPA / manual / AI action]	Defines technical and governance pattern
Business purpose	[Workflow, decision, report, control, AI use case]	Why the integration exists
Source system	[System / team / vendor / document / human process]	Where data or work starts
Target system	[System / dashboard / workflow / AI tool / team]	Where data or work lands
Source owner	[Name / role]	Owns the source system, source data, or source workflow
Target owner	[Name / role]	Owns the downstream system or process
Critical objects moved	[Records, metrics, tickets, decisions, documents, evidence]	What moves across the connection
Trigger	[Schedule / event / user action / AI action / approval]	What starts the integration
Frequency / latency	[Real-time / daily / batch / ad hoc / manual]	Expected timing and freshness
Transformation logic	[Mapping, enrichment, filtering, scoring, validation]	How the object changes in transit
Validation checks	[Schema, completeness, reconciliation, quality, access]	How correctness is confirmed
Failure path	[Alert, retry, ticket, owner, escalation, recovery]	What happens when it breaks
Observability signal	[Log, SLA, freshness monitor, error rate, queue depth]	How health is monitored
AI usage boundary	[Read / summarize / classify / trigger / update / act / blocked]	Defines what AI can do
Risk tier	[Low / moderate / high / critical / regulated]	Governance intensity
Evidence link	[Runbook, log, control, ticket, lineage, decision log]	Proof and traceability
Lifecycle state	[Discover / active / monitored / governed / replace / retire]	Current operating status
Review date	[Date]	Next ownership, control, and AI boundary review

7. Version for 500+ employee company

Dimension	Recommended pattern
Design intent	Create practical visibility into the integrations that keep the company running before tool sprawl and AI pilots create hidden risk.
Minimum scope	Map the top 25 to 50 integrations across revenue, delivery, finance, people, data, knowledge, and active AI workflows.
Operating pattern	Monthly review of critical integrations, manual handoffs, owner gaps, and AI-connected workflows.
AI focus	Do not let AI tools act through integrations until the source, target, owner, review rule, and audit trail are clear.
Governance need	Name owners, define failure paths, document manual handoffs, and establish source-of-truth relationships.
Red flags	Spreadsheet uploads, founder/operator-held logic, no support owner, daily CSV exports, AI connected to local files, and no failure alerts.

- Start with integrations that support revenue, customer support, delivery, finance, people operations, reporting, and active AI pilots.
- Map manual handoffs before automation decisions are made.
- Name owners and failure paths before adding more connectors.

8. Version for 5,000+ employee company

Dimension	Recommended pattern
Design intent	Move from scattered connector knowledge to domain-level integration governance across platforms, data products, controls, and AI use cases.
Minimum scope	Map Tier 1 integrations, critical data pipelines, workflow automations, cross-functional handoffs, and AI-connected systems.
Operating pattern	Quarterly integration governance by business domain with architecture, platform, data, risk, and transformation owners.
AI focus	Classify integrations by AI usage boundary: retrieve, summarize, recommend, trigger, update, approve, act, or blocked.
Governance need	Connect integration records to platform maps, data lineage maps, source-of-truth maps, decision logs, and control evidence.
Red flags	Duplicate connectors, brittle automations, inconsistent customer or employee data, weak pipeline ownership, and unmanaged vendor integrations.

- Organize integrations by business domain, critical object, platform category, and risk tier.
- Connect integration records to source-of-truth maps, data lineage maps, workflow inventory, and decision logs.
- Use integration health, duplication, control exceptions, and AI boundary gaps to drive architecture decisions.

9. Version for 10,000+ employee company

Dimension	Recommended pattern
Design intent	Create enterprise integration control across business units, shared services, regulated data, external partners, AI agents, and control systems.
Minimum scope	Map critical enterprise integrations, regulated data flows, external data sharing, high-risk automation, model inputs, agent tool-use paths, and control evidence feeds.
Operating pattern	Enterprise integration council, domain integration owners, automated dependency discovery, control evidence packs, and architecture exception governance.
AI focus	AI agents require integration-specific permission boundaries, model risk tiering, source lineage, human review, tool-use logs, and kill-switch rules.
Governance need	Tie integration changes to architecture, security, privacy, procurement, GRC, data governance, model risk, and executive decision rights.
Red flags	Silent API dependency, federated integration sprawl, unmanaged event consumers, vendor black boxes, external data leakage, and AI actions without approval authority.

- Treat the integration map as an enterprise control layer across systems, data domains, workflows, external partners, controls, and AI agents.
- Automate dependency discovery where possible, but keep ownership, risk tiering, and AI action permissions human-approved.
- Connect integration change to architecture, security, privacy, procurement, data governance, model risk, and executive escalation.

10. Scoring logic

Dimension	Score	What good looks like
Ownership clarity	0-5	Source, target, data, workflow, technical, and control owners are named for critical integrations.
Purpose clarity	0-5	The business purpose, supported workflow, decision, data object, or control is explicit.
Lineage clarity	0-5	Source, target, transformation logic, downstream consumers, and evidence links are traceable.
Reliability visibility	0-5	Failure detection, SLA, retry, monitoring, reconciliation, and support path are defined.
Governance strength	0-5	Access, approvals, privacy, retention, audit, exceptions, and controls are attached.
AI boundary clarity	0-5	AI retrieval, trigger, update, approval, and action permissions are explicit and reviewed.

Dimension	Score	What good looks like
Manual handoff exposure	0-5	Manual exports, uploads, spreadsheets, emails, and chat-based movement are visible and governed.
Lifecycle discipline	0-5	Integrations are actively governed as discover, pilot, active, monitored, governed, replace, retire, or exception.

Suggested readiness score: average the eight scores, then classify 0-1.9 as Fragmented, 2.0-3.4 as Mapped, 3.5-4.4 as Governed, and 4.5-5.0 as AI-ready.

11. AI prompts

- Given this integration inventory, classify each connection by integration type, risk tier, lifecycle state, source-of-truth relationship, and AI usage boundary.
- Identify owner gaps, brittle handoffs, duplicate connectors, undocumented manual integrations, stale data paths, weak failure handling, and ungoverned AI actions.
- Score each integration from 0 to 5 across ownership clarity, purpose clarity, lineage clarity, reliability visibility, governance strength, AI boundary clarity, manual handoff exposure, and lifecycle discipline.
- Generate an integration dependency map showing source systems, target systems, critical objects moved, transformation logic, downstream consumers, controls, AI consumers, and failure paths.
- Recommend whether each integration should be documented, monitored, governed harder, automated, replaced, consolidated, retired, or accepted as an exception.
- Create a Lapemo ingestion plan that converts this integration map into integration objects, platform objects, data lineage objects, control objects, workflow objects, evidence objects, and AI boundary rules.

12. Validation rules

- Every critical integration must have a source owner, target owner, and technical owner.
- Every integration moving critical data must identify the data owner, source-of-truth relationship, transformation logic, freshness expectation, and validation checks.
- Every workflow-critical integration must have a workflow owner, failure path, escalation path, and recovery rule.
- Every integration used by AI must have an approved AI usage boundary, human review rule, source lineage, and audit trail.
- No AI agent should trigger, update, approve, or act through an integration unless decision rights and control rules are defined.
- Manual handoffs must be mapped and scored instead of ignored because they often represent the highest automation and AI risk.
- A connection cannot be classified as governed if failure detection, owner notification, evidence, and review cadence are missing.
- Status must never be encoded only by color; use labels such as active, monitored, governed, constrained, duplicate, replace, retire, exception, blocked, or disputed.

13. Lapemo ingestion mapping

Lapemo object	Fields / entities	Use
Knowledge object	Integration Map	Canonical reusable artifact for system dependency control and AI-safe connectivity.
Integration object	Integration ID, type, source, target, trigger, lifecycle, risk tier	Creates the enterprise integration inventory and dependency layer.
Platform object	Source system, target system, upstream and downstream dependencies	Connects integrations to platforms and systems of record.
Ownership object	Source owner, target owner, data owner, workflow owner, technical owner, control owner	Connects every connection to accountable humans.
Data lineage object	Critical objects moved, transformation logic, freshness, quality checks, downstream consumers	Shows how data flows into decisions, dashboards, and AI.
Workflow object	Trigger, process path, handoffs, decision points, exception routing	Links integrations to recurring work and automation candidates.
Governance object	Access, approvals, retention, privacy, audit, exception, evidence link	Connects integrations to controls and auditability.
AI object	AI usage boundary, permissions, human review rule, agent logs, kill switch	Defines what AI can safely do through connected systems.

14. Reusable knowledge-object model

This artifact should exist in four synchronized forms: a human-readable guide, a downloadable template, a machine-readable JSON object, and a guided Lapemo skill. The knowledge object should be versioned, reviewed, scored, and connected to system data over time. It should not auto-update silently. Lapemo should flag missing owners, brittle integrations, undocumented manual handoffs, stale data movement, weak failure paths, duplicate connectors, ungoverned AI access, and high-risk dependency changes for human approval.