MICKAI™

MICKAI EBOOK SERIES · No. 18

The Trust Layer Between AI and the Chain.

Everyone is building models and everyone is building chains. Almost no one is building what makes a model's output admissible.

AUTHOR

Micky Irons

Founder and named inventor, Mickai LTD.

19 June 2026 · v1 · mickai.co.uk

Mickai LTD · Companies House 17166618 · press@mickai.co.uk · mickai.co.uk
UK IPO register, named inventor Mickarle Wagstaff-Irons · Trade mark UK00004373277

TABLE OF CONTENTS

Contents

FOREWORD

A note from the author

Two enormous waves are rising at once. The first is artificial intelligence, where every week brings a model larger, faster and more capable than the one before. The second is the chain, the long quiet revolution in cryptographic ledgers that lets strangers agree on what happened without trusting a central referee. Both are real and both are accelerating. What almost nobody is building is the bridge between them, the layer that takes the output of a model and turns it into something a court, a regulator, an auditor or an adversary can actually rely on. That bridge is the subject of this book, and it is what I have spent my working life building inside the Sovereign Intelligence Operating System.

I am Micky Irons, founder and chief executive of Mickai, and the named inventor on a portfolio of 101 filed UK patent applications carrying around 2,234 claims, all owned by Mickai Ltd. I did not come to this from an academic seat or a venture fund. I came to it from one stubborn observation. An intelligent system that cannot prove what it did, under whose authority it did it, and that the record has not been altered since, is not a system you can put in front of anything that matters. It is a demonstration. The gap between a clever demonstration and an admissible system is the whole game, and the industry has been racing straight past it.

This book makes a single argument from many angles. Proof of intelligent action, the sealed, anchored, authority-bound record of what a model decided and why, is not a feature you can bolt on after the fact. It has to be designed into the substrate, into the place where the decision is born, or it will never hold weight. I will show you why retrofitted logging fails, why a signature without an anchor is a promise without a witness, why an anchor without authority is a timestamp on an orphan, and how the three together change what an AI decision is allowed to be. I will use the architecture I know best, our own, because I will not write about systems I have not built.

I have written this in plain British English, with no hype and no jargon for its own sake. Where I make a claim about our own work I state exactly what is filed, what is built and what is still being trained, because a book about trust that overclaims would be a contradiction in terms. If you build models, this is about what happens after the model speaks. If you build chains, this is about what is worth anchoring in the first place. If you buy, regulate or insure these systems, this is the question you should put to every vendor who walks through your door. Let us begin.

Micky Irons

Founder and named inventor, Mickai LTD · 19 June 2026

PART I · THE PROBLEM

A model can be brilliant and still be worthless as evidence.

1. The demonstration trap

Walk into any technology showcase in 2026 and you will see something remarkable. A model reads a contract in seconds, flags the unusual indemnity clause, drafts a response and recommends a position. The room nods. The demonstration works. Then someone asks the only question that matters in the real world, the one the demonstration is designed never to provoke. Six months from now, when this recommendation is challenged, how do we prove the model actually saw that clause, that nobody quietly edited the output, and that the system was authorised to act at all. The room goes quiet, because the honest answer is that we cannot.

This is the demonstration trap, and almost the entire industry is caught in it. We have optimised relentlessly for the moment of output, the instant the model produces something impressive, and treated everything after that moment as somebody else's problem. The result is a generation of systems that dazzle on stage and are inadmissible in practice. They can tell you what they think. They cannot prove what they did. In any setting where the answer carries consequence, money, liberty, safety, reputation, that distinction is the difference between a tool and a liability.

> **We optimised for the moment a model speaks, and forgot that everything consequential happens in the moment someone has to rely on it.**

Consider what we demand of a human professional in the same position. A doctor who makes a diagnosis writes it into a record, signs it, dates it, and that record is kept in a system where tampering leaves traces. A solicitor who advises a client opens a file. A pilot follows a checklist that is logged. We do not accept their judgement in a vacuum. We accept it because it is wrapped in a structure of accountability that lets us reconstruct, afterwards, exactly what was known and decided and by whom. We have deployed artificial intelligence into these same arenas while stripping away the one thing that made human judgement trustworthy, the durable, attributable, tamper-evident record.

The trap is seductive because its cost is deferred. Nothing goes wrong at the demo. Nothing goes wrong in the pilot. The bill arrives later, when a decision is contested and no record survives scrutiny, when a regulator asks for an audit trail and you produce a log file that anyone with database access could have rewritten. By then the system is in production, the architecture is set, and the proof you needed had to exist at the moment of decision, not be conjured afterwards. The trap is not that these systems lie. It is that they cannot be made to tell the truth on demand.

2. The output is not the artefact

The deepest mistake in how we think about artificial intelligence is that we treat the output as the product. The model returns a paragraph, a number, a classification, a decision, and we ship that. But the output, on its own, is the least durable thing in the entire transaction. It is a string of text that any process can copy, alter, truncate or fabricate, and it carries no inherent evidence of its own origin. Strip away the surrounding context and a genuine model output is indistinguishable from one a person typed into a box and claimed the machine produced.

What an admissible artefact actually contains

The artefact with evidential value is not the answer. It is the answer bound to its circumstances: the output together with which model produced it, on which inputs, at what time, under whose authority, in what configuration, and sealed so that any later change is detectable. The answer is the smallest part of that bundle. Everything that makes it trustworthy is the context wrapped around it, and that context must be captured at the instant of generation because most of it cannot be reconstructed later. You cannot prove which version of a model ran if you did not record it then. You cannot prove the input was not altered if you never sealed it.

This is why exporting a chat transcript proves nothing. A transcript is a presentation of an output, formatted for human reading, with every cryptographically meaningful detail already discarded. It is a photograph of a footprint, not the footprint. It tells you what was shown on a screen, which is exactly the layer most easily faked. The artefact that matters lives below the presentation, in the structured record that ties the output to a verifiable chain of facts about how it came to exist.

Once you see the output as a fragment of a larger artefact, the whole engineering problem reorients. The question stops being how do I make the model produce a better answer and becomes how do I capture, at the moment of production, everything needed to stand behind that answer later. The first is a content problem. The second is an evidence problem. They are not the same discipline, and the industry has hired almost exclusively for the first while assuming the second will sort itself out.

The Mickai pantheon.

3. Why logging is not proof

The standard rebuttal is that we already have logging. Every serious system writes logs. The model call is logged, the input is logged, the response is logged, and surely that is the record we are asking for. It is not, and the gap between a log and a proof is the gap this entire book exists to close. A log is a story a system tells about itself, written into a place the system controls. Its fundamental weakness is that whoever can write the log can rewrite it, and nothing in the log can tell you whether they did.

A genuine proof has three properties ordinary logging lacks. It is tamper-evident, meaning any alteration is detectable, not merely discouraged by access controls. It is independently verifiable, meaning a party who does not trust you and has no access to your systems can still confirm the record is intact. And it is bound to a moment in time you could not have backdated. A log in your own database has none of these. You can edit it, you alone can vouch for it, and its timestamps are whatever your clock said, which is whatever you choose to make them say.

> **A log is a story a system tells about itself. A proof is a fact a stranger can check without trusting the storyteller.**

This is not a theoretical worry about malicious operators. It is structural. The moment a dispute arises, the value of your evidence depends entirely on whether the other side has to take your word for it. If your audit trail lives in infrastructure you control, with no external anchor and no cryptographic seal, every entry in it is contestable on the simple ground that you could have written it yesterday to suit your case. Good faith does not help you here. The point of proof is to convince precisely the people who do not assume your good faith, and a log can never clear that bar because it was never designed to.

So the problem statement for the rest of this book is now sharp. We need to turn the act of an intelligent system into an artefact that is sealed against tampering, anchored to a moment no one could fake, and bound to the authority under which it was taken. We need proof of intelligent action. And we need it produced at the moment of action, by the substrate itself, because every attempt to add it afterwards reproduces the very weakness it was meant to cure.

PART II · THE ANATOMY OF PROOF

Three things turn an output into evidence: a seal, an anchor and an authority.

4. Sealing: making tampering visible

The first pillar of proof is the seal. Sealing binds the record so that any change to it, however small, becomes detectable by anyone. The mechanism is a cryptographic signature computed over the full content of the record, the output and all its context, using a private key only the signing system holds. Anyone with the matching public key can verify that the record is exactly as it was when signed, and that the holder of that key signed it. Change a single character and the signature no longer verifies. The tampering is not hidden. It is exposed.

Inside the Sovereign Intelligence Operating System we seal every consequential action into what we call an Open Audit Record. The word consequential is doing real work. Not every keystroke needs a court-grade seal, but every action that could later be questioned, a decision, a recommendation, an automated step that changes the world, is sealed at the moment it happens. The record captures the action and its surrounding facts, and the seal is computed before that record is allowed to rest anywhere. Sealing is not a post-processing step we run at the end of the day. It is part of the act itself.

Why the algorithm choice is not a detail

Here is where most designs that bother to sign at all make a quiet, fatal compromise. They reach for classical signature schemes that are strong against today's computers and helpless against tomorrow's. A record sealed today may need to hold weight in ten or twenty years, well inside the horizon where a sufficiently capable quantum computer could forge classical signatures retroactively. An adversary recording your sealed records now, to break them later, defeats the entire purpose of the seal. So we seal under FIPS 204 ML-DSA-65, a post-quantum signature standard designed to remain unforgeable against quantum attack. We are sealing for a future that has to verify, not just a present that wants to demonstrate.

The discipline here is total. A seal that covers only part of the record leaves the rest free to be altered, so the seal must cover everything that matters, input, output, model identity, configuration, timing, authority, in one indivisible signed object. A seal that the same party can quietly recompute proves only that the party is internally consistent, so the keys and the sealing path must be protected such that resealing a forged record is not a move available to the operator. Sealing done properly is unforgiving, and it has to be, because its whole job is to make dishonesty visible.

The Mickai pantheon.

5. Anchoring: borrowing time you cannot fake

A seal proves the record has not changed since it was signed. It does not, by itself, prove when it was signed. The signer controls their own clock, and a signature can carry any timestamp the signer cares to assert. This is the second gap, and anchoring closes it. To anchor a record is to commit a fingerprint of it into an external ledger whose history cannot be rewritten, so that the record's existence at a given time becomes a fact witnessed by a system you do not control and cannot bribe.

The mechanism is elegant. You take the sealed record, compute a compact cryptographic digest of it, and write that digest into a ledger. Once the ledger incorporates your digest into its permanent, append-only history, you hold an unforgeable statement that the record existed in exactly its current form at that point in the ledger's timeline. You cannot now backdate the record, because doing so would mean rewriting the ledger's history, which by design you cannot. You have borrowed the ledger's immutability and lent it to your private record.

> **A signature says this is mine. An anchor says and it already existed when the rest of the world was watching.**

This is why the choice of what to anchor into is a question of how much immutability you are actually buying. A ledger is only as good a witness as it is hard to rewrite. We anchor into Pantheon, our sovereign Layer 1, and Pantheon in turn anchors to Bitcoin, the most thoroughly tested and most expensive-to-rewrite ledger humanity has built. This is not tribal loyalty to any chain. The strength of your anchor is the cost an adversary would have to pay to fake the timeline, and you want that cost to be astronomical. Anchoring to a ledger that is cheap to reorganise is anchoring to sand.

Anchoring also solves a problem pure sealing cannot touch, the deleted record. A sealed record that is quietly destroyed leaves no evidence it ever existed. But once its digest is anchored, the anchor remains in the public ledger forever, a permanent marker that something specific existed at a specific time. If the record later cannot be produced, or is produced in altered form, the anchor convicts the alteration. The ledger remembers what your own systems might be tempted to forget. That is the deeper gift of anchoring, it makes absence and alteration equally detectable.

6. Authority: the binding most systems forget

Seal and anchor together give you a record that is intact and that provably existed at a given time. One thing is still missing, and it is the thing that most distinguishes a system you can govern from one you merely operate. The record does not yet say under whose authority the action was taken. An action can be perfectly sealed and perfectly anchored and still be something the system was never permitted to do. Without the authority binding, you have proof of what happened and no proof that it was allowed to happen.

Authority binding means the record carries, as part of the sealed content, a verifiable statement of the permission under which the action was taken: which operator or policy authorised it, what scope that authority granted, and that the action fell within that scope. This is the difference between an audit trail and a chain of command. A bound record does not just tell you the model recommended a payment. It tells you the model was operating under an authority that permitted recommendations of that kind, up to that value, in that context, and that the authority was valid at the moment it was exercised.

Why authority has to be inside the seal

It is tempting to keep authority as a separate access-control concern, handled by some permissions system off to the side. That is exactly the mistake. If authority lives outside the sealed record, the record does not prove the action was authorised, it only proves the action occurred, and you are back to taking someone's word for the permissions. Bind the authority into the sealed and anchored record itself and the permission becomes part of the evidence. The proof now answers all three questions at once, what was done, when, and by what right. Pull any one of the three out and the proof collapses into something weaker than it looks.

This is the binding the industry most consistently forgets, because it is the least glamorous and the most political. Sealing is a cryptography problem and anchoring is a ledger problem, but authority is a governance problem, and governance does not demo well. Yet authority is what turns a technically impressive record into an accountable one. An action without a bound authority is an orphan, impressive perhaps, sealed perhaps, anchored perhaps, but answerable to no one. The whole purpose of proof of intelligent action is to make machine decisions answerable, and a decision answers to its authority or it answers to nothing.

The Mickai pantheon.

PART III · WHY IT CANNOT BE BOLTED ON

Proof captured after the fact is proof you already lost.

7. The retrofit fallacy

The most expensive belief in enterprise artificial intelligence is that accountability is a layer you add later. Ship the capability now, the reasoning goes, and wrap it in audit and compliance once the value is proven. This is the retrofit fallacy, and it fails for a reason that is structural, not a matter of effort or budget. Most of what makes a record provable exists only at the instant of the action. The model version, the exact input, the configuration, the authority context, the precise moment, these are facts about a fleeting event. If you did not seal them as the event happened, they are gone, and no amount of later engineering can recover a fact that was never captured.

You can, of course, add logging to an existing system. What you cannot add is proof, because proof requires that the sealing happen inside the trust boundary of the action, before the output leaves the place where it was born. A wrapper bolted on outside that boundary can only attest to what it was handed, not to what actually occurred. It signs a copy, and a copy is exactly the thing that could have been altered before it reached the wrapper. The retrofit signs the photograph of the footprint and calls it the footprint, and it does so with complete sincerity, which is what makes it dangerous.

> **You cannot seal a moment that has already passed. The footprint had to be captured while the foot was still in the ground.**

There is a second, subtler failure. Even if you could capture everything after the fact, a bolted-on audit layer is a separate system, with its own trust assumptions, its own keys, its own attack surface, sitting outside the thing it claims to attest. It becomes one more component the operator controls and could compromise, which means it inherits the very weakness it was meant to cure. The only audit trail an adversary cannot quietly defeat is one produced by the substrate itself, as an inseparable part of the action, with no gap between the doing and the recording for anyone to slip through.

8. Proof at the substrate, not the application

If proof cannot be bolted on, it has to be born in. The seal, the anchor and the authority binding have to be properties of the substrate, the operating layer in which intelligent actions occur, not properties of each application that happens to remember to ask for them. This is the architectural heart of a Sovereign Intelligence Operating System. The sealing is not a library an application chooses to call. It is part of what it means to take an action at all. You cannot do a consequential thing in the system without the system sealing it, the way you cannot move in physical space without leaving the world slightly rearranged.

Putting proof at the substrate changes who has to be trusted. When sealing lives in the application, every application is a place where the discipline could lapse, where a developer under deadline could skip the seal, where a corner could be cut and no one would notice until the dispute. When sealing lives in the substrate, the discipline is enforced once, in one place, by the platform, and every action inherits it whether the application author thought about it or not. Trust concentrates where it can be most carefully guarded, instead of being diffused across every team that touches the system.

The fifty brains and one record

In our architecture the intelligence is not one model but fifty specialised brains, twenty-five domain and twenty-five operational, running on the operator's own hardware and fully offline-capable. They span intelligence and defence, governance and strategy, health and humanity, science and engineering, and identity, sitting on top of a kernel and a body of custodians and specialists. What matters for this book is that no matter which brain acts, the act flows through the same sealing and anchoring path. The intelligence is plural and specialised. The proof is singular and uniform. That is only possible because proof lives below the brains, in the substrate they all stand on.

This is also why offline capability and proof reinforce each other rather than conflict. A system that must phone home to some external service to record its actions has made its accountability dependent on connectivity and on a third party. A substrate that seals and anchors locally can produce proof on an aircraft, in a secure facility, on a contested network, anywhere, and reconcile the anchor once a ledger connection is available. Sovereignty over your intelligence and provability of your intelligence turn out to be the same design goal seen from two directions, and a system that gives you one without the other has given you neither in full.

The Mickai pantheon.

9. The cost of getting the order wrong

Architecture is largely a question of what you decide first. Decide on capability first and accountability second, and you build a system whose entire shape assumes the output is the product, with proof as an afterthought clinging to the outside. Decide on accountability first, and capability flows through it, every action shaped from birth to be provable. These two systems can look identical in a demonstration and could not be more different the first time a decision is challenged. The order of decisions is not a stylistic preference. It is destiny, baked into the foundations long before anyone notices it mattered.

I have watched organisations learn this the hard way, and the lesson always lands as a surprise even though it should not. They deploy a capable system, it performs, and then a single contested decision exposes the absence at the centre. There is no record that survives scrutiny. The remediation is not a patch. It is a re-architecture, because the proof had to be designed in at the substrate and instead the substrate was designed around its absence. The cost of getting the order wrong is not the cost of adding a feature later. It is the cost of rebuilding a foundation while the building stands on it.

> **You can add capability to an accountable system. You cannot add accountability to a system built to live without it.**

There is a regulatory clock on this too. Frameworks like the European Union's AI Act, and the sectoral rules following in its wake, are moving steadily from principles towards demands for demonstrable records of how high-impact automated decisions were made. The systems that will weather that transition are the ones that can already produce sealed, anchored, authority-bound proof on request, because that capability cannot be conjured the week the auditor calls. It had to be there from the first line. The organisations deciding their architecture today are deciding whether they will be ready or whether they will be rebuilding under pressure.

So the practical counsel is simple and uncomfortable. If you are choosing or building an intelligent system for anything that carries consequence, the question is not only what can it do. The question is what can it prove, who decided it could act, and will that proof hold against someone who does not trust you. Ask those questions before the architecture sets, because afterwards the honest answer may be that the proof you need would require starting again. Get the order right and everything else becomes possible. Get it wrong and the cleverest model in the world is still just a very expensive demonstration.

PART IV · THE TRUST LAYER IN PRACTICE

When proof is built in, intelligence finally becomes something you can stand behind.

10. What changes when actions are admissible

Something shifts the moment a system can produce proof of intelligent action on demand. The relationship between operator and machine changes from supervision to delegation. When you cannot prove what an automated system did, you have to keep a human hovering over it, checking, re-checking, ready to carry the blame the machine cannot. That human is the real bottleneck in every serious deployment, and they are there because the system cannot stand behind itself. Give the system the ability to seal, anchor and bind every action, and the human can step back from the loop and into oversight of the proofs, which is a far more scalable place to stand.

Admissibility also changes what you are willing to let the system touch. An automated decision you can fully reconstruct and defend afterwards is one you can let reach into consequential territory, payments, clinical flags, legal positions, operational commitments, because the safety net is no longer a hope that nothing goes wrong but a guarantee that whatever happens is provable. The fear that keeps capable systems penned into low-stakes corners is, at bottom, the fear of an action you cannot account for. Remove that fear at the root and the frontier of what you can responsibly automate moves outward.

For the people on the receiving end, the change is just as real. A patient, a claimant, a defendant, a customer told that a machine made a decision about them is, today, often told simply to accept it. With proof of intelligent action they can be shown exactly what was decided, on what basis, under whose authority, and assured the record cannot have been altered to suit the institution. That is not a small courtesy. It is the difference between automated power that is accountable and automated power that is merely imposed, and over time it is the difference between systems the public will tolerate and systems it will revolt against.

The Mickai pantheon.

11. The open record and the right to verify

We named our sealed record the Open Audit Record deliberately, and the word open carries the weight. A proof that only its author can verify is not really a proof, it is an assertion with extra steps. The value of the record is realised only when a second party, a regulator, an opponent, an auditor, an ordinary person, can independently confirm that it is intact, that it existed when it claims to have, and that it was taken under the authority it names, without needing access to our systems and without needing to trust us at all.

This is what verifiability really means, and it is a higher standard than transparency. Transparency is being shown things. Verifiability is being able to check them yourself. A system can be transparent and still untrustworthy, because it shows you only what it chooses and you have no way to confirm the showing is honest. A verifiable system hands you the cryptographic tools to confirm the facts independently, and in doing so it removes itself from the position of being believed. The seal is checked against the public key. The anchor is checked against the public ledger. The authority is checked against the bound permission. At no point are you asked to take anyone's word.

> **Transparency shows you what it chooses. Verifiability lets you check what it cannot hide.**

There is a sovereignty dimension to this that I care about deeply. The right to verify should not depend on the goodwill of the vendor or the survival of the company. Because the anchor lives in a public ledger and the seal is checkable against a public key, an Open Audit Record remains verifiable even if Mickai vanished tomorrow. The proof outlives the institution that produced it. That is the correct relationship between people and the systems that increasingly decide things about them. The proof

belongs to the world, not to the company, and it stays checkable long after any particular company is gone.

This durability is also why post-quantum sealing was never optional for us. A record meant to be verifiable for decades has to be sealed against the computers of decades hence, not merely the computers of today. Choosing ML-DSA-65 under FIPS 204 was a statement that we intend these proofs to still hold when the people relying on them have grandchildren. A trust layer that expires the moment cryptography advances is not a trust layer. It is a delay, and we were not building a delay.

12. Building the bridge, not another tower

I began this book with two waves, the models and the chains, both rising fast, and the strange fact that almost no one is building the bridge between them. I want to end by being precise about why that bridge is the most valuable thing left to build. The models are becoming abundant, more capable and more numerous every month, increasingly a commodity. The chains are maturing into reliable, immutable foundations. What remains scarce, and will stay scarce, is the layer that lets a model's output cross onto the chain as something that means anything: sealed, anchored, authority-bound, admissible. Everyone is building taller towers on each side. The bridge is empty, and the bridge is where the value pools.

That bridge is what proof of intelligent action is. It is not a model and it is not a ledger. It is the trust layer between them, the discipline that turns a fleeting, copyable output into a durable, verifiable, accountable artefact. It takes the intelligence on one side and the immutability on the other and binds them so that the cleverness of the model finally inherits the permanence and witness of the chain. Without it the model is brilliant and forgettable and the chain is permanent and empty. With it, an intelligent decision becomes, for the first time, something you can put your full weight on.

I have built our version of this bridge into the substrate, the way I have argued throughout that it must be, because anything else is a tower pretending to be a bridge. The Sovereign Intelligence Operating System seals every consequential action into a post-quantum Open Audit Record, anchors it through Pantheon to Bitcoin, and binds it to the authority under which it was taken, with fifty specialised brains running on the operator's own hardware, fully offline-capable. I describe it not to sell it within these pages but because I will not theorise about a bridge I have not walked across. The architecture is the argument, and the argument is that this had to be done from the foundation up.

> **The model is the cleverness. The chain is the permanence. The trust layer is what lets one inherit the other.**

So here is where I leave you. If you build models, the most important work left is no longer making them speak more impressively. It is making what they say admissible, and that work happens after the model and below the application, in the trust layer. If you build chains, the most valuable thing you can anchor is not another token but the sealed proof of a decision that mattered. And if you buy, regulate or depend on these systems, carry into every room the one question the demonstration is built to avoid. Not what can it do, but what can it prove. Build the bridge. Everyone else is still building towers,

and towers, however tall, connect to nothing until someone builds what makes a model's output admissible. That is the whole of the work, and it is worth doing right.

The Mickai pantheon.

APPENDIX · ABOUT THE AUTHOR

Micky Irons

Founder and chief executive of Mickai LTD (Companies House 17166618, registered office 20 Wenlock Road, London, N1 7GU) and named inventor on the Mickai SIOS patent corpus: 101 filed UK patent applications, around 2,234 claims. Trade mark Mickai registered at UK00004373277.

Profiles

mickai.co.uk

crunchbase.com/person/micky-irons

linkedin.com/in/mickyirons

References and further reading

· European Commission, Regulation (EU) 2024/1689 (the EU Artificial Intelligence Act), official text and the high-risk-system documentation and record-keeping obligations.

· National Institute of Standards and Technology, FIPS 204: Module-Lattice-Based Digital Signature Standard (ML-DSA), 2024, the basis for post-quantum sealing.

· S. Nakamoto, Bitcoin: A Peer-to-Peer Electronic Cash System, 2008, on the immutability and proof-of-work timeline that makes a ledger a credible witness for anchoring.

· S. Haber and W. S. Stornetta, How to Time-Stamp a Digital Document, Journal of Cryptology, 1991, the foundational work on cryptographic timestamping and anchoring digests into an append-only history.

· National Institute of Standards and Technology, AI Risk Management Framework (AI RMF 1.0), 2023, on accountability, traceability and the governance of automated decisions.

· D. K. Mulligan and K. A. Bamberger, and related scholarship on algorithmic accountability and the right to meaningful explanation and verification of automated decisions.