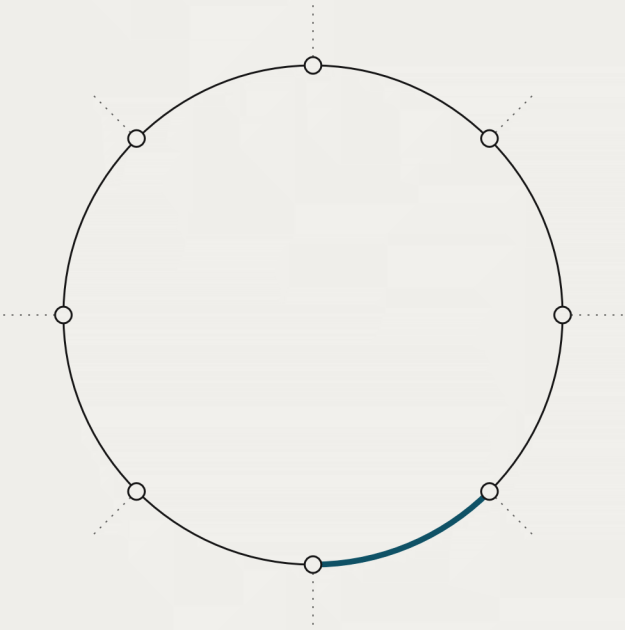
HUMAN ACCOUNTABLE FOR THE LOOP

How To Keep Ownership, Authority And Control When AI Acts



RYAN McDONOUGH

HUMAN ACCOUNTABLE FOR THE LOOP

FREE SAMPLE

How to keep ownership, authority and control when AI acts

Ryan McDonough

theloop.legal

Digital PDF sample edition · 9 August 2026

The legal and regulatory examples in this book illustrate governance questions. They are not legal advice. Liability and professional obligations depend on the jurisdiction, facts and applicable law.

Companion worksheets and interactive tools are available at theloop.legal. They are practical aids rather than certifications or permission to deploy.

Contents

About this sample

This free sample contains the opening two chapters of *Human Accountable for the Loop*. Chapter 1 establishes the accountability problem. Chapter 2 introduces the complete HAL framework and the eight questions it asks of an AI-enabled workflow.

The full edition develops each discipline through cases, counterpoints and practical control patterns, then brings all eight parts together in a worked contracting example. It also includes the HAL Workbook for teams applying the framework to a live workflow.

A note to the reader

This book is about the governance of AI-enabled workflows: systems in which models, data, software tools and people combine to produce decisions or actions.

I wrote it for leaders, lawyers, risk professionals, product teams, engineers and operational owners who need to decide when automation may act and how the organisation remains answerable when it does.

The legal and regulatory examples illustrate governance questions; they are not legal advice, and liability will depend on the jurisdiction, the facts and the applicable law. Case descriptions distinguish allegations, findings and settlements wherever that distinction is material.

Technical controls are presented as risk-based patterns rather than universal requirements. The right design depends on the workflow's consequence, speed, scale, reversibility and operating environment.

I call the framework **Human Accountable for the Loop**, or **HAL**. It does not require a person to review every automated action. It requires the workflow to remain owned, bounded, challengeable, reconstructable, monitored, periodically reconsidered and connected to people and organisations able to control its consequences.

I distinguish three Loop patterns: **Review** or Human-in-the-Loop, **Monitor** or Human-on-the-Loop, and **Own** or Human-Accountable-for-the-Loop. My detailed focus is the third pattern: workflows that act at a speed or scale where individual review is absent or insufficient. The same eight disciplines can also test consequential Review and Monitor arrangements, because a review step or dashboard does not by itself establish accountability.

CHAPTER 1

The Accountability Illusion

For years, branch accounts produced by the Post Office's Horizon system showed money missing. The people running those branches were expected to make the accounts balance. When they could not explain an apparent shortfall, the system's number was treated as evidence about their conduct.

The asymmetry was extraordinary: the Post Office and its technology supplier possessed the system, support records and technical knowledge, while a subpostmaster facing an allegation possessed only a terminal, a contract and a figure they could not independently verify. The accounting system appeared authoritative; the person was visible and available to blame.

In April 2021, the Court of Appeal quashed the convictions of thirty-nine former subpostmasters. Its judgment concerned cases in which Horizon data had been essential to the prosecution and there was no independent evidence of an actual loss. The court recorded findings from earlier civil litigation that bugs, errors and defects created a material risk that an apparent shortfall did not represent missing cash or stock. It concluded that the Post Office knew there were serious questions about Horizon's reliability, yet had consistently asserted that the system was robust and had failed adequately to investigate or disclose relevant problems.[1]

The Horizon scandal isn't an artificial intelligence case, which is precisely why it belongs at the start of this book.

AI didn't invent the practice of assigning responsibility to people who lack access to the system said to justify it. It also didn't invent institutional faith in automated evidence. It inherits both tendencies, then adds systems whose behaviour may be probabilistic, whose components change rapidly and whose control is distributed across providers, deployers and professional users.

The Horizon cases show the underlying accountability error without the distraction of debating whether a system was truly intelligent. Humans were everywhere: entering transactions, balancing accounts, operating support services, investigating discrepancies, bringing prosecutions and making management decisions. What was missing was a fair relationship between responsibility and the information, authority and evidence needed to exercise it.

This distinction is the starting point for the book. Putting a human beside, before or after an automated system does not, by itself, make the system accountable. It may simply provide a person to blame when the surrounding controls fail.

The phrase *human in the loop* describes where a person appears in a workflow, but says almost nothing about what that person can know, decide or prevent. They may have a seat, a screen, an alert or perhaps an approval button; none of those proves control. Before treating human presence as a safeguard, we need to examine the capability attached to it.

That is the accountability illusion: the belief that because a person is present, responsibility has been meaningfully placed and risk has been controlled.

The person and the system

Automation is often presented as a transfer of work from people to machines, which is only partly true: the activity may move, but the consequences don’t disappear. A model can recommend a loan decision, classify a medical image, draft a contract or instruct another system to transfer money. If the result harms someone, the explanation can't end with the observation that "the software did it", and it shouldn’t automatically end with the nearest employee either.

The most visible person in a workflow isn't necessarily the person or organisation that shaped it. The operator may not have selected the model, determined its training data, designed the interface, set the operating threshold, approved the deployment or decided how much work one person should supervise. They may be responsible for a task while having little influence over the conditions under which the task is performed.

Organisations have always distributed action across people, tools, rules and institutions. AI makes that distribution harder to see. A conventional process often leaves recognisable decision points, such as someone signing a contract, releasing a payment or sending a letter. An AI-enabled process can turn those moments into a continuous chain of classifications and actions, so by the time a human sees an output, the consequential decision may already have been made. The design of the workflow therefore matters at least as much as the job title attached to it.

Consider three arrangements that might all be described as human oversight:

1. A radiologist receives an AI-generated observation, examines the original scan and decides what enters the clinical report.

2 A fraud analyst sees a queue of transactions that have already been blocked and can release individual payments after reviewing the supporting data.

3 A compliance manager receives a weekly dashboard summarising thousands of decisions made and executed by an automated system.

The first person makes the operative decision; the second can reverse an action after it has temporarily taken effect; the third can identify patterns only after the system has acted. These are different allocations of authority, time and risk, even though calling all three *human in the loop* makes them sound like minor variations of the same control.

This book uses three Loop patterns to distinguish those arrangements. In **Review**, usually called Human-in-the-Loop, a person examines an individual output before action. In **Monitor**, or Human-on-the-Loop, the system operates within defined boundaries while people watch its behaviour and intervene. In **Own**, or Human-Accountable-for-the-Loop, a named owner remains answerable for the workflow's authority, controls, evidence and outcomes even though no person reviews every action.

These patterns describe governance, not levels on a maturity ladder. A workflow may combine them: routine cases can operate inside owned limits, exceptions can be monitored and a narrow class of decisions can require individual review. The correct pattern depends on consequence, speed, scale and the judgement that can't safely be delegated.

The label still proves nothing by itself. A person might review every output without understanding any of them, a monitor may receive signals too late to intervene, or an owner may be named without being able to change or suspend the system. Interaction becomes accountability only when the surrounding capabilities are real.

Human contact with the workflow tells us very little. The organisation has to build a workable relationship between human judgement and machine action.

The approval button

Few interface elements carry more organisational meaning than a button labelled **Approve**. It creates a timestamp, identifies a user and can be shown to an auditor, apparently marking the moment at which machine work became a human decision.

Sometimes that is exactly what it does: a well-designed approval step can hold an action in a reversible state, present the relevant evidence and give a qualified person a genuine choice. The problem lies not in the button but in what organisations infer from its existence.

An approval is meaningful only when four conditions are present.

Information

The reviewer needs the information required to judge the proposed action, which does not mean exposing every internal parameter of a model; more information can obscure as easily as it can illuminate. What matters is the evidence material to the professional decision: the relevant input, the proposed output, the applicable rule or policy, known limitations and enough provenance to challenge what the system has produced.

A confidence score isn't a substitute for this context. It may not be calibrated, and in many generative systems it may not be available in a meaningful form at all. An explanation generated by the same model isn't independent evidence. A reviewer needs to see what bears on the decision, not merely a more fluent account of the system’s conclusion.

Time

The reviewer needs enough time before the action becomes irreversible. That period may be minutes in a payment workflow, hours in a hiring process or fractions of a second in an industrial safety system. Where the available intervention time is shorter than a person’s realistic response time, human review can't be the primary safety mechanism. The architecture has to contain the risk automatically or place the system in a safe state until someone can act.

A notification delivered after execution supports investigation, not prevention. Calling it approval changes nothing.

Authority

The reviewer needs a recognised right to disagree, and that right has both organisational and technical dimensions. A policy may say that an analyst can reject an output while performance targets punish anyone who slows the queue. An interface may display a rejection option while requiring three levels of permission to use it. A manager may be accountable on the organisation chart yet lack access to suspend the service.

Formal authority that can't be exercised in practice is ceremonial, just as technical access without organisational protection is fragile; both have to be real.

Effect

The reviewer's decision needs to change what happens. Pressing Reject should prevent, reverse or redirect the relevant action as the workflow promises; escalating should reach someone able to decide, while suspending should place the system in a defined and safe condition. If the interface records dissent while the automated process continues, the human has supplied commentary, not control.

These four conditions, information, time, authority and effect, turn a click into a decision. Remove any one of them and the approval begins to resemble evidence gathered for the organisation rather than power granted to the individual.

This is why approval buttons are so attractive as governance theatre. They are cheap to add, easy to count and reassuring to describe. They turn the difficult question *who could have prevented this outcome?* into a database field: `approved_by`.

The field proves that someone clicked a button, all the while leaving their understanding, alternatives and practical effect unknown.

Why good people become rubber stamps

The usual response to weak review is to demand greater vigilance, reminding staff to read carefully, challenge the system and take personal responsibility. That response treats unreliable oversight as a character defect when it is often the predictable result of badly designed work.

Human-factors researchers were describing this problem long before the current wave of AI. Raja Parasuraman and Victor Riley distinguished between the use, misuse, disuse and abuse of automation. They described misuse as over-reliance that can produce monitoring failures and decision bias; disuse as the neglect of automation, often following false alarms; and abuse as the introduction of automation without adequate regard for its effects on human performance.[2]

That final category deserves more attention because organisations tend to discuss automation bias as a weakness in the operator: the human trusted the machine too much. Yet trust is shaped by the system around the operator; if the tool is usually correct, the queue is long, disagreement is slow and approval is the

default, reliance isn't surprising. It may be the behaviour the workflow was designed to produce.

Four pressures repeatedly degrade review.

Reliability

Automation is most difficult to supervise when it works well almost all the time. A reviewer who encounters an error in every third case remains alert. A reviewer who encounters one error in ten thousand cases is being asked to sustain attention through hours of correct output in anticipation of a rare failure. The better the automation performs on ordinary cases, the harder it can become for a person to detect the exceptional one.

This creates an uncomfortable truth: the human fallback may become less capable as the system becomes more capable. Skills will decay, attention wanders and the person sees fewer examples requiring independent judgement. The organisation continues to describe the human as its safety layer even as the conditions needed to maintain that layer disappear.

Volume

Review capacity has a physical limit. If a system generates ten thousand alerts and a team can examine five hundred properly, the organisation has not created ten thousand human-reviewed decisions. It has created a prioritisation problem and hiding that mismatch behind a completion target encourages superficial approval and makes the audit record less truthful.

Sampling may be a legitimate control, as may automated containment, risk tiers and retrospective review; the important point is to name the control honestly. A sample isn't comprehensive review, a person watching aggregate metrics isn't approving individual decisions, and a queue cleared under impossible time constraints isn't evidence that every item received professional judgement.

Defaults

Interfaces teach users what the organisation expects. If approving requires one click while rejecting requires a written explanation, an escalation category and a conversation with a manager, the path of least resistance is obvious. If the model’s recommendation is displayed prominently while contradictory evidence sits behind another tab, the recommendation acquires authority through layout. If an item proceeds automatically when the reviewer does nothing, silence becomes consent.

These choices are sometimes defended as matters of usability, but they are also policy: the interface allocates friction, and friction allocates behaviour.

Consequences

People learn whether challenge is truly welcome. An organisation can celebrate critical thinking in its governance policy while rewarding throughput in its performance system. Reviewers quickly discover whether overrides are treated as valuable interventions or awkward exceptions. If disagreement consistently creates more work, delays colleagues and attracts scrutiny, the permission to challenge will be exercised far less often.

Individual responsibility remains, but now in context. Competent people still make mistakes, ignore evidence and breach duties, and accountability should recognise those failures alongside the system design that shaped their choices. The task is to reconstruct how the outcome became possible, wherever that account leads.

Four ideas that should not be confused

Discussions about AI governance become slippery because the same words are used for different things. I use four of them carefully throughout this book.

Responsibility is an assigned duty: a reviewer may check an output, an engineering team may maintain a service, and several people may hold different responsibilities within the same workflow.

Authority is the legitimate power to decide or act. It includes permission, access and the practical ability to exercise them. Responsibility without corresponding authority creates exposure: a person is expected to produce an outcome while depending on choices they can't control.

Accountability is the obligation to answer for how a system was designed, authorised, operated and corrected. It can exist at several levels. An operator may answer for a particular intervention; an executive may answer for accepting the deployment risk; an organisation may answer for the service it provided. Accountability does not require pretending that only one actor influenced the result. It requires clarity about who must explain which decisions and who is expected to put them right.

Liability is a legal conclusion. It may arise through statute, contract, negligence, professional duty, vicarious liability or other rules that vary by jurisdiction and

context. A person's ability to intervene may be relevant, but it isn't a universal precondition. An organisation can't make liability disappear by designing a workflow in which nobody can stop the machine. Nor should a governance framework claim to decide legal liability from a system diagram.

Keeping these concepts separate prevents two opposite errors. One is scapegoating: assigning a broad organisational failure to the person who was nearest the event because their name appears in the log. The other is absolution: treating distributed causation as proof that nobody can be held to account.

Although complex systems involve many contributors, they still need owners, decision rights and routes for remedy. Governance isn't a search for one guilty person to pin it on; it makes the system's choices visible enough for responsibility to be assessed fairly and future harm prevented.

Oversight is a design claim

The presence of a human reviewer is often described as though it were an ethical property of the product: *this system is safe because a human remains in control.* I think that statement should be treated as a testable design claim. We need to know what the human controls, when they can act, which information reaches them, what happens if they disagree and what the system does if they take no action. The answers should change with the risk.

A writing assistant that proposes alternative wording need not pause for a governance committee before displaying a sentence. A system that drafts a customer email can rely on the user choosing whether to send it when that decision point is real. A system that sends thousands of messages, denies access to a service or controls physical machinery needs different boundaries. The magnitude, reversibility, speed and scale of the possible harm determine what kind of human control can work.

This is one reason universal prescriptions fail. Requiring a person to approve every low-risk action can create delay without improving safety. Relying on a person to approve every high-speed action can create the appearance of safety without the possibility of achieving it. In one context, a stop mechanism may be a user-facing control. In another, safe interruption may require automated isolation, transaction rollback, redundant physical controls or a staged shutdown. Judge the collection of controls against the system's actual failure modes, not against a feature checklist.

Current governance frameworks increasingly reflect this functional view. Once the relevant high-risk rules apply, Article 14 of the European Union's AI Act requires high-risk AI systems to be designed so that assigned people can, as appropriate, understand their capacities and limitations, remain alert to automation bias, interpret outputs, disregard or reverse them and interrupt operation so the system reaches a safe state. The provision does not turn every AI output into a mandatory human approval. It describes capabilities that must be appropriate and proportionate to the system's risks.[3]

The US National Institute of Standards and Technology takes a similarly organisational view. Its AI Risk Management Framework calls for clear roles, trained and empowered personnel, executive responsibility for deployment risk, continuing monitoring and differentiation between forms of human-AI oversight. It also recognises that some systems may not need direct human oversight at all.[4]

I think that direction is sensible because oversight is not a single control. It is a relationship between people, technology and institutions that has to work over the life of the system.

The accountability test

Before an organisation claims that a human is in control of an AI-enabled workflow, ask five questions.

1. What can the system cause?

Start with actions and consequences, not model capabilities. What can this workflow send, file, deny, purchase, publish, delete, diagnose or move? Then consider how easily a mistake can be corrected, how many people one error could affect and how quickly harm could spread.

An inventory of models won't answer these questions. Accountability attaches to workflows because workflows connect outputs to the world.

2. Where is judgement supposed to occur?

Identify the points at which uncertainty, policy or professional judgement changes what should happen. Some decisions may be delegated completely within defined limits, others may require human consideration, and still others may be prohibited regardless of who requests them.

If a diagram simply places an approval box at the end, it may be recording acceptance after the consequential choice has already been embedded in the system.

3. Who has information, time, authority and effect?

Name the role, then test the reality behind it by recording what the person will see, how long they will have, whether they can disagree without seeking unavailable permission, whether their intervention reliably changes the outcome and whether they are trained for the judgement being assigned.

If the answer varies by shift, vendor availability or system load then that variation belongs in the risk assessment.

4. What happens when the human fails?

People become distracted, misunderstand information, make poor decisions and fail to respond. A design that depends on perfect vigilance isn't a controlled design.

The right fallback depends on context. A transaction might remain pending, a service might degrade to a narrower mode, an action might be limited in scale, or the system might stop safely. Choose that fallback deliberately instead of accepting whatever happens when an alert expires.

This question also exposes whether the person is truly a safeguard or merely the last component to which failure can be attributed.

5. What will remain afterwards?

An accountable system leaves enough evidence to reconstruct material events: relevant inputs and outputs, the system and policy versions in use, actions taken, human interventions and the sources on which consequential claims depended. The required record is risk- and context-dependent, because capturing everything can violate privacy, create security risks and bury investigators in noise, while capturing only the final result makes meaningful review impossible.

The test is practical: after an incident, could an independent person understand what the system did, what the human could see and do, and why the organisation considered the arrangement acceptable?

These questions do not provide a compliance badge. They reveal where the work of governance has not yet been done.

From human in the loop to human accountable for the loop

Human accountability should not mean placing one individual beneath every possible consequence of a complex system, which would reproduce the illusion in a harsher form; nor does it require a person to inspect every automated act. At useful scale, many systems will perform work no human reviews individually.

It means that automation operates inside a structure for which people and organisations remain answerable.

Someone owns the workflow and has the standing to change or suspend it; the system's authority is explicit rather than inferred from its tool permissions, and boundaries prevent or contain actions whose risk can't be accepted. Exceptions can reach people able to challenge not only an individual output but the premise or operation of the system, while material events leave evidence, performance is monitored and changes trigger review. Legal and financial consequences are considered rather than displaced into the phrase *the model did it*.

Together, these capabilities are the substance of Human Accountable for the Loop. They also change how we understand control. The accountable human isn't necessarily the person clicking on each output but may instead determine what the system can do, set its limits, ensure that intervention routes work and answer for the decision to keep it running. The role is ownership of the conditions under which the machine acts, rather than ceremonial supervision of the machine.

The Horizon cases demonstrate why this distinction matters, as individual conduct can matter without turning a system-generated allegation into self-proving truth. A complete account has room for personal duties and institutional design, for the evidence a person could see and the evidence withheld from them, and for the organisation that decided the automated record should carry decisive weight.

The accountability illusion erases that fuller account. It points to the named human and asks no further questions.

Together, those questions become the HAL framework. The first discipline is the most basic one: when an automated workflow acts in the world, who owns the conditions that allowed it to act?

Notes

[1] *Hamilton and others v Post Office Limited* [2021] EWCA Crim 577, Court of Appeal (Criminal Division), 23 April 2021, official judgment and summary. The Post Office Horizon IT Inquiry's first final-report volume, published in July 2025, addresses human impact and redress. In a progress update dated 8 July 2026, the Chair said the remaining five volumes were still months from publication.

[2] Raja Parasuraman and Victor Riley, "Humans and Automation: Use, Misuse, Disuse, Abuse", *Human Factors* 39, no. 2 (1997): 230-253, https://doi.org/10.1518/001872097778543886.

[3] Regulation (EU) 2024/1689, as amended by Regulation (EU) 2026/1744, Article 14, "Human oversight", official EU AI Act Service Desk text. Under the amended timetable, the high-risk rules apply to Annex III systems from 2 December 2027 and to systems covered through Annex I product legislation from 2 August 2028; see the European Commission implementation timeline and amending regulation.

[4] National Institute of Standards and Technology, *Artificial Intelligence Risk Management Framework (AI RMF 1.0)* (2023), particularly GOVERN 1.5, GOVERN 2 and GOVERN 3.2, AI RMF Core and Appendix C: AI Risk Management and Human-AI Interaction. NIST identifies AI RMF 1.0 as the published framework while a revision is in progress; see the current AI RMF page.

CHAPTER 2

The HAL Framework

Human presence does not prove human control. A person can sit inside a workflow without the information, time, authority or practical effect required to change its outcome. An approval can record a click after the consequential choice has already been made. A dashboard can display activity nobody is empowered to stop.

I developed Human Accountable for the Loop (HAL) because the familiar labels describe where a person appears in a process, yet say very little about whether anyone can govern its consequences. HAL tests and designs the alternative.

The alternative begins with a simple proposition:

> *Automation may perform the work, but people and organisations remain answerable for the conditions under which it acts.*

Where HAL fits

I separate three Loop governance patterns because governance discussions often treat reviewing an output, watching a service and owning its operation as interchangeable. They are different jobs with different powers.

Pattern	Human role	Operating arrangement	Typical fit
Review / Human-in-the-Loop	Reviewer	The system produces an output; a person reviews it before action.	Drafting, research, low-volume recommendations and decisions where individual professional judgement must remain operative.
Monitor / Human-on-the-Loop	Monitor	The system operates within defined boundaries; people observe, investigate exceptions and intervene.	Continuous monitoring, screening and medium-volume automation where people can still intervene in time.
Own / Human-Accountable-for-the-Loop	Owner	People define and own the purpose, authority, controls and consequences; the system may act without review of every action.	Agentic systems, orchestration, large-scale triage and autonomous or high-volume operational workflows.

They are not a maturity ladder. Treating them that way would be like treating a brake as an immature form of steering: both are controls, but they solve different problems. HAL does not replace professional judgement where a person must

make the decision. A single workflow can also combine the patterns: an owned system may act on ordinary cases, route anomalies to monitors and hold specified consequences for individual review.

The choice is practical. If the system only produces material for a person to use, Review will often fit. If it operates continuously or at scale and people can still detect and address exceptions, Monitor may fit. If it creates records, routes work, communicates, triggers processes or otherwise affects outcomes without every action being reviewed, HAL is likely to fit. Where failure could be consequential, the eight HAL disciplines provide a useful test even when individual stages retain Review or Monitor controls.

HAL goes beyond putting a reviewer at the end of a process. It is the detailed operating model for the **Own** pattern, made from eight connected disciplines.

The framework's formal domain names are **Ownership, Authority, Limits, Escalation, Evidence, Monitoring, Review** and **Liability**. The detailed chapters translate those domains into operating questions that an owner can answer against the live workflow and its evidence.

What matters to me here is the quality of the joins, not the existence of eight tidy documents. A named owner who lacks authority is a figurehead, while authority without limits leaves the system free to act beyond what the organisation can accept. A limit that has no escalation route creates a dead end; an escalation that arrives without evidence becomes a contest of opinion. Evidence that nobody monitors is only an archive, and monitoring that never reopens approval allows an old decision to govern a system that has since changed. Liability and remedy complete the picture by identifying who carries the consequence. The parts have to work together.

Govern the workflow, not just the model

I use the word *workflow* deliberately because HAL governs the complete AI-enabled service, not a model in isolation.

The model may be the novel component, but consequences emerge from the full system:

- the purpose and policy;
- data and knowledge sources;
- model and configuration;

- tools, credentials and downstream services;
- user interface and default behaviour;
- people, workload and incentives;
- suppliers and contracts; and
- the action or decision experienced by the affected person.

A model can behave exactly as evaluated while the workflow fails: a current policy may be retrieved incorrectly, a safe recommendation presented as a command, a drafting tool given permission to send or a valid classification applied to a population for which it was never intended. Governing only the model is like auditing a payment by inspecting the calculator. The data, account, permission, approval and transfer still determine what happens.

Follow the consequence to draw the workflow boundary. If a customer receives an automated answer and acts on it, the governed object includes the source content, retrieval process, interface and route to correction, as well as the language model that generated the sentence. A practical first question is: what can this workflow send, change, deny or spend?

Eight questions

I frame each HAL discipline as a question because a real owner should be able to answer it about the live system, without translating some opaque governance vocabulary first.

Who owns the outcome?

Ownership connects component responsibilities to the service experienced by the user. The accountable owner has enough organisational standing to keep the workflow within its approved purpose and risk tolerance. They do not perform every control or carry automatic personal legal liability. They can pause deployment, remove a permission, require a new evaluation and send unresolved risk to the senior authority able to accept it.

Evidence of ownership includes a named role and current holder, a defined mandate, a workflow boundary, decision rights and continuity when the owner is unavailable.

What has been delegated?

Delegated authority makes the move from output to consequence explicit. It distinguishes advice, preparation, decision and action, then bounds the grant by actions, objects, counterparties, data, value, volume, conditions, duration and exceptions.

Enforce the grant through actual permissions and workflow state. A prompt saying “never send without approval” is guidance. A service identity incapable of sending until an approved state exists is a control. Ask which permission makes the consequence possible and who can remove it today.

What must not happen?

Hard limits protect invariants that can't depend on model judgement or human vigilance alone. They may constrain data, recipients, value, volume, state transitions or physical action. They are enforced outside the actor being constrained and lead to a defined fallback on uncertainty. A policy can't stop a payment, but a proper control can.

Hard does not mean eternal, infallible or safe to trigger in every circumstance. Limits require threat modelling, controlled change, tested recovery and a safe or degraded state designed for the system.

Where does the problem go?

Escalation routes distinguish an individual exception, an operational incident and a challenge to the workflow’s premise. Each trigger has a destination, decision authority, response time and holding state.

Connect frontline concerns, complaints, appeals, technical incidents and legal decisions through the route. Correcting one case can't be allowed to prevent a pattern from reaching someone able to change or suspend the system.

What can be proved?

Reconstructable evidence links purpose and authority to important inputs, system state, output, human intervention, execution, outcome and remedy. The practical test is whether an independent person can reconstruct last Tuesday’s decision without relying on memory.

Hidden chain-of-thought, universal blockchain logging and exact replay of every hosted model are unnecessary. Preserve enough evidence for an independent person to evaluate what occurred and which controls were available.

How will we know it changed?

Monitoring observes the workflow at the service, input, behaviour, action, human and outcome layers. It detects more than statistical data drift: provider updates, policy change, new permissions, workload pressure, purpose expansion and emerging harm can all invalidate the original approval. Those signals then have to map to decisions: observe, investigate, contain, stop or withdraw.

When must approval reopen?

Scheduled review asks whether the complete governance case still holds. It is triggered both by time and by material changes, performance evidence, incidents, complaints, research, law and provider disclosures.

Approval attaches to a purpose, population, configuration, authority and period of operation. It does not attach permanently to a product name. Review ends with a recorded decision to continue, constrain, narrow, pause, replace or retire.

Who carries the consequences?

Liability mapping records the duties, control, dependencies, evidence, contractual allocation and remedy across providers, deployers, professionals and operators. It follows the complete decision chain and captures contributions that the formal final decision can hide.

Liability mapping isn't a liability verdict

I include liability in HAL so that lawyers receive usable operating facts before a dispute: who owned the workflow, which duties and powers were assigned, what controls existed, who holds the evidence and who can put the outcome right. Liability itself still depends on the jurisdiction, cause of action, contract, statute, professional duty and facts. It may sit with an organisation even where no individual could intervene, or be shared across suppliers and deployers. A missing control may be relevant to negligence, while a clear control can't guarantee a defence.

Before deployment, legal analysis should shape authority, safeguards, disclosure, contracting, insurance, record-keeping and routes for review. After an event, reconstructed evidence supports decisions about notification, remedy and allocation of loss. HAL makes those operating facts clearer; it does not promise immunity.

Proportionality

Not every workflow needs the same controls. Match the effort to the consequence, considering:

- potential severity and scale of harm;
- reversibility and speed;
- effect on safety, rights and essential services;
- sensitivity of the data;
- autonomy and external action;
- uncertainty and novelty;
- populations affected;
- dependence on suppliers; and
- availability of effective remedy.

A private tool that suggests headings for an internal document may need clear ownership, ordinary security and user review. A system that communicates legal deadlines to customers needs controlled sources, evaluation, evidence and a correction route. A system that can freeze accounts or move money needs enforced authority, aggregate limits, staged action, rapid containment and independent review.

Proportionality prevents two forms of failure: too little control exposes people to unmanaged consequences, while too much ceremonial control overwhelms teams, slows low-risk work and encourages bypass. The aim is governance that fits the risk, not the largest possible collection of controls.

The HAL control profile

The framework includes a structured assessment so that an attractive governance description can be tested against evidence. It asks three questions in each domain and records each control as **Absent, Partial, Implemented** or **Tested**. The result is an eight-domain profile rather than a total or maturity band. It keeps the underlying evidence and dissent visible instead of allowing strengths in one area to compensate numerically for a serious weakness in another.

Four controls remain critical gates. The control case fails if the owner can't invoke tested containment in the required time, authority exists only in a prompt,

a hard limit does not actually stop or contain the affected action, or a material decision can't be reconstructed. No overall label can cure one of those failures.

The proposed authority still matters, but it changes the controls and evidence the organisation should expect rather than setting a numerical target. Advice requires a real human decision point and controlled scope; recommendation requires evidence about influence, challenge and escalation; execution requires enforced authority, effective limits, reconstructable action and containment; consequential autonomous action demands the strongest forms of those controls, independent challenge and workable remedy. The completed evidence, critical gates, unresolved uncertainty and applicable legal or professional duties determine whether the decision is to proceed, constrain, stage, narrow, pause or retire. The workbook contains the full twenty-four-test profile and decision record.

The HAL canvas

A one-page HAL canvas can expose whether a proposed workflow is ready for detailed review.

Question	Required answer	Evidence
Purpose	What outcome justifies the workflow, for whom, and what is outside scope?	Approved use case and risk assessment
Ownership	Who owns the end-to-end outcome and where does unresolved risk go?	Owner declaration and decision map
Authority	What may each human and system actor advise, prepare, decide and execute?	Authority envelope matched to permissions
Limits	Which consequences are prevented or contained independently?	Limits register and test results
Escalation	Which events move beyond ordinary handling, to whom, and in what state?	Escalation register and exercises
Evidence	Can a material event be reconstructed without relying on memory?	Evidence specification and reconstruction drill
Monitoring	Which signals trigger investigation or containment while the system operates?	Monitoring plan and decision thresholds
Review	Which time, change, performance, incident and external triggers reopen approval?	Review register and current operating decision
Liability	How are duties, dependencies, evidence, incident work and redress allocated?	Liability and remedy register, contracts and exercises
Remedy	How can an affected person obtain explanation, correction or redress?	User route, service process and incident record
Change	Which changes require testing, reapproval or suspension?	Change classification and review history

The canvas isn't a compliance score. A confident but unsupported answer is a risk signal. Its value lies in forcing technical, operational and legal participants to describe the same workflow.

HAL across the lifecycle

I use a lifecycle view because an approval can be sound on Monday and obsolete after a supplier update, a new permission or a change of population. Accountability is maintained through decisions over time.

Approve

Define the purpose, owner, affected people, authority envelope, risk tolerance and evidence required to justify deployment. Identify non-AI and narrower alternatives.

Build and acquire

Translate authority into identities, permissions, policy enforcement, state transitions and limits. Contract for supplier information, change notice, incident support and exit.

Evaluate

Test capability, known failure modes, affected populations, human operation, hard limits, escalation and reconstruction in the real workflow, not a model demonstration.

Deploy

Constrain initial exposure where uncertainty remains. Confirm ownership, responder availability, monitoring, fallback and remedy before external action begins.

Operate

Monitor the complete workflow, investigate signals, support challenge, review performance and maintain evidence. Treat workarounds and complaints as governance data.

Review

Reconsider the complete operating case on schedule and when material triggers occur. Test whether claims, controls, contracts and remedy still describe the live workflow, then record a new decision.

Change

Classify changes to models, data, tools, permissions, interfaces, population and policy. Re-test and reauthorise where the original control case may no longer apply.

Respond

Contain harm, preserve evidence, make decisions at the correct escalation level, communicate with affected people and convert incidents into control improvement.

Retire

Remove permissions, preserve or delete evidence appropriately, support remaining obligations and ensure users do not continue relying on an abandoned channel.

Seen across the lifecycle, approval can no longer masquerade as a permanent certificate attached to a changing system.

An illustrative workflow

Take an internal agent that triages requests sent to a legal team. It reads the request, assigns a category, creates a matter record and routes the work. No lawyer reviews every classification, and the system does not provide legal advice or decide the matter.

Under HAL:

- the head of legal operations owns the triage workflow while lawyers retain responsibility for professional decisions;
- the agent may read the intake, classify it, create a draft record and route it only to approved queues;
- it can't close a matter, communicate externally or change a legal deadline;
- protected matters and uncertain conflicts enter a restricted queue;

- repeated routing errors, missing source data and access outside the approved matter class trigger escalation;
- each route is linked to the intake, configuration, rule and any human correction;
- monitoring covers queue delay, misrouting, overrides, complaints and changes in request types;
- material changes to data access or action permissions require renewed approval;
- scheduled review checks whether the original triage purpose and controls remain justified; and
- contracts and operating procedures state who investigates, corrects and remedies failures across any supplier boundary.

The framework does not insert a lawyer into every routing decision. It makes autonomous routing answerable to an owned and bounded system.

Human in the Loop describes a possible interaction; Human Accountable for the Loop describes the conditions under which an organisation can justify allowing the workflow to operate. Ownership comes first because someone has to answer for those conditions before the other controls can mean anything.

Continue reading

The first two chapters establish the problem and map the HAL framework. The full book turns that map into an operating method.

Chapters 3 to 10 examine ownership, delegated authority, hard limits, escalation routes, reconstructable evidence, live monitoring, scheduled review and liability mapping. Chapter 11 then applies all eight parts to a single supplier-contracting workflow, showing how they work together before a decision is made to deploy.

The appendix turns the framework into a 24-test control profile, evidence record and first-90-days implementation route.

To continue, return to this book's Leanpub page for the complete edition. Companion worksheets and interactive tools are available at theloop.legal.