

LPM KNOWLEDGE OBJECT

Reusable Decision Rights Matrix

Three versions for 500+, 5,000+, and 10,000+ employee companies

A practical template for making decision authority visible before AI scales work across people, systems, governance, and agents.

Object field	Value
Object type	LPM Knowledge Object
LPM layer	Decision Architecture
Primary use	Define authority, input, approval, evidence, escalation, and AI boundaries
Website use	PDF, DOCX, Markdown, JSON, CSV, and future Lapemo guided skill
Version	1.0
Last reviewed	2026-06-23
Owner	LPM / Lapemo

Core claim: AI does not remove decision rights. It exposes them. This matrix defines where judgment lives before AI amplifies execution.

1. Knowledge object model

How the Decision Rights Matrix becomes reusable, ingestible, and updatable

A Decision Rights Matrix should not live only as a static download. It should be stored as a reusable knowledge object with human-readable guidance and machine-readable structure.

Reusable layer	Purpose	Website / Lapemo output
Human guide	Explains how to define decision authority and escalation paths	PDF guide, executive briefing, workshop instructions
Template artifact	Gives teams a practical matrix to complete before onboarding	Word doc, worksheet, CSV import template
Machine-readable schema	Stores fields, scoring, validation, mappings, and version metadata	JSON knowledge object and ingestion source
Guided skill	Uses AI to classify, draft, validate, score, and render the matrix	Lapemo in-app workflow or website assistant

Leadership principle: Do not let AI update decision rights silently. Let AI propose updates with evidence, then require a human owner to approve and publish a new version.

2. Canonical fields and scoring

The base structure every profile uses

Field	What it captures	Required?
Decision domain	Area of the business the decision affects	Yes
Decision type	Strategic, operating, product, technology, data, risk, AI, people, or exception	Yes
Decision owner	Accountable owner of the decision outcome	Yes
Decision maker	Person or role with authority to decide	Yes
Approver	Role required when threshold is met	Conditional
Consulted roles	Roles that must provide input before decision	Yes
Evidence required	Data, policy, metric, architecture, customer, risk, or financial evidence	Yes
Decision SLA	Expected cycle time for the decision	Yes
Escalation trigger	Condition that moves decision upward or across governance	Yes
AI involvement	None, assisted, recommended, executed with approval, or executed with monitoring	Yes
System of record	Where decision, evidence, and rationale are stored	Yes
Review cadence	Monthly, quarterly, semiannual, annual, or event-driven	Yes

Score	Meaning
0	No named owner or authority path
1	Owner exists, but authority is informal or personality-driven
2	Decision path exists, but evidence, approval, or escalation rules are incomplete
3	Decision path is documented, evidence-based, and connected to operating workflows
4	Decision path is governed, measured, updated, and ready for AI-assisted execution

3. Version A - 500+ employee company

From founder-led decisions to scalable operating authority

A 500+ person company usually has speed, strong relationships, and founder or executive pattern recognition. The risk is that authority is still informal. AI makes this worse if employees and agents act on recommendations without clear ownership.

Decision area	Who decides	Required input	Evidence required	Escalation / AI boundary
Strategic priorities	CEO or executive team	Function leaders, finance, product or operations lead	Annual goals, capacity, financial impact, customer impact	Escalate when tradeoffs affect more than one function. AI summarizes options, not strategy.
Product and service priorities	Product, business, or operations owner	Sales, customer success, engineering, support	Customer signals, revenue impact, delivery capacity	Escalate when priority changes shift roadmap, funding, or customer commitments.
Process changes	Functional owner	Process participants, technology owner, compliance if needed	Current-state flow, pain points, expected improvement	Escalate when change crosses functions or impacts controls. AI may draft process maps.
Technology changes	Technology lead or platform owner	Security, architecture, impacted function owner	Architecture note, cost, risk, integration impact	Escalate when new tools create data, security, or workflow dependencies.
Data definitions and access	Data owner or analytics lead	Business owner, security, privacy, technology	Definition, source, access purpose, retention need	Escalate when data is sensitive or used by AI models.
AI use cases	Business owner with tech and risk review	Process owner, data owner, security, legal if needed	Use case, users, data, expected action, risk tier	Human approval required for customer, employee, financial, or regulated decisions.
Exceptions	Functional owner	Impacted owner, finance or risk as applicable	Exception reason, impact, duration, mitigation	Escalate if exception repeats or becomes operating norm.

500+ design principle: keep it lightweight. The goal is to remove founder bottlenecks without creating bureaucracy.

4. 500+ implementation notes

Operating pattern, failure modes, and first workshop focus

- Start with the top 20 recurring decisions that slow execution.
- Assign one accountable owner for each decision area.
- Set clear escalation triggers instead of adding standing committees.
- Store decision rationale in one accessible system, such as Confluence, Notion, SharePoint, Jira, or the operating platform.
- Review monthly until decision latency stabilizes.

Failure mode	Signal	Fix
Founder or executive bottleneck	Teams wait for informal approval	Move repeatable decisions to functional owners with thresholds
Consensus masquerading as ownership	Everyone gives input, no one decides	Separate consulted roles from decision maker
Tool sprawl creates decision drift	Decisions live in Slack, email, and docs	Pick a system of record and require rationale
AI pilots launch without authority	AI use cases are local experiments	Require decision owner, data owner, and risk boundary before pilot

5. Version B - 5,000+ employee company

From functional maturity to cross-functional authority

A 5,000+ employee company usually has process maturity, but decision rights break across departments, portfolios, systems, and governance paths. The problem is rarely absence of process. It is unclear authority across functions.

Decision area	Who decides	Required input	Evidence required	Escalation / AI boundary
Enterprise priorities	Executive committee or portfolio governance	BU leaders, finance, technology, risk	Portfolio tradeoffs, capital plan, strategic outcomes	Escalate when funding, headcount, risk appetite, or customer commitments change.
Portfolio prioritization	Portfolio owner or product governance	Product, engineering, operations, finance	Demand, capacity, value score, dependency map	Escalate when conflicts span portfolios or shared platforms.
Operating model changes	Transformation or operations leader with business owner	HR, finance, technology, risk, impacted functions	Role impact, process impact, control impact	Escalate when changes affect spans, roles, incentives, or governance.
Shared platform changes	Platform owner or architecture board	Security, architecture, data, impacted product owners	Architecture decision record, dependency map, cost, risk	Escalate when change affects enterprise standards or critical workflows.
Data definitions and access	Data domain owner	Data governance, security, privacy, business owner	Data contract, classification, lineage, usage purpose	Escalate when data feeds AI, regulatory reporting, or customer-impacting decisions.
AI use cases	AI product owner with business accountable owner	Risk, legal, data, security, model owner, process owner	Use case tier, model card, data lineage, human review path	Human approval required for high-impact decisions and regulated processes.
Policy exceptions	Policy owner or risk governance	Business owner, compliance, legal, audit if needed	Exception rationale, risk acceptance, mitigation, expiration	Escalate if exception repeats, affects customers, or creates control weakness.
Cross-functional dependencies	Program, product, or operating owner	Dependent teams, platform owners, finance	Dependency chain, impact, date risk, decision needed	Escalate when dependency blocks committed outcome or AI workflow launch.

5,000+ design principle: define the minimum viable governance path. Reduce coordination drag, do not create another approval maze.

6. 5,000+ implementation notes

Decision tiers and governance hygiene

- Create decision tiers by risk and impact.
- Separate enterprise standards from local execution authority.
- Connect each decision area to a system of record, not a meeting series.
- Add review rules for AI, data, privacy, security, and customer-impacting decisions.
- Track decision latency, reversals, escalations, and unresolved ownership gaps.

Failure mode	Signal	Fix
Functional silos	Each function has its own approval path	Establish cross-functional decision domains and escalation thresholds
Governance theater	Committees meet but authority is unclear	Name the decision maker and make committees advisory unless approval is explicit
Evidence inconsistency	Decisions rely on different metrics	Standardize evidence requirements by decision type
AI governance is separate from operating governance	AI review happens after teams build	Require AI boundary during intake and design, not post-build review

7. Version C - 10,000+ employee company

Federated decision rights for enterprise scale

A 10,000+ employee company needs federated decision rights. Centralized governance alone becomes too slow. Local autonomy alone becomes too risky. The goal is a control layer that defines enterprise standards, local authority, and escalation rules by risk tier.

Decision area	Who decides	Required input	Evidence required	Escalation / AI boundary
Enterprise strategy and capital	CEO, executive committee, board where applicable	Segment presidents, CFO, CIO or CTO, risk, legal	Strategic plan, capital allocation, market and risk analysis	Escalate for board-level risk, material capital, regulatory exposure, or reputation impact.
Global operating model standards	Enterprise operating model owner	HR, finance, technology, legal, regional leaders	Role taxonomy, process architecture, control map	Escalate when local variation conflicts with global control or scale requirements.
Business unit execution	BU president or segment operating owner	Product, operations, finance, technology, risk	BU scorecard, capacity, customer and regulatory impact	Local authority within enterprise guardrails. Escalate when thresholds are exceeded.
Enterprise platform standards	Enterprise platform council or architecture authority	Platform owners, security, data, regions, procurement	Architecture decision record, dependency map, resilience and cost analysis	Escalate when introducing enterprise-wide platforms or high-risk integrations.
Data and information governance	Chief data office and domain data owners	Privacy, security, legal, analytics, business domains	Data classification, lineage, quality, access, usage rights	Escalate when data powers AI, customer decisions, regulatory reporting, or cross-border transfer.
AI systems and agentic workflows	AI governance body plus accountable business owner	Model risk, data owner, security, legal, compliance, process owner	Model card, agent permissions, audit trail, human-in-loop rule, kill switch	AI cannot execute high-impact decisions without authority, monitoring, and rollback controls.
Risk acceptance and exceptions	Risk owner, control owner, or executive risk committee	Business owner, legal, compliance, audit	Risk rating, mitigation, expiration, residual risk owner	Escalate when exceptions are repeated, systemic, customer-impacting, or regulatory.
M&A integration decisions	Integration executive and functional owners	Finance, HR, technology, risk, legal, operations	Integration thesis, target-state architecture, operating model gap map	Escalate when acquired processes, systems, or AI tools conflict with enterprise standards.

10,000+ design principle: centralize standards and decentralize execution within thresholds. The matrix must operate like a control plane, not a policy binder.

8. 10,000+ implementation notes

Federated governance and enterprise control

- Define global decision standards by domain.
- Delegate authority by business unit, region, product line, and risk tier.
- Require machine-readable decision records for high-impact decisions.
- Connect decision rights to control evidence, audit trails, and AI operating boundaries.
- Review decision rights after reorganizations, M&A, platform migrations, regulatory changes, and AI workflow launches.

Failure mode	Signal	Fix
Central governance bottleneck	Local teams wait weeks for routine decisions	Delegate routine authority with thresholds and audit evidence
Local variation becomes fragmentation	Different regions define the same decision differently	Establish enterprise decision taxonomy and local override rules
AI agents cross authority boundaries	Agents act across systems without clear owner	Bind agent permissions to decision domain, system, risk tier, and owner
Audit trail is not decision trail	Controls exist but rationale is missing	Require evidence, rationale, and approval path as part of the record

9. AI decision boundaries

How decision rights should govern AI-assisted and agentic workflows

AI involvement level	Description	Decision rights requirement
None	No AI is involved	Standard decision rights apply
AI-assisted	AI drafts, summarizes, or analyzes	Human decision maker remains fully accountable
AI-recommended	AI recommends an option or action	Decision owner must approve criteria and evidence
AI-executed with approval	AI executes after human approval	Human approval, audit trail, rollback path, and exception owner required
AI-executed with monitoring	AI executes within guardrails	Explicit authority, monitoring, threshold rules, kill switch, and review cadence required

- No AI workflow should be approved without a named business decision owner.
- No AI recommendation should be used without evidence requirements and escalation rules.
- No agentic workflow should cross systems unless platform ownership and data ownership are clear.
- No high-impact decision should be fully automated without governance, monitoring, and human override.
- Every AI-enabled decision path must have a review cadence and a system of record.

10. Operating workflow

How to run the Decision Rights Matrix before or during Lapemo onboarding

Step	Activity	Output
1	Select company profile	500+, 5,000+, or 10,000+ version
2	List recurring decisions	Decision inventory
3	Classify decision type and impact	Decision taxonomy and risk tier
4	Assign owner, maker, approver, consulted, and informed	Draft decision rights matrix
5	Define evidence and system of record	Decision record standard
6	Define escalation and AI boundary	Governance path and AI guardrail
7	Score readiness	Decision readiness score and gap list
8	Publish version	Website artifact, internal guide, or Lapemo ingestion object

Review triggers

- Reorganization or new operating model.
- New executive, function owner, product owner, or platform owner.
- Major AI workflow, agent, or automation introduced.
- New system of record, data catalog, workflow platform, or governance platform.
- Repeated escalations or unresolved cross-functional conflicts.
- Policy, regulatory, security, or audit requirement changes.
- M&A integration, divestiture, or major vendor change.

11. Website and skill model

How this becomes more than a static download

Asset	Use
PDF guide	Executive education and briefing
Word template	Workshop and consulting artifact
Markdown page	Website content and documentation source
JSON knowledge object	Lapemo ingestion and future skill execution
CSV template	Bulk decision inventory import
In-app workflow	Guided decision rights setup in Lapemo

Future Lapemo skill actions

- Classify decision areas by domain, risk, AI involvement, and company profile.
 - Draft the matrix from interview notes, org data, system data, policies, and existing artifacts.
 - Score clarity, authority, evidence, latency, escalation, and AI boundary readiness.
 - Validate missing owners, vague approvers, stale reviews, and conflicting authority paths.
 - Render the object into DOCX, PDF, CSV, JSON, website copy, or in-app module.
- Automatic update pattern: systems generate change signals, AI proposes updates with evidence, a human owner approves, and a new version is published with a change log.

12. Validation and mapping rules

How the knowledge object stays trustworthy and ingestible

Validation rule	Pass condition
Named decision owner	Every decision area has one accountable owner
Authority clarity	Decision maker and approver are not confused
Evidence standard	Required evidence is listed for every decision type
Escalation trigger	Escalation rule is explicit and measurable
AI boundary	AI involvement level and human approval rule are defined
System of record	Decision rationale has a place to live
Review cadence	Review frequency or event trigger is listed
Version control	Version, owner, date, and change rationale are captured

Matrix field	Lapemo mapping
Decision domain	Decision Architecture module
Decision owner	Ownership Map / Identity & Incentives layer
Evidence required	Information Ecology and Data layer
System of record	Platform Structure layer
Escalation trigger	Governance Architecture layer
AI involvement	AI Amplification layer
Decision SLA	Communication Architecture and operating cadence
Score	Readiness, risk, and lineage dashboards

13. Version control

Change log and next reusable artifacts

Version	Date	Owner	Change
1.0	2026-06-23	LPM / Lapemo	Initial reusable Decision Rights Matrix knowledge object

Recommended next artifacts

- AI Use Case Intake Map
- Operating Model Dependency Map
- Governance Escalation Path
- System of Record Control Map
- Role and Accountability Ledger

Publishing recommendation: host this as a library object on the LPM website with four download options: PDF, DOCX, Markdown, and JSON. In Lapemo, treat the JSON as the source object and render all human-facing formats from that source.