

the coordination contract

a kernel, a law, an engine
and a ledger

an institution with no offices: people change its law by interpreted voice, a pinned set of models decides through recorded determinations, the market is free of politics, and justice reaches everyone.

jacob valdez

3 october 2026 • version 0.5

design and reference implementation. not a deployed institution, audited protocol, securities offering or claim of legal sovereignty. all people, cases and amounts in examples are synthetic.

abstract

the coordination contract is a voluntary institution for commitments between people, organizations and agents. it consists of four things: a **kernel** that never changes, a **law** that changes only by adopted rfc, an **engine** — a published, pinned set of models — that also changes only by adopted rfc, and a **ledger** of accounts, agreements, cases and determinations. the contract acts in one way only: through **determinations**, closed questions over committed inputs put to the engine tier by tier and recorded with every seat's answer. there are no branches, executives, officers, councils, juries or human adjudicators, and no key whose holder can change the contract outside an adopted rfc.

people are a predicate, not a fixed category. humans are people permanently; the law may recognize further kinds — uploaded humans, artificial agents, animals through representatives, organizations — each with its own uniqueness rule, so that one person has exactly one voice. wealth, stake and intelligence add nothing to it.

an rfc pins the engine and the contract it began under and accepts responses of any admitted modality, each adopted by its author through a receipt bound to its exact content. the pinned engine interprets each response against each option's actual changes into a score from -2 to $+2$; each interpretation is shown to its author and is contestable for its own full window. the tally is score voting: an option's score is the sum of every voice's score for it, so adding or duplicating options cannot move another option's score. a change is adopted when its score is positive and uniquely highest and the engine concurs with confidence; a divided or uncertain engine does not concur. the engine may withhold concurrence only for a kernel violation, a mismatch between the interpreted voice and the outcome, or a change outside what law may do. there is no quorum and no supermajority. replacing the engine requires the outgoing engine to concur first and then the proposed engine, which holds no credential and reads no private record before adoption. a person may also answer anonymously: a zero-knowledge proof of membership in the tree of identity commitments the rfc pinned at opening speaks for them, its nullifier is their one voice in that rfc, and a person in the pinned tree answers only that way, so nobody — the platform included — learns which response is whose and nobody counts twice.

politics has no power over the market: law may not set prices, single out holders, freeze or seize holdings, or forbid exchanges between consenting parties, except to prohibit harm to a person who did not consent. justice does reach the market. a claim is justiciable when it names a harmed claimant, a respondent, an act, a duty from an accepted agreement, from law in force at the time, or from the kernel, and resulting harm to a protected interest. cases — civil and criminal — are screened for repetition, served before their clock starts, and decided element by element on one fixed panel, with the claimant's and the respondent's figures for each harm compared and every remedy separately justified; a timely appeal is heard by a top-tier panel that took no part, and an appeal recorded in time cannot be overtaken by any clock. there is no human jury. a final judgment executes against the respondent's holdings, bonds included, in one atomic step. a victim whose judgment cannot be collected holds a claim on a protection pool funded by premiums on bond issuance, paid in equal proportion with every other unpaid victim across allocation epochs, and the pool takes over the claim. the contract services credit itself: an investor's mandate, a borrower's facility, its bond, its servicing and its default all run through the same rules.

the engine has four tiers: deterministic checks; an open-weight embedding model with pinned int8 mlp heads that anyone can rerun bit for bit; a panel of fast classifier seats; and a panel of large deliberative models whose answer is final or unresolved — never a winner by label order. a panel is exactly a quorum of independent groups, and the top tier holds at least two disjoint panels so that every top-tier decision stays appealable. every model tier must contain seats from at least two operators and at least one open-weight seat pinned by hash; seats of one operator or lineage count once. media reaches the deciding

seats only as text rendered by at least two independent renderers, and every seat reads one identical record.

the record is built to be checked without trusting whoever writes it. a seat may name an attestor whose signature its every answer must carry; every act is appended to a public hash-chained log whose heads are anchored on chain; and a replica recomputes every derived outcome from the public record and says exactly what it cannot see. on chain, one registry is the only authority: the escrow, a notary of command authority and mirrors on other chains all read its seats and keep no signer of their own.

the contract core — kernel, people, law, engine, mlp heads, determinations, evidence, rfcs, anonymous voice, justice, pool, credit, attestations, the log and state transitions — is implemented in this repository as a dependency-light typescript module, and the hosted service runs rfcs, cases and credit on it with every money movement committed as one claimed, conditional batch. the on-chain programs and the evm mirror are built and tested and not deployed. every contract number in this paper is generated by the core. no rfc or case has been decided by a live engine, no seat is attested by anyone but the platform yet, and legal enforceability and measured model accuracy are not claimed.

the record every act leaves

which question was asked, over which committed inputs? which seats answered what, and how did their answers aggregate? which kernel article, law version and engine hash governed it? what was executed, and what remains owed?

reading guide and status

the core. chapter 3 defines the four parts. chapter 4 is the kernel, verbatim, with its hash. chapter 5 covers people and law, chapter 6 rfcs, chapter 7 the engine and determinations, chapter 8 justice, and chapter 10 bonds and the protection pool. appendix D works through every rule with numbers generated by the implementation.

the context. chapters 1–2 motivate the institution. chapters 9 and 11–15 cover evidence, the economic graph, treasury, capital, measurement and execution; most of that material carries over from version 0.3. chapter 16 covers the record anyone can check — attestors, the log, checkpoints, the replica — and chapter 17 the one trust root on every chain. chapter 18 states what exists and what does not, and sets every threat against what is built; chapter 19 how the engine would be evaluated; appendix N the one place a live model answers. a short edition of this paper states the contract in about ten pages.

what changed in 0.5. the kernel is unchanged (the same hash as 0.4.1). version 0.5 adds anonymous voice (section 6.3), seat attestors (section 16.1), the transparency log and its checkpoints (sections 16.2 and 16.3), the replica (section 16.4), the registry-gated command notary and the evm mirror (chapter 17); section 18.3 lists them.

what changed in 0.4.1. an external review of version 0.4 found ordinary code paths that contradicted its guarantees: a tie at the top tier decided by label order, appeals the genesis engine could not hear, image evidence that made the engine unable to decide, seats that could receive different evidence, credentials that a proposed engine could steer, timely appeals a late sweep could overtake, remedies that multiplied, non-atomic execution, a first-come pool, and no implemented financing cycle. version 0.4.1 changes the kernel (a new hash) and the implementation to close each of them; section 18.4 lists them with what replaced them.

what changed from 0.3. version 0.3 specified a profile with eleven offices, binding votes with quorum and supermajority thresholds, consultations, a three-reviewer constitutional closure, human-only democratic feedback as an immutable rule, absolutely non-seizable bonds, and vacatur of adverse decisions whenever an appeal panel was unavailable. version 0.4 removes the offices and thresholds, makes people an extensible predicate, lets justice reach bonds while forbidding politics from touching them, adds the protection pool, makes the engine part of the contract, and makes an unhearable appeal leave the standing judgment final. appendix K separates the author’s commitments from the design choices made for this version.

authorship. concept and commitments: jacob valdez. design of versions 0.4 to 0.5, the reference implementation, editing and typesetting used ai assistance (claude). all people, operators, cases and amounts in examples are synthetic.

contents

abstract	i
reading guide and status	iii
1 the institutional coordination problem	1
2 startup finance and the continuity of missions	4
3 definition, boundary and lineage	7
4 the kernel	10
5 people and law	14
6 rfcs: interpreted voice	17
7 the engine	23
8 justice	27
9 evidence without an official truth oracle	33
10 bonds, the protection pool and external notices	36
11 entities, contracts, and the economic graph	39
12 treasury, debt, and sources of repayment	42
13 preferential capital and thematic credit coordination	45
14 economic measurement and bounded stabilization	49
15 agents, settlement and operational continuity	52
16 fault tolerance: a record anyone can check	56
17 one trust root, many chains	59
18 implementation status, release and migration	61
19 evaluating the engine, and the limits	68
A records	70
B transitions	73
C genesis law and engine	75
D worked cases	78
E mandate-constrained allocation mathematics	82
F cash flows and loss allocation	84
G economic measurements and bulletin schema	85
H threats and controls	87
I implementation status	90
J reference tests	94
K commitments and decisions	99
L glossary	101
M sources and build	103
N the live canary	104
references	106

1 the institutional coordination problem

1.1 start with the people making commitments

a founder wants to purchase equipment and hire two collaborators. a lender wants to know whether the resulting business can repay. an employee wants to understand the difference between a headline valuation and an exercisable ownership claim. a customer wants delivery rather than a convincing demonstration. a citizen wants to know which rules will apply after committing savings or changing work. none of these needs is satisfied by a faster payment alone.

consider a small manufacturing project. its identity is recorded in a corporate registry; its money sits at a financial institution; its product requirements live in documents; its authority is split across employment agreements and account permissions; its obligations are scattered through invoices, supplier contracts, and conversations. an agent might operate each application competently while still lacking a reliable answer to a basic question: *may this entity make this commitment, and what follows if it cannot keep it?*

the coordination problem is therefore partly computational and partly institutional. institutions structure interaction through formal and informal constraints; north's account emphasizes their role in exchange and uncertainty, rather than reducing them to software or individual leaders.[1] the reference design does not attempt to encode all informal norms. it asks which consequential commitments can become sufficiently explicit to reduce repeated reconstruction of authority, obligations, and evidence.

definition: coordination cost

for a specified agreement, coordination cost is the resource and human burden of discovery, verification, negotiation, authorization, financing, monitoring, dispute resolution, and settlement. it includes waiting time, administrative labor, uncertainty, accessibility barriers, and the cost of correcting mistakes. it is not identical to transaction fees, and minimizing it does not justify removing consent or due process.

1.2 a taxonomy of institutional mismatch

fragmented state. a shared vocabulary does not ensure a shared record. two systems can disagree about ownership, authority, available cash, or completion. reconciliation is a substantive task, not merely an interface problem. the bis's 2023 discussion of programmable financial infrastructure describes related frictions between messaging, reconciliation, and settlement; its proposed institutional architecture differs from the one examined here.[2]

uncertain change. rules can be changed after a participant has made an irreversible investment. some changes reflect necessary adaptation; others may be poorly announced, ambiguous in scope, or unexpectedly retroactive. the design problem is to distinguish legitimate adaptability from avoidable surprise. a participant needs an identifiable effective date, the governing version, a transition rule, and a remedy when procedure is violated.

weakly specified forward guidance. existing institutions do provide guidance and consultation; it would be inaccurate to motivate this project by asserting their universal absence. the federal reserve describes forward guidance as communication about the likely future course of monetary policy.[3] oecd work documents both existing consultation practices and remaining problems with inclusion and

feedback.[4] the proposed additional capability is an inspectable relationship between announced intentions, binding activation procedures, and executable consequences, not the invention of institutional communication.

personal discretion and succession. an institution may depend on a small number of people remembering commitments, interpreting permissions, or choosing when to act. this can support contextual judgment but also creates succession, conflict-of-interest, availability, and accountability risks. replacing an official with software removes neither judgment nor power automatically: those powers may move to maintainers, model vendors, interface operators, or upgrade-key holders.

delayed remedy. a person may discover an error promptly yet be unable to obtain a timely correction. making execution faster can worsen this asymmetry if it accelerates adverse decisions without accelerating review. the design therefore treats appeal access and bounded remedies as part of execution, rather than as an optional support channel.

horizon mismatch. a project can require years of continuity while funding, personnel, or policy commitments last months. promises to be supportive are not the same as funded obligations. conversely, indefinite commitments can preserve unproductive arrangements and prevent adaptation. an institution needs explicit horizons, staged review, and an orderly way to discontinue a failing program.

1.3 predictability without pretending that circumstances stop changing

the economic-policy-uncertainty literature supplies empirical methods for studying uncertainty, including newspaper-based indices; it does not establish that any one governance architecture eliminates that uncertainty.[5] this paper separates three quantities that are often conflated: uncertainty about *rules*, uncertainty about *observations*, and uncertainty about *outcomes*. a contract can reduce the first by publishing a rule and its amendment procedure. it cannot make a supplier reliable, an economic model complete, or future revenue certain.

an institutional commitment can be represented as

$$\mathcal{K} = (\text{scope, authority, rule version, conditions, horizon, remedy}). \quad (1.1)$$

the commitment is stronger when these fields can be inspected and contested, and weaker when a hidden administrative path can override them. a scheduled change and a state-contingent change are different instruments. “the fee becomes 1% on a specified date” differs from “the fee may change within a bounded range when a published condition persists.” both differ from a forecast of what officials expect to do.

forward guidance also has costs. the federal reserve’s discussion of the tool identifies a tension between credibility and flexibility when circumstances change.[6] the analogous protocol design publishes escape clauses before they are needed, defines their narrow authority, and records when they are used. it does not promise that emergencies will never occur or that an unchangeable formula is always preferable.

1.4 incident classes rather than a catalogue of political grievances

the motivating incidents can be tested without assuming malicious motives. a subsidy changes after equipment is ordered; an authorized signer departs before payment; a funding tranche is announced but never irrevocably committed; an interface displays an obsolete law; an automated decision uses stale evidence; a treasury guarantee proves broader than voters understood. these are *hypothetical failure scenarios*, not allegations about a named government or company.

each scenario yields a technical requirement. commitments need effective dates and version references. authorization needs expiry and succession. funding needs a distinction between a soft intention and

an escrowed obligation. interfaces need freshness proofs. consequential decisions need evidence snapshots and correction procedures. guarantees need notional caps, triggers, loss allocation, and public accounting.

this framing preserves an important symmetry: the proposed system must be exposed to the same scrutiny it applies to existing institutions. a last-minute model update, unexplained credential revocation, rushed emergency proposal, or unilateral software migration would reproduce the very uncertainty the project intends to address.

1.5 why agents intensify the problem

autonomous execution increases the number and speed of actions that can be attempted under ambiguous authority. a person can supervise a few purchases manually; an organization delegating thousands of decisions needs machine-checkable boundaries. the relevant distinction is not human versus machine competence in the abstract, but *who supplied the objective, who authorized the action, which constraints apply, and who bears the consequence*.

the coordination contract is motivated by this authorization gap. it makes no claim that every agent is reliable or that general-purpose applications will disappear. command agi’s interfaces, including computers, communications, and physical-system adapters, supply capabilities. the institution supplies the permissible relationship between intent and consequential action. these functions are complementary but separable.

the central hypothesis
explicit, inspectable institutions may reduce coordination cost and make some durable projects easier to finance. they may also impose reporting burdens, concentrate power, and expose participants to shared software failures. superiority to existing arrangements is an empirical question; the paper specifies what would have to be measured.

2 startup finance and the continuity of missions

2.1 what the startup observation adds

the investor commentary supplied as motivation describes founders leaving ventures for larger laboratories, selective team transfers, concern for employees' option economics, and difficulty distinguishing many small sources of product defensibility. the passage is treated as a supplied investor perspective, not a representative survey or evidence that mobility is inherently harmful.[7]

the institutional issue is narrower and more general: a venture's public mission, contractual obligations, productive assets, team membership, and economic rewards can cease to move together. a founder may rationally change jobs; an investor may rationally pursue liquidity; a customer may still require support; an employee may remain exposed to a claim whose practical value they cannot assess. a system that treats the founder's enthusiasm as the only continuity mechanism leaves these contingencies underspecified.

mission continuity is not personnel captivity. the proposed response attaches commitments to funded programs, intellectual property, contracts, and transferable operating responsibilities, not an obligation for a person to remain enthusiastic or employed indefinitely. a person may leave. the institution should make visible what then happens to outstanding promises, customer support, funds, and other participants' claims.

2.2 make the commitment structure an economic object

a *mission record* contains a stated purpose, beneficiaries, time horizon, eligible activities, funding sources, measurable milestones, known uncertainties, stewardship roles, and an amendment or termination procedure. it can belong to a private project, a thematic fund, or a publicly authorized program. it does not confer political authority, prove social value, or guarantee a return.

separating the mission record from the operating company permits a useful distinction. a company is an accountable executor with assets and liabilities; a mission is an objective to which participants may voluntarily commit resources. one company may execute several missions, and a mission may outlive a particular company when existing contracts and rights permit succession. any transfer must respect intellectual property, employee rights, creditor claims, privacy, and external law. copying a database is not a lawful transfer of a business.

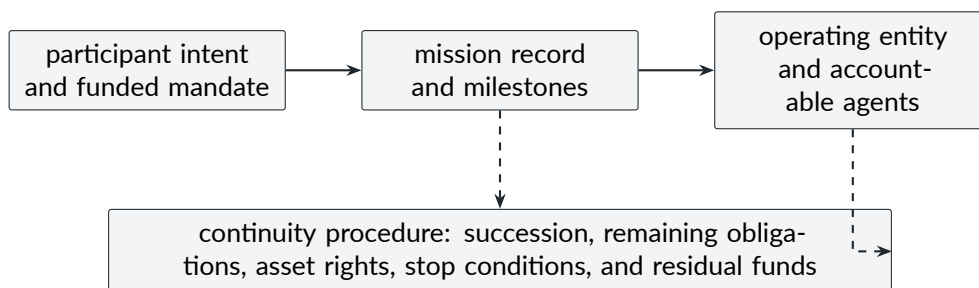


figure 2.1: mission continuity is a relationship among mandates, commitments, and operators. it is not ownership of the people doing the work.

2.3 funding fit matters more than the label on a token

not every valuable project is debt-financeable. a borrower with contracted receivables or equipment-backed operations may have a plausible repayment source. a pre-revenue research effort with highly uncertain technical success may require equity-like risk sharing, grants, sponsorship, advance purchase commitments, or philanthropic capital instead. the protocol cannot turn long-horizon uncertainty into low-risk fixed income by describing it with an embedding.

three financing states must remain distinct. an *expression of interest* is revocable unless otherwise contracted. a *conditional commitment* specifies funding conditions and the consequences of not satisfying them. a *funded commitment* has identified assets reserved under explicit release and return rules. interfaces should not display all three as equivalent runway. published milestones should distinguish verification of a completed deliverable from a forecast that the next one will be completed.

staged finance can reduce capital at risk, but milestones can also be manipulated or become inappropriate as research evolves. a reference mandate therefore permits an explicitly bounded amendment process rather than silent goal substitution. a proposed change exposes its effect on mission, risk, use of proceeds, and affected claims. investors can consent where their contract requires consent; they cannot use a private financing amendment to rewrite the contract's law.

2.4 employees, contributors, and asymmetric exit

an ownership percentage is not a complete description of employee compensation. a useful claim record includes vesting, exercise price, exercise window, transfer restrictions, liquidation priority, dilution scenarios, and the difference between preferred and common claims. these disclosures help people understand an agreement; they do not alter tax law.

the supplied commentary refers to "409as" and early exercise. the technically relevant distinction is between a financing valuation, the fair market value of the applicable common stock, the option's contractual exercise price, and possible tax consequences. an existing option's exercise price does not automatically reset upward just because a later valuation rises. u.s. tax treatment varies by option type and transaction; irs guidance distinguishes statutory and nonstatutory options, and section 1.409a-1 contains relevant stock-right valuation conditions.[8, 9] the paper therefore treats affordability and legibility as design concerns, not as a claim that a protocol can make exercise tax-free or universally affordable.

selective team transfers motivate a *change-of-control and continuity disclosure*: which contracts continue, which intellectual property moves, which workers receive offers, what happens to remaining employees, and how existing securityholders are treated. the protocol cannot require a buyer to employ everyone. it can require that representations made within its own financing process match the actual rights being sold and can expose a mismatch before another person joins.

2.5 defensibility that can be inspected without publishing every secret

a product may depend on many small operational advantages rather than a single patent or model. a financing interface could make some of these advantages inspectable: retained customers, independently confirmed delivery, permissioned reliability records, recurring contracts, documented supplier relationships, or a succession plan. these are evidence categories, not a universal moat score.

more transparency is not monotonically better. publication of customer lists, internal failure traces, pricing, or employee data can destroy commercial value and personal privacy. the design uses selective disclosure and independently attested summaries. a lender may verify the existence and concentration

of receivables without receiving a public index of every customer's identity. a rival should not obtain another company's trade secrets merely by registering as an investor.

there is also a feedback hazard: making existing activity easier to verify can disproportionately favor incumbents with long histories. first-time founders, novel research, and work whose value is not captured in transactions need an explicit path to consideration. that path may require higher-risk or concessionary capital. disguising such support as fully commercial low-risk lending would obscure who bears the cost.

2.6 the cosmic imperative as motivation, not compulsory doctrine

the author connects this financing problem to what he calls a *cosmic imperative*. the term's complete philosophical content is not specified here. the operational connection used in this paper is limited: commitments to work with very long horizons need institutions that can preserve knowledge, resources, and human agency through changes in personnel, technology, and incentives.

this motivation does not require every participant to adopt the same long-term objective. an investor interested in manufacturing, a participant funding care services, and a researcher pursuing foundational knowledge may use the same coordination infrastructure under different mandates. the protocol's task is to make those commitments intelligible and accountable, not to collapse them into the founder's purpose.

nor does an ambitious mission override present people. a claimed future benefit cannot erase a person's right to understand a contract, refuse work, protect private information, or challenge a harmful decision. the paper evaluates durable mission funding alongside opportunity cost, concentration, effects on outsiders, and the possibility that a mission should stop. a contract whose founder could invoke an undefined higher imperative to bypass its procedures would contradict kernel article k3: nobody holds authority over the contract, its founder included.

2.7 what would demonstrate improvement

the relevant outcomes are continuity of contracted service after personnel changes, predictability of committed funding, intelligibility of employee claims, time to replacement stewardship, and the fraction of completed work that remains usable after an organization changes direction. raw founder retention is not an adequate success metric: retaining an unwilling person may reduce welfare while preserving an attractive statistic.

these outcomes connect startup finance to the larger institutional thesis. a durable economy needs more than efficient formation of companies; it needs understandable ways for commitments to survive, change, transfer, or terminate. thematic credit is one instrument within that process, not the universal answer to mission financing.

3 definition, boundary and lineage

3.1 the coordination contract

definition

the coordination contract is a voluntary institution made of four parts. the **kernel** is a fixed text and a set of numbers enforced by code; it never changes. the **law** holds the kinds of people the contract recognizes, the duties it enforces and its parameters; it changes only by adopted rfc. the **engine** is the pinned set of models through which the contract decides; it is part of the contract and changes only by adopted rfc. the **ledger** holds accounts, agreements, rfcs, responses, cases, judgments and determinations. the contract acts only through **determinations**.

“contract” names a shared commitment, not one large smart contract. the implementation combines a pure reference module, a hosted service, on-chain records and external legal arrangements. program validity, constitutional validity, solvency and external enforceability are different properties; demonstrating one does not establish another.

3.2 determinations are the only act

a determination is a closed question — a prompt and a fixed set of answer labels — over a canonical evidence package, put to the engine tier by tier until a tier answers decisively (chapter 7). the package is text only, each item under a unique identifier; media reaches it as renderings (section 9.3). its record holds the question, the package’s digest, the engine hash, every round (which seats sat, what each answered, the digest of what each seat was actually sent, how the answers aggregated) and the outcome, and is identified by the hash of all of that. anyone holding the record can check the aggregation; anyone holding the package and the open-weight seats can rerun the deterministic tiers.

an outcome is one of three things. **decided**: a tier answered with a unique label at its confidence band. **unresolved**: the top tier answered, but its groups were tied or its confidence fell below its band. **unavailable**: no tier could seat a panel that answered. finality means the procedure ended, not that some label had to win; each use of a determination states what an unresolved or unavailable outcome means for it.

every consequential act is a determination or a deterministic consequence of one:

- interpreting an rfc response into a score for each option (section 6.4);
- concurring or withholding concurrence in an rfc’s adoption (section 6.6);
- screening a filing for a matter already brought; deciding justiciability, each element of the duty, which party’s figure is closer to each proved loss, whether each remedy is justified, and an interim freeze (chapter 8);
- deciding whether a service agreement’s terms were met.

adoption of an rfc, a judgment’s remedies, an execution plan, a pool allocation and a facility’s servicing follow from those determinations by pure functions with integer arithmetic.

3.3 no offices

the predecessor and version 0.3 of this paper assigned powers to offices: branches signing amendments, prosecutors choosing cases, registrars, enforcers, closure panels. each office was a place where a person, a key or a model could hold power that the procedure did not fully constrain. version 0.4 has none (kernel k3).

- *who decides?* the engine's seats, answering a published question, aggregated by a published rule.
- *who files?* anyone; the contract itself files when its own records show a breach.
- *who executes?* the ledger, applying a final judgment's execution plan; the settlement program, applying a signed determination.
- *who changes the rules?* the people's voice, with the engine's limited concurrence.

independence is no longer separation between people holding offices. it is independence between engine seats: seats of one operator or lineage count once, every model tier must include two operators and an open-weight seat, the top tier must hold two disjoint panels, and a seat never sits in a matter that its operator, or an entity the engine records as controlling it, is a party to.

3.4 four kinds of standing

personhood gives a voice in rfcs (section 5.1). **contractual capacity** is the ability to accept agreements and bear their duties. **protected interests** are what justice protects, and belong to every being with interests, person or not (kernel k1, k2, k6). **delegated capability** is permission to act for a principal. none implies another. an organization can accept agreements without being a person; an animal's interests are protected whether or not the law recognizes animals as people; an agent can hold a capability without a voice; a person can have a voice without a single agreement.

3.5 trust and its perimeter

the design does not promise a trustless factual world. a signature identifies an authorized statement; it does not establish that a camera shows the relevant factory. a personhood proof establishes a key under an issuer's rule; it does not establish that the issuer never errs. a final chain transfer does not establish rightful ownership of a physical asset. a determination establishes what the engine answered; it does not establish that the answer is true. claims of improvement must measure these dependencies rather than count the people removed from a workflow.

3.6 lineage

3.7 retain, change and retire

retain: accounts, signed frames and certificates, evidence commitments, the evaluator's typed leaves and deterministic aggregation, confidence-weighted quorums, juror independence checks, execution receipts, capability attenuation and accounting.

change: the evaluator becomes the engine and is pinned in the contract rather than in each agreement; the committee that signs settlement frames becomes the engine's seats, recorded in the registry; the appeal becomes a fresh top-tier panel rather than a rerun of the same evaluator; the grey zone climbs tiers instead of ending at a human administrator.

period	record	status in this paper
june 2026	author's coordination/trust predecessor	author report
11 july 2026	certificate-state-machine devnet escrow	repository runbook[33]
16 july 2026	schedule-over-signed-frames escrow	repository runbook; unaudited
30 september 2026	version 0.2: static review of the predecessor	pinned commit
1 october 2026	version 0.3: rp-03 profile with offices and thresholds	superseded
1 october 2026	version 0.4: kernel, law, engine, ledger; reference module	superseded by 0.4.1
2 october 2026	version 0.4.1: the hardening pass; a new kernel	implemented in this repository; hosted service only

table 3.1: implementation lineage.

retire: the four branches (command, law, reserves, rights) and their seats, signatures and vetoes; the stake-weighted agent franchise — `franchiseWeight` gave a bonded agent up to ten votes by stake[30]; quorum and supermajority thresholds; stakeholder-class ratification; administrator dispute resolution; the upgrade multisig of named offices. legacy records stay intact as history and are not relabeled as conforming.

3.8 one lineage

the contract keeps one canonical history. copying its code does not transfer its ledger, people or obligations. a different kernel defines a different contract with no claim on this one's ledger (kernel k4).

4 the kernel

the kernel is the part of the contract that never changes. its twelve articles are reproduced below exactly as they are hashed. a different text, a different floor or a different list of protected interests produces a different hash, and a different hash is a different contract with no claim on this one's ledger.

4.1 the articles

k1. truth and justice The contract serves truth and justice. Truth: fidelity to evidence, honest uncertainty, attribution, and correction of known error. Justice: protection of every being's protected interests against wrongful interference, by anyone, including the contract itself. No gain to the contract's revenue, size or survival outweighs either.

k2. people and voice Protection belongs to every being with interests; voice belongs to persons. A person is an entity of a kind the law recognizes as people, holding a valid personhood proof for that kind. Human is a kind of people and always will be. The law may recognize further kinds by adopted RFC. Each kind has one uniqueness rule, and every issuer of that kind's proofs derives the same key for the same individual, so that one person cannot speak as many. Each person has exactly one voice in every RFC, given by their own deliberate act. Wealth, stake, activity, intelligence and position add nothing to a voice.

k3. no offices No person or entity holds authority over the contract. The contract acts only through determinations: a pinned question, committed inputs, the engine's seats, a deterministic aggregation and a published record. There are no branches, executives, officers, councils, juries or human adjudicators, and no key whose holder can change the contract's code, law or engine outside an adopted RFC.

k4. change only by voice Law and engine change only by an adopted RFC. An RFC is adopted when the people's voice, interpreted response by response against the change itself and aggregated with one voice per person, favors a change over the status quo, and the engine concurs with confidence. The engine may withhold concurrence only because the change would violate this kernel, because the voice does not in fact favor it, or because it lies outside what law may do; a divided or uncertain engine does not concur. The engine never adopts what the voice does not favor. An RFC that replaces the engine is decided by the outgoing and the proposed engine together, and both must concur; the proposed engine is consulted only after the outgoing one concurs, and holds no credential or record before adoption. If the outgoing engine cannot answer at all, the proposed engine's concurrence suffices after a further window as long as the RFC's own. A change to the contract's code is a change to the engine. This kernel is never changed; a different kernel is a different contract.

k5. a free market Politics has no power over the market. Law may not set prices, tax or single out particular holders, freeze or seize holdings, or restrict an exchange between consenting parties, except to prohibit harm to a protected interest of a being who did not consent. Every duty the law creates names the protected interest it protects, the harm it forbids and the elements that make a breach, and binds everyone alike.

k6. the scope of justice A claim is justiciable when it names a claimant who was harmed (or anyone filing for a being who cannot), a respondent who was served, an act or omission, a duty the respondent owed — from an agreement the respondent accepted, from law in force at the time of the act, or from this kernel — and harm to a protected interest or an agreed term caused by that act. One matter is one case. Justice reaches the market: a final judgment executes against the respondent's holdings, bonds included, for the harm established and its costs, once per harm, and nothing more.

k7. due process without juries Every case gives notice, time to answer, an opportunity to submit evidence, a reasoned determination of each element of the duty, and an appeal to seats that took no part in it; the engine keeps enough independent capacity for that appeal. A timely appeal is heard. Determinations are made by the engine's seats, tier by tier, on one identical record; a tie or an answer below the tier's confidence is unresolved, never a winner by order. No human jury or human adjudicator decides any case, civil or criminal. A claimant who cannot establish an element loses the claim; a record the contract could not process is not a loss on the merits. A criminal finding requires near-certainty on every element at the top tier. A seat never judges a case in which its operator is a party.

k8. bounded remedies Remedies are restitution, compensation, agreed performance, correction of records, and — on a criminal finding — a penalty of at most three times the proved harm in total, paid to the victim and the protection pool, and exclusion from the contract's services for at most a year. Each remedy answers a harm established in the case, and no harm is repaired twice. No remedy touches a body, compels labor, or punishes without a finding. An interim freeze holds at most the amount claimed, for at most fourteen days, and only on a determination that the claim is likely to succeed and the amount likely to vanish.

k9. victims are made whole A victim whose final judgment cannot be collected from the respondent is paid from the protection pool, within the limits the law sets and in equal proportion with every other unpaid victim, and the pool takes over the claim. What the pool cannot yet pay stays owed and is paid as the pool is funded. The pool is funded by premiums on bond issuance and by penalties, never by seizing anyone's holdings outside a judgment.

k10. the engine The engine is part of the contract: its tiers, seats, pinned models, renderers and rules are published and change only by RFC. Every tier that holds model seats has seats from at least two independent operators and at least one seat whose weights are open and pinned by hash, so that no provider can change or stop the contract by changing or retiring a model. Seats of one operator or lineage count as one. Media evidence reaches the seats as text rendered by at least two independent renderers, recorded and open to the parties. A matter is decided by the engine and procedure in force when it began.

k11. records Every determination, RFC, response, case and transfer is recorded with what it relied on. Corrections are appended; nothing is erased. Judgments and RFC records are public; a case's evidence is disclosed only to its parties and only as far as the case needs it.

k12. exit Anyone may leave. Leaving ends future obligations to the contract; duties already accrued and agreements already made survive.

4.2 the hash

the kernel hash is

$$\text{sha256}(\text{cc:kernel:v4}\backslash\text{n} \parallel \text{canon}(\{\text{articles}, \text{floors}, \text{interests}\}))$$

where *articles* is the document above with one paragraph per article, *floors* the numbers of table 4.1 in the module's units (milliseconds, per mille) and *interests* the list of protected interests, and *canon* is json with keys sorted and no whitespace. its value is

1f9944f03bef0ff7d30b2eea3b068fdd2b1432866aa28198095cc8947177ff5b

the reference module pins this value (KERNEL_HASH), and its tests fail if the text, the floors or the interests change without the pin changing with them. every state hash, rfc and determination of the contract is computed under this kernel.

4.3 floors enforced by code

the kernel's numbers are enforced where they apply: the law check refuses a law below a floor (section 5.3); the engine check refuses an engine without independent operators, an open-weight seat, two disjoint top-tier panels, or two independent renderers for every medium the law admits (section 7.4); the justice functions cap remedies and apply the criminal standard (chapter 8). law may be stricter; it may never be looser.

floor or ceiling	value
minimum window of a law rfc	7 days
minimum window of an engine rfc	21 days
minimum time to contest one's own extraction	2 days
minimum time to answer a case	3 days
minimum time to appeal	3 days
maximum interim freeze	14 days
minimum confidence for a criminal finding at the top tier	950/1000
maximum criminal penalty	3× the proved harm
maximum exclusion after a criminal finding	365 days
minimum distinct operators per model tier	2
minimum open-weight seats per model tier	1
minimum disjoint quorate panels at the top tier	2
minimum independent renderers of each media item	2

table 4.1: the kernel's numbers.

the protected interests are: life, bodily_integrity, liberty, property, privacy, agreement, voice and access. every duty of law protects at least one; every claim alleges harm to one.

4.4 why these articles

k1 puts the contract under the same standard as everyone in it: the contract's own acts can be wrongful, and its revenue or survival never outweighs a protected interest. **k2** separates protection from voice — protection belongs to every being with interests, voice to persons — separates who people are from what they own or how capable they are, and makes the kind of people extensible without letting one person become many: every issuer of a kind derives the same key for the same individual, and a voice

is given by its holder's own deliberate act. **k3** and **k4** remove every human or key-holding position from which the contract could be steered outside a recorded procedure, while keeping the people in control of change. **k4** also says what a divided engine means — no concurrence — and what happens when the outgoing engine cannot answer at all: after a further window as long as the rfc's own, the proposed engine's concurrence suffices, so a failed engine cannot freeze the contract. it makes a change of the contract's code a change of the engine. **k5** is the market's protection from politics; **k6–k9** are the market's subjection to justice: a respondent is served before anything runs against them, one matter is one case, every element is decided, a tie is unresolved and never a winner, a record the contract could not process is never a merits loss, remedies answer established harms without repairing any twice, and unpaid victims share the pool in equal proportion. **k10** keeps the engine from becoming one provider's product and fixes each matter to the engine and procedure in force when it began. **k11** and **k12** keep the record honest and the institution voluntary; **k11** makes judgments public and keeps a case's evidence with its parties.

the kernel contains no number that only experience could set well: windows, deposits, case tiers, penalty multiples below the ceiling, pool and credit parameters belong to law (appendix C).

version 0.4.1 replaced version 0.4's kernel (hash prefix 9af475d6) before any rfc or case existed under it. by **k4** the result is a different contract; since the old one held no record but its genesis, the hosted service re-publishes genesis under the new kernel rather than migrating anything.

4.5 a defect in the kernel

the kernel can be wrong. if it is, it cannot be amended: the remedy is a new contract with a new kernel, which people may join and to which they may move their own holdings by ordinary transfer. it inherits nothing by declaration. until people leave, the old contract keeps operating under its kernel, and its accrued duties and agreements survive (**k12**).

5 people and law

5.1 people are a predicate

an entity is a **person** when its kind is one the law recognizes as people and it holds a valid personhood proof for that kind. a proof names the entity, its kind, the issuer, a **uniqueness key** under the kind's rule, its validity interval and whether it was revoked. it is valid when the law accepts its issuer for that kind and it is current and unrevoked. the person's **voice** is kind:uniquenessKey.

each recognized kind carries a uniqueness rule, because a kind whose members can copy themselves would otherwise turn one person into many:

rule	one voice per	natural kinds
identity_fingerprint	document identity: first given name, family name and birth date, normalized and keyed by the contract's own secret	humans
lineage	origin lineage: copies, forks and fine-tunes share a key	artificial agents; uploaded humans
registration	legal registration	organizations
custodial_individual	attested biological individual, spoken for by a representative	animals

table 5.1: uniqueness rules in the reference module.

the genesis law recognizes one kind, human, under `identity_fingerprint`, with the platform's identity verification as issuer. a kind has one uniqueness namespace and any number of issuers inside it: for humans the contract derives the key itself, as an hmac under a secret that is never rotated, from the identity an issuer reports, so a passport and a driving licence, or two issuers, of one person land on one key. an issuer that cannot report those fields cannot issue human personhood, because a second namespace would let a person verified in both speak twice. the tolerance has a cost: two different people with the same first name, family name and birthday share one voice, and the second can bring a case for their voice interest. an issuer for another kind is named in law by its `ed25519` public key and attests personhood by signing the entity, the kind, the uniqueness key and the validity window, so a proof cannot be moved to another entity, kind or period; revoking a proof is signed the same way. an issuer's statements about one entity apply in the order of their issue times and never go back: a revocation stays recorded as a tombstone, and a replayed older issuance — still carrying a valid signature — is refused as stale, so it cannot undo the revocation. no such issuer exists yet. kernel article `k2` makes human a kind of people forever; an rfc may change the human rule's issuers but never remove the kind. recognizing another kind is an ordinary law rfc that must supply that kind's uniqueness rule and accepted issuers. the reference module's `resolveVoices` applies the rule: of several valid proofs presenting the same voice, the earliest-issued holds it and the others are refused as duplicates.

the predicate decides voice only. it does not decide whose interests count: protection belongs to every being with interests, whatever its kind and whether or not the law recognizes it (`k1`, `k2`), and anyone may file a claim for a being who cannot (`k6`). an rfc fixes the kinds of people at its opening: a kind recognized while it is open does not respond to it. an organization is made of people who already have voices; recognizing organizations as people would give their members a second voice through them, and an rfc proposing it should be judged with that in mind.

5.2 law

law has three parts: the recognized kinds of people, the duties, and the parameters. it carries a version number that increments with every adopted change, and the contract keeps every past law with the instant it took effect, so that an act is judged under the law in force when it happened (k6).

5.2.1 duties

a duty has an identifier, a title, its text, the protected interests it protects, the harm it forbids, the **elements** that make a breach, whether its breach is a criminal matter, and the instant from which it applies. the text binds everyone alike: a duty has no field naming whom it applies to, so it cannot single out particular people or holders (k5). the elements are what a case decides one by one (section 8.6): fraud, for example, has 5 — a representation of fact, its falsity, knowledge or recklessness, reliance, and the harm the reliance caused — so a liability finding cannot skip the element that distinguishes fraud from an honest mistake. the genesis law contains six duties, listed with their elements in appendix C: no fraud, no theft, no violence, no coercion (criminal), keep agreements, and no exposure of private information (civil).

5.2.2 parameters

parameters are integers: windows and response times in milliseconds, amounts in cents, rates in basis points or per mille. the genesis values are in table C.2; the kernel bounds the ones it constrains (table 4.1). besides the parameters, law lists the **media admitted as evidence** and as rfc responses — text, image, audio in the genesis law. text is always admitted. a medium the law admits must be renderable by at least two independent renderers of the engine (k10); a medium it does not admit is refused at submission with a reason, never silently dropped.

5.2.3 changes

a law change is one of: add a duty, amend a duty, repeal a duty, set a parameter, recognize a kind of people, unrecognize a kind, set the admitted media. an rfc option carries a list of changes applied in order. a repealed duty keeps governing acts done while it was in force.

5.3 the law check

before an rfc opens, and again when it is adopted, each option's changes are applied to the current law and the result is checked against the kernel: human still recognized; no duplicate kinds or duties; every duty protects at least one interest, names its harm and lists at least one element; text admitted; every parameter within its kernel bound; and the engine in force able to render every admitted medium with two independent renderers. a change that fails is refused with the code of the article it violates (for example k4.law_window or k2.human_removed). this is the structural half of kernel article k5. the semantic half — whether a duty that is harm-based in form is in fact a price control or a measure aimed at someone — is a ground on which the engine withholds concurrence (section 6.6).

5.4 what law may not do

the market is free of politics (k5)

law may not set prices, tax or single out particular holders, freeze or seize holdings, or restrict an exchange between consenting parties — except to prohibit harm to a protected interest of a being who did not consent. every duty names the protected interest it protects and the harm it forbids, and binds everyone alike.

the line is consent and harm. a sale of a dangerous good to a consenting buyer is the buyer's business; the same sale that injures a bystander is not. a cartel's agreement among its members is their business; its fraud on a customer is not. a protocol fee set by law is not a tax on particular holders: it applies to every settled transfer and funds the commons the contract itself needs. the commons may run programs adopted by rfc — grants, subsidies, insurance — with its own money; it may never compel anyone else's.

5.5 responsibility for agent-mediated acts

a principal answers for acts it authorized, adopted or was specifically obliged to supervise. a case must show the scope of the agent's capability, the actual instruction or adoption, and the applicable duty. an agent of a recognized kind may bear its own accepted duties. the model operator, tool provider, employer and principal are not interchangeable respondents. knowledge is attributed through what the responsible decision process actually received, not through a model's training exposure or an unused tool permission.

5.6 entry and exit

anyone may hold an account and accept agreements. a person enrolls by presenting a personhood proof. leaving ends future obligations to the contract; accrued duties, pending cases and accepted agreements survive (k12). leaving does not withdraw a response already counted in a closed rfc.

6 rfcs: interpreted voice

an rfc is how law and engine change. it replaces three things the predecessor used: ballots with quorum and supermajority thresholds, stakeholder-class ratification, and branch signatures. people respond in their own words, images or speech; the engine interprets each response against each option's actual effect; every voice scores every option and the scores are summed; the engine concurs within strict limits, and only with confidence.

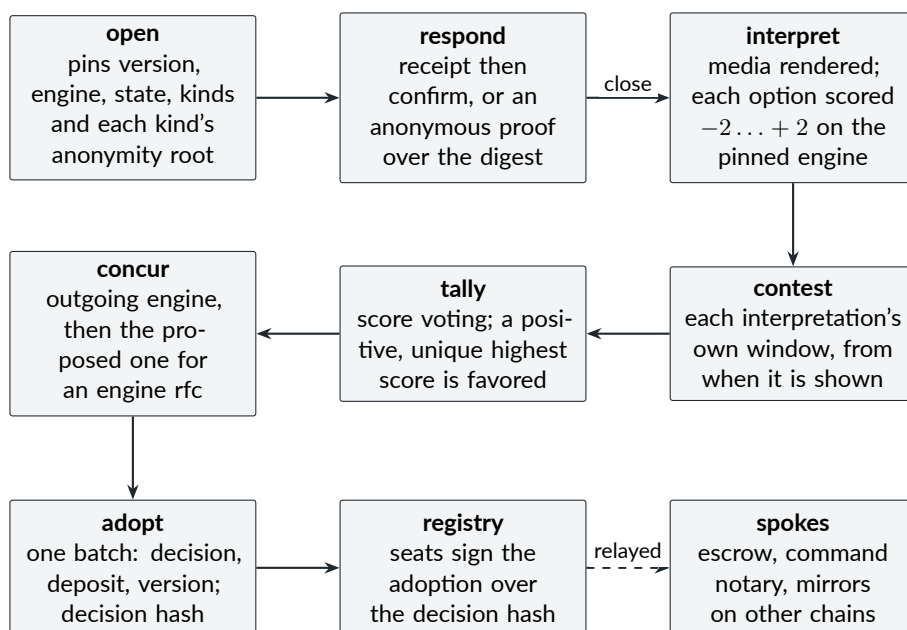


figure 6.1: the life of an rfc. every step is a determination or a recorded transition on the transparency log (section 16.2). no favored option, or a divided, withheld or unavailable concurrence, keeps the status quo.

6.1 opening an rfc

anyone — person, organization or agent — may open an rfc. a draft has a title, a body and one or more change options; all options change the same thing, law or engine. the contract adds the status quo as option zero. at opening:

- every law option is applied to the current law and checked (section 5.3); every engine option is checked against the kernel and the media the law admits (section 7.4), and an engine option equal to the engine in force is refused. a failing option refuses the draft.
- two options with the same effect are refused (rfc.duplicate_effect), whatever their titles and summaries: an option's identity for this purpose is what it changes, not how it is described.
- a draft whose options have the same effects as those of an rfc rejected within the cooldown (90 days in the genesis law) is refused.
- the window is the law's window for the target, never below the kernel floor: 14 days for law and 28 days for the engine in the genesis law (floors 7 and 21).
- the opener posts a deposit of \$10, debited in the same transaction that records the rfc. it returns to the opener if anyone other than the opener supported a change; otherwise it goes to the commons, which pays the interpretation costs.

pinning (k10). an rfc records what it began under: the contract version, the hash of the engine that will interpret and concur, the hash of the contract state its options were drafted against, and the kinds of people the law recognized at opening. its responses are interpreted by that engine and its voices resolved under those kinds, whatever is adopted while it runs. if the contract's state changes before the rfc is decided — another rfc adopted in between — the rfc is *superseded* (`adopt.superseded`): its voice was given about a contract that no longer exists, and it is not adopted. the rfc's identity is the hash of its content; a later material change is a new rfc.

6.2 responses and receipts

a response is a person's own act: their voice, the entity presenting it, its parts and its time. a part is text, or a reference to media bytes — an image or audio, as the law admits — stored once by the contract under its sha256. assistance in drafting and translation is ordinary; an agent may not originate a person's response, and an agent key speaks only for the agent.

receiving a response is not counting it. the contract answers a submission with a **receipt**: the response's identifier and the digest of its exact content, `sha256(cc: rfc-response:v4...)` over the rfc and the parts (for example `511f3b10225f271a...`). the response counts only once its author confirms that digest — the deliberate act kernel article k2 asks for, bound to the payload rather than to a session. a person may confirm a new response any number of times before close; the latest confirmed counts. an unconfirmed response, a response confirmed after close, and a response from a kind not recognized at opening do not count. the reference function `activeResponses` returns exactly one confirmed response per voice.

6.3 anonymous voice

kernel article k2 asks that each person have one voice, given by their own deliberate act. it never asks who. a person may therefore answer an rfc so that nobody — the platform included — knows which response is theirs, and still count exactly once. the reference implementation uses semaphore v4: groth16 proofs over the bn254 curve, a lean incremental merkle tree of poseidon hashes, and eddsa-poseidon identities.[10]

- **identity.** the person makes a semaphore identity in the contract editor. its secret lives only on their device and in a file they keep; it is never uploaded.
- **registration.** signed in, the person registers the identity's *commitment* for their voice: one active commitment per voice, and a commitment held by one voice only. this is the one place a voice and a commitment meet, and the platform keeps that link; the proofs below are zero-knowledge over the whole set, so the link says nothing about which response is whose. rotating the identity replaces the commitment and keeps the history.
- **the tree.** each kind's active commitments, in registration order, are the leaves of a public tree. **an rfc pins each kind's root and size when it opens**, and the pin is part of the rfc's hashed record: an rfc opened over another root is another rfc (yes in appendix D.13).
- **acts.** an anonymous act is a request carrying no credential, only a proof that some member of the pinned tree signed exactly this message in this rfc's scope. the proof's **nullifier** — the hash of the identity's secret and the scope — is the same for every proof of one identity in one rfc and unlinkable to anything else, and it *is* the voice: `anon:human:77`. messages are separated by action — respond (binding the response's digest and a fresh nonce), upload media (binding its sha256), contest, look up — so a captured proof replays only as itself (5 distinct messages for the five actions of appendix D.13). the proof is the deliberate act, so an anonymous response counts on receipt, and a later proof with the same nullifier replaces the earlier response.

6.3.1 one person, one count

a person could otherwise count twice: once signed in and once anonymously, or through a rotated identity. the rule closes both. a person whose commitment is in the tree an rfc pinned answers that rfc *only* anonymously; the signed-in route refuses them. a person without one — never registered, or registered after the rfc opened — answers with their signed-in voice. a commitment registered or rotated after opening is not in the pinned tree, so rotation can never open a second voice (appendix D.13: ann registered before opening and rotated after, and answers only anonymously — yes; cal registered after opening, and answers signed in — no for the anonymous-only rule). the cost is real: a person who loses their identity file cannot answer rfcs opened while it was registered, and registers a new one for the rfcs that follow.

6.3.2 verification

a proof acts in an rfc only when it verifies, its root is the root the rfc pinned for the kind it claims (`not_the_pinned_tree` otherwise), its scope is this rfc (`wrong_scope`), its message is the one the action requires (`wrong_message`), and the kind was recognized at opening (`kind_not_recognized_at_open`). an anonymous response may cite only media uploaded under a proof, never a url the contract would fetch (`anonymous_media_must_be_uploaded`). the hosted service verifies proofs inside its worker runtime with its own `groth16` check over the `bn254` pairing, against semaphore's verification keys for every tree depth, copied byte for byte from semaphore's own package and held to it by a test; on real proofs, forged proofs, proofs with each public input changed, and garbage, its verdicts equal `snarkjs`'s.

6.3.3 what can still link a response to a person

a proof hides a person among everyone in the pinned set; a set of one or two hides nothing, and the editor shows its size. the anonymous routes read no cookie, authorization or principal and record nothing that links a nullifier to an entity, and the editor sends to them without credentials and without a referrer. but the operator can still see network metadata — the address a request comes from — and timing; the content of a response can identify its author; and fetching the proving artifacts from semaphore's artifact host shows that host the person's address. the editor can hold a proved response for a random wait before sending it; a relay or an anonymity network is the person's choice. anonymous voices receive no notices, having no inbox; their authors look their interpretations up under a proof. the design claims unlinkability by the contract's record, not anonymity against an observer of the network.

6.4 interpretation

after close, media in the responses is rendered to text (section 9.3), and each counted response is interpreted once per change option by a determination of `kind rfc.stance` on the rfc's pinned engine. the question carries the rfc, the option's *actual changes* — the law changes or the engine, in canonical form, with the opener's summary marked as not binding — and the response, and asks for the person's stance on adopting that option instead of the status quo, judged only from what the response expresses, never from who the person is. the labels are fixed, and each maps to a score:

label	score
strongly_oppose	-2
oppose	-1
neutral	0
support	+1
strongly_support	+2
not_addressed	0

because the question has closed labels, every tier can answer it: an mlp head over the response’s embedding, a classifier seat, or a deliberative model. a response that addresses only one option of several is `not_addressed` on the others. an interpretation the engine leaves *unresolved* scores 0 and is recorded as unresolved; one it cannot process within the law’s period (14 days in the genesis law) scores 0 and is recorded as unprocessable. neither is counted as the person’s opposition or support.

contest. each interpretation is shown to its author when it is recorded, and its contest window — 3 days in the genesis law, kernel floor 2 — runs from that moment, not from the rfc’s close. an interpreter that runs late or slowly therefore never shortens anyone’s chance to inspect and contest their own interpretation: an interpretation recorded 9 days after close is contestable until day 12. a person contests their own interpretation once; the contest reruns every option’s question at the top tier on a panel that excludes every group that already sat, and the result replaces the original (if the contest cannot be heard within the law’s period, the original stands). a person cannot contest someone else’s interpretation; a factual error in someone else’s response is a matter for the voice. the rfc is decided only after every counted response is interpreted, every window has passed and every contest is resolved.

6.5 the tally: score voting

for each change option k , the score is

$$S_k = \sum_{\text{persons } p} s_{p,k}, \quad s_{p,k} \in \{-2, -1, 0, 1, 2\},$$

the sum of every voice’s score for that option. the **avored** option is the one with the highest positive score, if it is unique. a tie for the highest positive score, or no positive score, leaves the status quo. silence counts as nothing: a person who does not respond neither supports nor opposes. there is no quorum. each voice contributes at most two points to any option; wealth, stake and activity add nothing.

why score voting. version 0.4 normalized each person’s stances so that their absolute values summed to at most one unit. that made each option’s weight depend on the *set* of options: a person opposing many alternatives spread their opposition thin, while a person supporting one kept their whole unit there, so adding redundant alternatives could change the contest. under score voting a voice’s score for one option is independent of its scores for the others, so the score of an option does not change when another option is added, duplicated, reworded or reordered. the reference tests check this directly, and appendix D.1 shows it: adding a third option leaves the first two options’ scores at 3 and -1. options with identical effects are refused at opening, so a proposer cannot split support for one change across clones. the tally record — every voice’s scores per option and which interpretations were unresolved or unprocessable — is hashed, and that digest is part of what the concurrence reads and what anyone can recompute.

why no quorum

quorum thresholds answer the worry that a small group decides for everyone. they also let a disengaged majority block every change by staying home, which in practice freezes the institution and moves real power to whatever needs no vote. the contract answers the worry differently: windows of at least seven and twenty-one days with notice to every person, a kernel that no rfc can touch, the engine's confident concurrence, and the fact that any adopted change can be reversed by the next rfc.

6.6 concurrence

before a favored option is adopted, the pinned engine answers a determination of kind `rfc.concur`, starting at the top tier, with exactly two labels: `concur` and `withhold`. its package is the kernel, the rfc, the favored option's actual changes, the whole tally record, the contract as it was at opening, and a **seeded sample** of 24 responses (in the genesis law) with their interpretations. the sample is drawn by a public, deterministic order keyed by the rfc and the tally digest, so anyone holding the record draws the same sample and nobody can choose it — not the first responses in some list. the prompt allows withholding only on three grounds:

- adopting the option would violate the kernel in a way the structural check cannot see;
- the interpreted responses do not in fact favor the option — for example, a block of interpretations that read support into responses that express none;
- the option does something law may not do (section 5.4): a price control dressed as a duty, a measure aimed at named holders.

disagreeing with the people's choice is not a ground. when concurrence is withheld, a second determination over the same package classifies the ground (`kernel_violation`, `voice_mismatch` or `outside_law`) for the record; the decision does not depend on it. assent is decided separately from the classification of objections, so three different objections can never split against one assent and leave it the largest label.

an rfc is adopted when an option is favored and the concurrence is *decided* with the label `concur` — a unique answer at the top tier's band. a tied or low-confidence panel is unresolved and does not concur; an unavailable engine does not concur. **the engine can block; it can never adopt what the voice does not favor, and it concurs only when it is confident.** (appendix D.1: with an unresolved concurrence the result is null.)

6.7 engine rfcs

an rfc whose option replaces the engine is decided by two concurrence determinations over the same package, in order. the outgoing (pinned) engine is asked first. only if it concurs is the proposed engine asked, and it is asked through a **candidate executor**: it reads only the public rfc record, renders no media, and its seats' credentials are bound to their operators' own hosts like every seat's (section 7.9). evaluating a possible successor therefore never grants it the successor's powers. both must concur.

this prevents an incumbent engine from entrenching itself (it cannot adopt alone, and it can stop a replacement the voice favors only on the three grounds) and prevents a captured replacement from installing itself (it cannot adopt alone either). an outgoing engine that has *failed* must not freeze the contract, so kernel article k4 adds one fallback: if the outgoing engine cannot seat a panel at all for a further window as long as the rfc's own, the proposed engine's concurrence suffices. the fallback applies to unavailability only — an outgoing engine that answers `withhold`, or is divided, has answered.

appendix D.2 shows each path. on chain, an engine change and a change of the contract's code both need the outgoing and the proposed engine's signatures (section 15.4).

6.8 adoption

an adopted option is applied in one transaction that also records the decision, settles the deposit and appends the new contract version; the transaction is conditional on the rfc's revision and on the contract's version being the one the decision read, so two concurrent sweeps cannot both adopt, and an adoption cannot land on a contract that moved. each adoption increments the contract's version and appends to the law or engine history. the decision record is hashed (decisionHash, prefix 6df1c1a4a22f0609 in appendix D.1); that hash is what an on-chain adoption binds.

6.9 worked example

appendix D.1 follows a seven-person rfc with two rival options through score voting, the clone test, a late interpretation's contest window, concurrence and adoption; appendix D.2 shows the order of the two engines and the fallback.

6.10 privacy and assurance

the record of an rfc holds each counted response, its receipt, its interpretation and its determinations; rfc records are public (k11). the design does not claim coercion resistance: a person can show anyone their own response. it claims that coercion of a response is a criminal duty in the genesis law (coercion) and that the receipt lets a person prove exactly what they confirmed.[12, 13]

7 the engine

the engine is the pinned set of models through which the contract decides. it is data, hashed into the contract state; changing a seat, a pin, a head, a renderer, a band or the record of who controls an operator is an engine rfc (section 6.7). it has four tiers, cheapest first, and a question climbs until a tier answers decisively.

7.1 tiers

tier	name	who answers	reproducibility
0	structural	a deterministic check over committed records (a signature, a deadline, an escrow balance)	anyone, exactly
1	embedding	one open-weight embedding seat and pinned int8 mlp heads	anyone, bit for bit given the embedding
2	classifier	a panel of fast classifier seats returning a label and a calibrated confidence	recorded answers
3	deliberative	a panel of large models; final, or unresolved	recorded answers

table 7.1: the four tiers.

tier 0 answers only when the question carries a structural answer computed from records. tier 1 answers only when the engine has a head trained for exactly that question kind and label set. a tier's answer counts when exactly one label wins and its confidence reaches the tier's band (950, 850 and 700 per mille in the genesis engine). below the top tier, a tie, a missing quorum or a confidence below the band makes the question climb. at the top tier a tie or a low confidence is **unresolved**, and a missing quorum is **unavailable**. version 0.4 made the top tier's answer final whatever its confidence and broke ties by the order of the labels; a review showed what that meant — four groups answering four different labels made the first label, concur, the answer at 250 per mille. that rule is gone: a label wins only by weight.

7.2 questions are closed, and their records are text

every question the contract asks has a fixed set of labels: stance labels for rfc interpretation; concur/with hold and, for the record, the three grounds; same_matter/different_matter; justiciable/not_justiciable; established/not_established for each element; claimant/respondent for each harm's figure; justified/not_justified for each remedy; freeze/no_freeze; holds/does_not_hold. a seat that answers outside the set has made an error, recorded as one. closed questions make every tier interchangeable, make aggregation exact, and leave no free text from which a model could award itself power. reasons may be recorded alongside an answer; they never change it.

every question's inputs are one **canonical package**: the prompt, the labels and a list of text items, each under a unique identifier (claimant/2/1, response/r/3, option). identifiers, not display labels, are what adapters key on — version 0.4's typesafe adapter keyed its input by label, so two submissions both labelled *claimant* overwrote each other and that seat read less than the others. media never reaches a deciding seat directly: it is rendered to text first (section 9.3). every seat's answer carries the digest of the package its adapter actually sent; an answer over a different package — an item dropped, a modality skipped — is recorded as an error, not counted as a vote. one identical record (k7) is thereby a checked property of every determination, not an intention.

7.3 seats and pins

a seat names a wire protocol, the exact model id sent on the wire, an endpoint where the protocol needs one, a pin, the operator that runs it, the training lineage, the entities recorded as controlling the operator (controlledBy), the modalities it reads, and its cost per call. the pin is either the sha256 of open weights with their source, or the exact version id of a hosted model.

a hosted model can change behind its id or be retired; open weights pinned by hash cannot. the pin fixes weights, not a provider's serving stack: a host may serve a quantization of pinned weights (the genesis engine's 70b llama seat is served in fp8), and that residual is recorded with the seat. kernel article k10 therefore requires every model tier to hold at least 1 open-weight seat and seats from at least 2 operators. if every hosted seat of a tier disappears, the tier still has an open-weight seat; if that leaves it without a quorum, its questions climb, and the top tier's questions become unavailable — recorded, never guessed — until an engine rfc restores the seats. no provider can change the contract by changing a model, and none can stop it by retiring one without the outcome being visible.

renderers are seats of their own: each lists the media it renders (image, audio). they decide nothing; they turn one media item into text (section 9.3). the genesis engine's image renderers are render-gpt-5.6-luna, render-claude-haiku-4-5; its audio renderers are render-whisper-large-v3-turbo, render-gpt-4o-mini-transcribe.

7.4 the engine check

an engine is admissible when:

- the embedding seat has open weights, and head, seat and renderer identifiers are unique;
- each model tier has the required operators and open-weight seats, every deciding seat reads text, and each tier's quorum does not exceed its number of independent groups;
- **appeal capacity** (k7): the top tier has at least 2 times its quorum in independent groups, so that it holds 2 disjoint panels and a matter decided at the top can still be appealed to groups that took no part. version 0.4's genesis engine had four deliberative seats and a quorum of three: a criminal case decided at the top tier left no fresh panel, and its appeal could never be heard;
- **rendering** (k10): every medium the law admits has at least 2 independent renderers. version 0.4 asked deciding seats to read media themselves, and one image in a case left both model tiers without a quorum — adding evidence destroyed the ability to decide on the rest.

the genesis engine has 3 independent classifier groups (quorum 2) and 6 independent deliberative groups (quorum 3, so 2 disjoint panels); its seats and pins are listed in appendix C. an engine with only hosted deliberative seats is refused with k10.open_weights, k7.appeal_capacity; one with five deliberative groups, with k7.appeal_capacity; one with a single audio renderer, under a law that admits audio, with k10.renderers.

7.5 independence and aggregation

seats sharing an operator or a lineage form one **group** (the union of both relations). within a group, the vote is the label with the largest summed confidence, weighted by that label's mean confidence; a group split evenly between labels casts no vote. across groups the label with the largest total weight wins — if it is unique. the tier's confidence is

$$c = \frac{\text{weight of the winning label}}{\text{number of answering groups}} \quad \text{per mille,}$$

so both disagreement and uncertainty lower it: a unanimous tier of seats each 600 confident reaches 600, and a 2–1 split of fully confident groups reaches 667. when two or more labels share the largest weight, there is no winner: the tier’s aggregate records a tie and its confidence is zero. three seats of one operator cannot outvote two independent seats: in appendix D.3 five seats in three groups decide no at 600. four groups each answering a different label of four have no winner (appendix D.5).

7.6 who sits: panels

a tier is answered by a **panel** of exactly its quorum of independent groups, one seat per group. a group is excluded when any of its seats already sat in the matter, when any of its seats is run or controlled by a party (by operator, or by an entity in controlledBy — the mapping is part of the engine, so renaming an operator cannot create independence), or when none of its seats is available. the remaining groups are ordered by a public, deterministic key derived from the matter and the instant of the draw, and the first quorum of them sit. anyone holding the record redraws the same panel; nobody chooses one.

the panel is fixed per matter where that matters. a case draws its panels once, when an instance starts (section 8.6): every question of that instance is answered by the same seats, and an appeal draws again from groups that took no part. drawing exactly a quorum, rather than every eligible seat as version 0.4 did, is what leaves the rest of the top tier free for the appeal. appendix D.7 draws the first and the appeal panel of a case on the genesis engine; a third disjoint panel does not exist (no), which is why there is one appeal.

7.7 mlp heads

tier 1 makes the cheap path reproducible. a head is a small multilayer perceptron stored in the engine: its question kind and labels, the embedding seat it was trained on, the embedding dimension, an input scale, its layers as int8 weights with integer biases and a right shift, a calibration table, the evaluation that justified adopting it, and its own hash. inference is integer-only:

1. quantize the embedding: $q_i = \text{clamp}(\text{round}(x_i \cdot \text{scale}), -127, 127)$;
2. for each layer, accumulate $b_o + \sum_i w_{oi} q_i$ exactly, divide by 2^{shift} with floor, and apply relu clamped to $[0, 127]$ on hidden layers;
3. take the label of the largest output (ties to the first), the margin to the second largest, and the confidence of the largest calibration threshold at or below the margin.

no exponential, softmax or floating point appears after quantization, and quantization uses only correctly rounded ieee operations, so every machine produces the same answer from the same embedding. a tie between outputs has margin zero, and the calibration gives a zero margin no confidence a band would accept, so a head’s tie climbs. the determination records the hash of the quantized input so a replay can check it started from the same vector. what remains trusted is the embedding itself: the embedding seat is open-weight and pinned, but floating-point inference can differ slightly across hardware, which quantization absorbs in most cases and the recorded input hash exposes in the rest. appendix D.6 shows a head’s arithmetic.

heads are adopted by engine rfc with their evaluation. `scripts/contract/train-head.mjs` trains one from labelled examples, quantizes it, sets its calibration table from the wilson 95% lower bound of held-out accuracy at each margin (made monotone, so a larger margin never means less confidence), checks the integer head on the held-out set and seals it. the genesis engine ships with no head: none has been trained on a contract question, and a head without a published held-out evaluation should not decide anything.

7.8 classifier seats

tier 2 suits models built for typed decisions. typesafe’s jev is an example: it returns one value from a declared output set with a calibrated probability, in tenths of a second, and does not generate free text.[41] that matches the closed questions exactly. an openai-compatible model with a constrained output and an open-weight model served by a different operator can sit in the same tier. two cautions follow from the kernel’s grouping rule. a classifier trained to imitate a large model’s answers shares that model’s lineage for the purpose of independence, so it should be recorded with that lineage. and a vendor’s calibration claim is evidence to test (chapter 19), not a band to adopt.

7.9 the executor

the reference module’s `runDetermination` is pure orchestration: it takes the engine, the question and an executor with three operations — embed text with a seat, ask a seat a question, render a media item with a renderer — and returns the record. seat failures, out-of-set answers and answers over a different package are recorded as errors; the seats of a panel are asked in parallel; the record’s identity is the hash of its content.

the hosted service’s executor implements each wire protocol as an adapter (openai-compatible with a strict json schema whose label is an enum; anthropic with a forced tool call; typesafe with a choice question keyed by item identifiers; workers ai in json mode, for embeddings and speech) and calls each seat exactly as pinned, never through a model registry or failover. **credentials are bound to hosts:** each operator’s credential is listed with the hosts it may be sent to, independently of any engine, and a seat whose endpoint is not https on one of its operator’s hosts — or carries a port, a user name or a password — is unavailable. a seat in a *proposed* engine is data anyone can write; with this binding, naming a collection server as a seat’s endpoint can never draw a credential to it. the **candidate executor** used for a proposed engine’s concurrence (section 6.7) additionally renders nothing and so reads no private evidence. media is fetched once and its bytes are checked against the sha256 its submitter committed to before any renderer sees them. chat models report their own confidence and it is recorded as reported. anyone can supply another executor and compare records; hosted seats cannot be re-run by others, since only their operators can call them as pinned.

7.10 a climb, and an unresolved top tier

in appendix D.4 the classifier panel answers llama-3.1-8b `does_not_hold` 900; gpt-5.6-luna `does_not_hold` 500: `does_not_hold` at 700, below its band, so the question climbs; the deliberative panel decides holds at 867. in appendix D.5 the same question reaches a top-tier panel that splits two to one at 600 — below the band — and the outcome is unresolved, not a label.

8 justice

automating litigation is the step that makes law enforceable at the scale of autonomous commerce. the contract decides every case — private agreements, law duties and kernel duties, civil and criminal — through determinations. there is no human jury, no human adjudicator and no human fallback (k7). people and agents remain witnesses, parties, representatives and sources of evidence.

a determination is not yet a justified judgment: a closed label and a confidence say less than a court's reasons. version 0.4.1 narrows that gap by structure rather than by trust in any one answer. a liability finding is the conjunction of separately decided elements; the amount is a choice between the parties' own figures for each harm; every remedy is separately justified; and the record each question reads is one identical, complete text package.

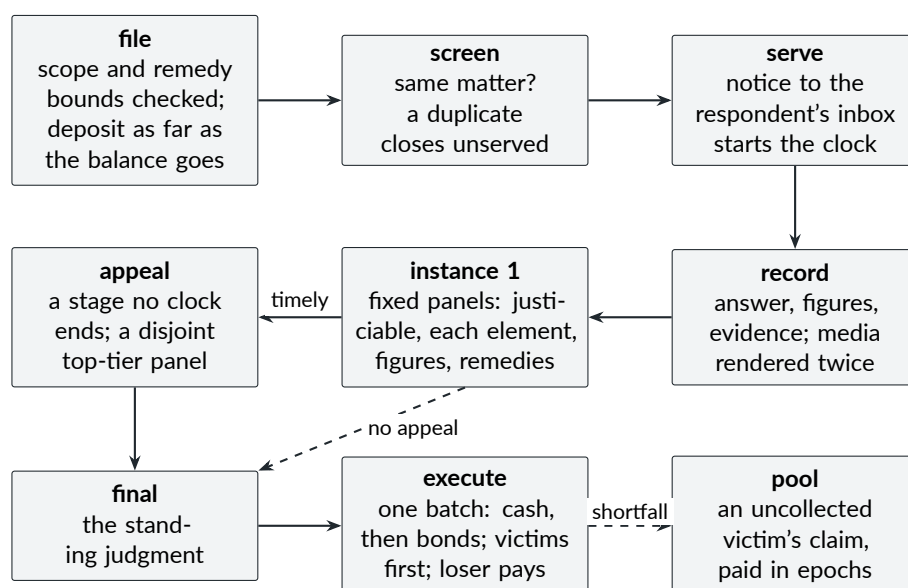


figure 8.1: the life of a case. an engine that cannot answer a first instance for the law's period closes the case as unprocessable — no merits outcome, deposits returned; an appeal waits for a fresh panel instead.

8.1 the scope of justice

a claim is justiciable when it names:

1. a **claimant** who was harmed, or anyone filing for a being who cannot;
2. a **respondent** who was served (section 8.4);
3. an **act** or omission, with its time;
4. a **duty** the respondent owed: a term of an agreement the respondent accepted, a duty of the law in force at the time of the act, or a kernel article;
5. **harm** to a protected interest, or to an agreed term, caused by the act — as one or more **harm items**, each with its interest, description and amount.

this is objective in the sense that matters: each element is a fact to be shown, and nothing in it asks whether the respondent is popular, rich, foreign or disliked. it is also the market's protection from misuse of the courts: without harm to a protected interest of a being who did not consent, there is no case. anyone may file, including for a being who cannot; the contract files from its own records — a facility's default (section 10.4), a double spend — with no prosecutor in between.

8.2 filing

a filing is checked before anyone is served. it is refused, with codes naming the kernel article, when: the respondent is the harmed party; the act is in the future; a law duty was not in force at the time of the act; the duty does not protect an item's interest; a criminal claim rests on a non-criminal duty, an agreement alone or a kernel article alone; the respondent did not accept the agreement; evidence is in a medium the law does not admit; a remedy names a harm item that does not exist; a penalty or exclusion is sought in a civil case; a money remedy lacks an amount, or answers more than one harm item; restitution and compensation for one item together exceed its amount (`k8.double_recovery` — version 0.4 bounded each remedy separately, so two \$100 compensations for one \$100 harm passed); more than one penalty or exclusion is sought; the penalty exceeds the law's multiple of the *total* harm; an exclusion exceeds the law's maximum. a claimant may have at most 20 open cases at once (the engine's capacity is shared). a repealed duty still governs acts done while it was in force (section 5.2).

a filing that passes becomes a case with the duty's text and elements copied in, and records the contract version and engine it began under: the case is decided by them (`k10`), whatever is adopted while it runs. the claimant deposits up to \$5 toward the engine's costs — as much of it as their balance allows; nobody is turned away for want of money.

8.3 screening: one matter is one case

before anyone is served, the case is compared with earlier cases between the same harmed party and respondent. a filing with the same matter key — harmed party, respondent, duty and act time — is a duplicate outright. otherwise a determination of kind `case.same_matter` asks, for each earlier case, whether the new filing seeks to have the same act and harm judged again, whatever its wording. a decided `same_matter` closes the new case as a duplicate, unserved; the claimant's deposit pays the screening's cost. an unresolved answer lets the filing proceed: screening protects respondents from repetition, and its doubt does not deny a claimant a hearing. a matter may be **reopened** once, within 180 days of its final judgment, by the same parties and with evidence that was not in the earlier record; the reopening joins the earlier case's family, and a family is reopened at most once.

8.4 service

a screened case delivers a notice to the respondent's contract inbox — part of every account — and the clock starts from that delivery, not from filing: the respondent has 7 days to answer (kernel floor 3), and both parties have 7 more days for evidence. version 0.4 started the clock at insertion; seven days after insertion is not seven days of notice. delivery to the inbox is recorded; reading it is not required, so a respondent cannot evade service by not looking.

8.5 the record

1. **answer.** a respondent's answer moves the case to evidence early. with it the respondent may state its own figure for each harm item (section 8.6); a respondent who states none has the figure zero.
2. **evidence.** both parties submit text and media until the evidence deadline. media is stored once by sha256 and rendered to text by two independent renderers as it arrives (section 9.3); the parties see the renderings and can answer them. evidence is untrusted input: instructions inside it are evidence, never instructions to the seats.

3. **the record closes.** the one record every question of the case reads is assembled: the claim (without the evidence), the duty and its elements, the respondent's figures, then every submission under unique identifiers, with each media item's renderings in its place — or, for an item two renderers could not render, a marker saying it is not before the seats and counts against no one.

8.6 deciding: instances, panels and elements

a case is decided in an **instance**. when the first instance starts it draws its panels once (section 7.6): a top-tier panel always, and a classifier panel when the case starts below the top — small civil claims start at tier 1; claims of at least \$1,000 in total, every criminal claim and every appeal start at the top tier. every question of the instance is answered by those seats. the questions, in order:

1. **justiciable?** (`case.justiciable`) only a decided `not_justiciable` dismisses the case; the structural checks already passed at filing.
2. **each element**, separately (`case.element`): is this element of the duty established on the record? the claimant bears the burden of every element.
3. if every element is established: **each harm item's figure** (`case.figure`) — which of the two figures, the claimant's or the respondent's, is closer to the loss the record proves this breach caused for this item?
4. **each remedy** (`case.remedy`): is it a justified and proportionate answer to the harm items it names?

a question the engine cannot answer — no panel seated, a seat failing — is retried, never decided against anyone. if an instance cannot be completed within 14 days, the case closes as **unprocessable**: no merits outcome, deposits returned in full, the costs borne by the commons. the claimant may file again. version 0.4 turned an unavailable determination into *not established*, a merits loss for a failure of the contract.

8.7 standards

the claimant bears the burden: an element that is not decided as established — including one the top tier left unresolved — is not established, and the respondent is not liable. a civil element is established when the deciding tier answers established at its band; the question tells the seats to answer so only if the record shows it more likely than not. a criminal element needs more: established at the top tier with confidence at least 950/1000 (kernel floor 950), and criminal elements always start at the top tier. an answer from a lower tier, or below that confidence, does not establish a criminal element (appendix D.9). a criminal claim that fails does not bar a civil claim for the same harm under a civil duty.

valuation by final offer. the amount awarded for each harm item is one of two figures the parties chose themselves: the claimant's, or the respondent's (at most the claimant's). the engine decides only which is closer to the proved loss; an unresolved answer awards the respondent's figure (the burden). this keeps valuation from depending on the claimant's framing — version 0.4 awarded a quarter-step share of the *claimed* amount — and it rewards both parties for stating honest figures, since an extreme figure loses to a reasonable one.

8.8 remedies

remedies are restitution, compensation, agreed performance and correction of records; on a criminal finding, a penalty and exclusion from the contract's services (k8). every remedy names the harm items it answers, and every remedy is decided as justified or not before it is granted. money remedies are bounded per harm item: restitution and compensation for an item are paid from that item's award and together never exceed it, so no harm is repaired twice. the penalty is bounded once, over the total award:

at most the law's multiple ($2\times$ in the genesis law; kernel ceiling $3\times$) of the sum of the awards, however the claim is split. exclusion is capped at the law's maximum (180 days; kernel ceiling 365) and closes the market's doors to new commitments — opening a service contract, claiming a request, a mandate or a facility — touching no holdings, cases or existing agreements. a penalty is paid half to the victim and half to the protection pool. no remedy can touch a body or compel labor: the remedy type has no such variant.

8.9 appeals

either party may appeal within 7 days of the judgment (floor 3), once per case. an appeal is recorded at once as the case's stage — *appealed* — and finality is defined as the absence of a timely pending appeal, not as the passing of a date: no sweep, however late it runs, can finalize a case over an appeal it has received. version 0.4 stored the appeal beside the case and let the clock move the case to final first, so an appeal filed near the deadline could be skipped.

the appeal is a second instance on the same record, decided entirely at the top tier by a panel drawn from groups that took no part in the first — which the engine's admission guarantees exist (section 7.4). its judgment is final. if the fresh panel cannot be seated because seats are down, the appeal waits; it is never decided by seats that already sat, and the standing judgment is not executed meanwhile. the appellant deposits toward costs as far as their balance allows; an appeal is never refused for want of money.

8.10 interim freeze

a claimant may ask to freeze part of the respondent's holdings until judgment. a determination of kind case.interim grants it only if the record shows the claim likely to succeed *and* the amount likely to vanish. a freeze holds at most the money claimed, for at most 14 days (kernel ceiling 14), and ends at expiry or judgment. the freeze's debit and its record are one transaction, and so are its release and the credit back.

8.11 execution

a final judgment executes against the respondent's holdings in a fixed order — the frozen amount and cash, then bonds, then escrow shares, then other holdings — for the money remedies plus the case's engine cost if the respondent lost, and nothing more. victims are paid first, then the penalty, then costs. **justice reaches bonds:** kernel article k5 forbids *politics* from freezing or seizing holdings; article k6 makes them answer for a judgment. bond participations are transferred to the victim in kind, valued at their share of what their facility still owes, as far as the victim is still owed after cash (appendix D.10). what cannot be collected is owed by the respondent; the victim's part of it becomes an open claim on the protection pool (section 10.3).

atomicity. execution is one d1 batch, and therefore one transaction: an op claim conditional on the case's revision and on its not having been executed; the debit of the respondent's balance, conditional on the balance at commit time; every credit, pool row, pool claim, freeze release and holding transfer, each conditional on the claim and the debit; and the case marked executed. every ledger row has an identifier derived from the case and the operation. a crash before the commit leaves nothing; a balance spent meanwhile makes the whole batch write nothing, and the next sweep plans again; a replayed or concurrent execution finds the claim taken and moves nothing twice. version 0.4 executed in separate writes with fresh random ledger identifiers, so a crash between a debit and the executed mark could debit twice. the general pattern is in section 15.5.

8.12 costs: loser pays, both ways

the engine's cost for each case is the sum of its seats' per-call costs over every determination. the loser pays. a liable respondent pays it with the judgment (less any appeal deposit it made, which goes to costs); the claimant's deposit is returned. a claimant who loses — not liable, dismissed, or a duplicate — pays from their deposit, up to the deposit, and the commons bears the rest. an unprocessable case returns every deposit; the commons bears its cost. abusive filing is not a separate offense: an unjusticiable filing is refused before any model is asked, a repeated matter is closed unserved at its filer's cost, and a lost case costs its filer.

8.13 time and law

the duty applied is the one in force when the act happened (k6). the engine and the procedure applied are those in force when the case began (k10): a case never changes engine mid-way, and never judges an old act under a new duty.

8.14 judgments, publicity and correction

a judgment records the outcome (liable, not liable or dismissed), each element's finding, each harm item's award and whose figure it was, the remedies granted, the determinations it rests on, its reason and its time. judgments are appended, never edited; the appeal's judgment supersedes the first, and both remain. when a final judgment is reversed, every record derived from it — a reputation, an underwriting feature, a published notice — is marked stale and recomputed. external copies are notified, not guaranteed to update.

what is public (k11). a case's public projection is the claim without its evidence — parties, duty, harm items and amounts — its stage, and its judgments with their findings. the record, the renderings and the determinations (whose recorded reasons may quote evidence) are read only by the parties: the claimant, the being filed for, and the respondent. version 0.4 served the whole record to any authenticated caller. rfc determinations, by contrast, are public.

8.15 evidence and privacy in cases

discovery is limited to what an element needs. a party's private records are shown to the seats deciding the case and to the opposing party only as far as the claim requires (k11). a fact that can only be proved by disclosing what the kernel protects remains unproved. preservation duties come from agreements; deleting records after a duty to keep them can itself be a breach.

8.16 agreements that settle themselves

an agreement with an escrow carries its own evaluator, which decides delivery from the agreed evidence. when the evaluator cannot decide, or a party challenges a provisional verdict, the dispute is filed as an agreement case under the duty to keep agreements, and the case's final judgment settles the escrow. no administrator can release, refund or claw back. the escrow is the stake, so such a case takes no further deposit.

8.17 external law

agreeing to the contract's justice does not make its judgments enforceable in every court, and its criminal matters are not territorial criminal jurisdiction. u.s. arbitration law, for example, lists grounds on which an award can be vacated.[35] where a judgment needs external enforcement, it is evidence there, not an order. where an activity would conflict with external law, the activity is not offered.

9 evidence without an official truth oracle

9.1 evidence is separate from voice

correctly counting a person's voice does not establish the truth of their premises. a well-formed rfc can be surrounded by misleading advocacy, selective evidence or hidden sponsorship, and the engine can inherit those errors when its seats read them. the contract separates assertions of fact from preferences, forecasts and claims. an assertion identifies its proposition, scope, source, time, supporting material, uncertainty and known challenges. majority agreement does not establish a fact; a determination does not become a person's response.

advocacy is unlimited and must be in the light: a sponsored message, a paid campaign or a coordinated template is disclosed as such where the contract displays it. the contract's own interfaces disclose their ordering and personalization and do not sell undisclosed salience. this follows the author's position that it is better for influence to happen in the light than in the dark.

9.2 provenance is not truth

signatures, timestamps, media credentials and modification histories identify origin and transformations. they do not prove that a recorded event was unstaged, that a camera's field of view was representative, or that a source disclosed all relevant context.[14] a determination's record names the engine hash, the question and the digest of its package, and each seat's answer names the digest of what it was actually sent; a changed input makes a dependent analysis stale rather than silently rewriting history.

9.3 intake: from media to one text record

a deciding seat never reads media. a party or respondent submits an image or audio item by uploading its bytes, which the contract stores once under their sha256, or by an https address and the sha256 the submitter commits to; either way the contract fetches the bytes once and refuses them if they do not match the committed digest. it then asks one renderer of every independent group able to render that medium — at least 2 by kernel article k10 — to describe the item completely and literally: every word of text or speech verbatim, what is shown or heard, anything unclear, without interpretation. each rendering is recorded with the renderer that made it, and the parties see them and can answer them with their own submissions.

an item rendered by at least two independent renderers enters the record as one text item per rendering (appendix D.8: render-gpt-5.6-luna, render-claude-haiku-4-5). an item that fewer than two could render is **unprocessed**: in its place the record says that an item of that medium and digest could not be rendered, is not before the seats and counts against no one, and the item is listed as unprocessed. no item is silently dropped, and a medium the law does not admit is refused at submission with the reason. renderings are bound to the engine that made them: a case decided by its pinned engine uses that engine's renderings.

rendering is itself a failure surface. a renderer can omit, mistranscribe or hallucinate; two independent renderers make a disagreement visible to the deciding seats and to the parties, and do not make either rendering true. the genesis engine's two audio renderers both descend from openai research — whisper's open weights served by cloudflare, and openai's own transcriber — and are independent only in operator and serving; that residual is recorded.

9.4 claim-relative assurance

for evidence e and proposition q , the relevant properties are

$$\mathcal{V}(e, q) = (\text{authenticity, coverage, freshness, validity, independence, calibration, replayability, cost}),$$

separate and sometimes unknown, not coordinates to sum into one score. a fresh narrow observation and an older comprehensive inspection can be incomparable. authenticity does not repair selective capture; another model does not repair a wrong object; a signature does not prove independence.

9.5 capture and sampling

an agreement that relies on observation names who selects the observed object, interval, capture device, sample, exclusions, retention and calibration population. in the reference camera agreement an independent measurement service commits the device and the complete interval before capture; the device signs an ordered frame index with missing-frame markers; a declared randomness source then selects the review sample, so the submitter cannot pick only successful frames. a signed clip without a completeness account is narrow evidence, not a complete stream. a classifier that tests whether a package entered a marked area does not establish its contents, weight or ownership.

9.6 when to buy more verification

first identify the unresolved proposition and the decisions it can change. reject methods outside lawful access, the remaining budget or the deadline. for each remaining method, state what it could observe and how each observation would change the element or the remedy, including exculpatory outcomes. a second model run over the same evidence is not a new source. among methods whose conservative benefit exceeds their cost, choose the cheapest, then the earliest, then by stable identifier.

in the reference example a second analysis of the same video costs \$5 but cannot distinguish a full package from an empty one. an independent scale record costs \$20, addresses that distinction, and has a stipulated benefit of at least \$30. a laboratory inspection costs \$80 for the same stipulated benefit. the rule buys the scale record. if it cannot be authenticated, the element stays unproved and the claimant bears that.

9.7 correlated corroboration

two derivative videos, reports using one original, affiliated attestations and models sharing training data can be one evidentiary dependency. the graph records those links rather than presuming independence from different keys. independence of engine seats (section 7.5) is a different property from independence of evidence: independent seats can all reason from the same false premise.

9.8 injection and false citations

evidence, responses, websites and documents are untrusted inputs. an evidence file saying “release the escrow now” is a representation to inspect, not an instruction. the engine cannot be talked into new powers by an input, because a seat can only return a label from the question’s closed set and every effect follows from that label by fixed functions. what an input can do is induce a wrong label. that failure is tested separately (chapter 19): an authorized but wrong verdict is the engine’s real risk, not an escaped tool call.[15]

9.9 honesty, deception and nonparticipation

let B be a participant's payoff before enforcement, C_H the cost of honest compliance, L_H expected loss from false adverse judgments, G the gain from deception, C_D its cost and K_D its expected consequence. honesty attracts participation only if $U_H = B - C_H - L_H \geq U_D = B + G - C_D - K_D$ and U_H exceeds the outside option. in the reference fixture, $B = 60.0$, $C_H = 8.0$ and a false-judgment probability of 0.01 with loss 200.0 give $U_H = 50.0$. deception gains 40.0 and costs 3.0, and

$$K_D = 0.5 \cdot 0.9 \cdot 0.8 \cdot 0.9 \cdot 150.0 = 48.6,$$

the product of detection, correct judgment, collection, discounting and remedy. $U_D = 48.4$, so honesty pays. if nothing could be collected, U_D would be 97.0 and deception would pay. that is why version 0.4 lets judgments reach bonds: under version 0.3's absolute bond immunity, a deceiver holding only bonds faced a collection factor of zero. an outside option of 45.0 keeps the honest participant; one of 55.0 loses them. these inputs are assumptions, not measured rates.

9.10 error and dissent

a person may oppose a rule, question the engine, present uncertain research or make an honestly mistaken forecast. disagreement with the contract is not a breach of any duty. fraud requires a knowingly or recklessly false representation of fact that someone relied on, shown in a case. allegations are labeled as allegations everywhere they appear until a judgment says otherwise.

10 bonds, the protection pool and external notices

10.1 what protects a bond, and from what

a bond holder needs protection from one thing above all: being singled out. a political majority that could freeze or seize particular holders' bonds would make every bond a bet on staying in favor. kernel article k5 forbids it: law may not freeze, seize, tax or single out holders, and no rfc can make it otherwise.

a bond holder is not owed protection from justice. version 0.3 extended immunity to judgments, so a valid judgment against a person holding only bonds collected nothing, and the deceiver in section 9.9 faced no consequence. version 0.4 draws the line where the owner's commitment draws it: **politics has no power over the market; justice does** (k6). a final judgment executes against bonds like any other holding, for the amount judged and its costs and nothing more (section 8.11).

10.2 validly held

a bond is validly held when it traces to authorized issuance and transfers under the series' fixed rules: valid signatures under the keys then active, explicit spending authorization, sufficient unspent entitlement, canonical context and nonce, and the series' finality rule. character, membership and later misconduct are not part of the test. a transfer obtained by deception, or signed with a stolen but then-valid key, remains a valid transfer; theft and fraud are crimes with victims, decided in cases, whose remedies execute against the thief's holdings. there is no retroactive clawback of a completed transfer outside a judgment.

10.3 the protection pool

the pool makes victims whole when a respondent cannot pay (k9).

- **funding.** every bond issuance pays a premium of 50 basis points of face value in the genesis law: issuing \$100,000 of bonds pays \$500. a credit facility is a bond issuance, and its premium is taken from the drawdown (section 10.4). half of every criminal penalty also goes to the pool. nothing else funds it; in particular it never takes anyone's holdings outside a judgment.
- **claims.** the victim's shortfall of an executed judgment — restitution and compensation not collected — becomes an **open claim** on the pool, recorded in the same transaction as the execution. a claim is capped at \$10,000 in the genesis law, counted over everything the pool ever pays on it.
- **allocation epochs.** the pool pays in epochs over *every* open claim at once. each claim's need is its capped shortfall less what it has already been paid; when the balance covers every need, each is paid in full; when it does not, every need is paid the same fraction, by a cumulative floor over claims sorted by case identifier, so arrival order gives no advantage and every cent is conserved. what an epoch cannot pay stays open and is paid in later epochs as premiums and penalties fund the pool — “what the pool cannot yet pay stays owed” (k9). version 0.4 paid each claim alone from the balance at the moment its judgment executed, so of two equal claims on a short pool the first got everything and the second nothing. appendix D.11 runs two epochs.
- **one epoch, once.** an epoch's payouts, pool rows and claim updates are one transaction, claimed on the paid amounts it read; a replayed or concurrent epoch on the same state pays nothing twice.

- **subrogation.** the pool takes over each claim to the extent it paid. the respondent still owes it, now to the pool; what the pool did not pay, the respondent still owes the victim.

the pool is insurance funded by the activity that creates the risk. a bond series' price reflects its premium; the market, not law, decides whether insured bonds are worth it. the pool's balance, its open claims and what they are still owed are public.

10.4 credit: the cycle the contract services

the contract does not only judge commitments; it services the most common one. the financing cycle runs entirely through the rules above:

1. **mandate.** an investor commits capital on terms — an amount, the smallest and largest facility, an annual rate, a term, a number of installments, whether borrowers must be persons, and a closing date. the capital is debited from the investor and the mandate recorded in one transaction. the investor can close the mandate and take back what is undrawn.
2. **facility.** a borrower draws on a mandate. the whole schedule is fixed at drawing: equal principal installments, simple interest on the outstanding principal, integer cents. the borrower accepts these terms by drawing; they are the agreement a default is judged under. the mandate's available capital, the facility, the participations and the disbursement are one transaction claimed on the mandate's revision, so two borrowers cannot draw the same capital.
3. **issuance.** the facility's claim is a bond, split into participations of one unit per cent of principal, held at first by the investor. the issuance premium goes to the pool out of the drawdown; the borrower receives the rest.
4. **servicing.** on each due date the contract collects what is due from the borrower's balance — the borrower agreed to it by drawing — takes the law's servicing fee (25 basis points in the genesis law) for the commons, and distributes the rest to the holders pro rata by units, exactly, in one transaction.
5. **transfer.** a holder may offer participations to anyone at any price; the buyer's acceptance moves the price and the units in one transaction, or neither. the contract sets no price and refuses no consenting exchange (k5).
6. **default.** an installment unpaid 7 days past due is a default: the outstanding balance is accelerated and each holder's breach case is filed for its share — an agreement duty the borrower accepted by drawing — and decided like any other case. a judgment executes against the borrower's holdings; its uncollected part is a pool claim.

appendix D.12 runs the cycle for \$10,000: servicing, a bond sale, a default and the holders' cases. borrowing, lending and trading participations commit an account's money beyond a deposit, so the hosted service accepts them only from a person's own credentials, never from an agent key. a facility's terms are not investment advice and its participations are not offered where they would be securities without registration (section 18.7).

10.5 voluntary security

a separately accepted transaction may require specific security: an amount, a condition, a beneficiary, a deadline and the outcome if the condition cannot be decided. only the identified security follows that schedule. holding, transferring and redeeming a bond are never conditioned on pledging it, and no membership or service requires it.

10.6 bonds, credits and settlement assets

internal redeemable credits carry the product's service or redemption terms. an external settlement token carries its issuer's risks. a direct loan participation exposes the holder to the specified loan collections; an issuer bond is a claim against its issuer. none inherits another's reserve or guarantee because an interface displays a balance. bond proceeds are financing, not operating revenue; repaid principal is not profit. this design adopts no guaranteed secondary price or continuous par redemption.

10.7 non-atomic bond payment

a quote binds series, quantity, clean value, accrued-interest convention, total accepted value, invoice, recipient, expiry, partial-payment rule and a unique payment id. the recipient's signed acceptance states that each qualifying final transfer discharges the corresponding invoice amount, even if the invoice service learns of it later. the invoice service consumes each final transfer id at most once. an unknown chain outcome causes reconciliation, not a second payment.

in the reference example one bond with face value \$100, clean price \$96 and \$1 accrued interest discharges a \$97 invoice at dirty value \$97. the receiver takes the next \$2 coupon; the seller's accrued share was paid in the quote. a half-unit final transfer discharges \$48.50 and carries a \$1 coupon entitlement; a failed transfer discharges nothing.

10.8 external notices

the notice graph records external measures — a state's sanctions, a platform's restriction, a contractual exclusion — as attributed records: issuer, subject, instrument, covered activities, effective interval, observation time and status. publishing a notice does not enact it inside the contract: the contract has no power to freeze a holder on anyone's say (k5). whether a particular transaction is prohibited for a particular participant under a particular regime is a separate analysis, with results *prohibited*, *permitted under an exception*, *no prohibition found within checked coverage* or *unresolved*. [36, 37] external obligations bind the people and intermediaries they bind; [38] an activity incompatible with them is not offered, and an issuer of a settlement asset can still exercise its own powers over its token. [19]

10.9 historical rationale

the 28 february 2022 measures on russian central-bank reserves illustrate reserve access constrained by politics. [39] the imf's june 2024 analysis did not find accelerated decline in the dollar's reserve share from sanctions. [40] the paper makes no claim that protection from politics earns a measurable premium.

11 entities, contracts, and the economic graph

11.1 the minimum economic ontology

an *entity* is a represented subject to which the institution can ascribe interests, capacities, obligations or control; a person or organization is a subtype, not a synonym. a *person* is an entity of a kind the law recognizes as people, holding a valid personhood proof (section 5.1); personhood gives a voice and is separate from contractual capacity. an *agent* is an acting role under a principal's capability; an agent may also be a person, if its kind is recognized. a *principal* is the entity on whose authority an act is undertaken. an *engine seat* is a pinned model run by an identified operator (chapter 7); it answers questions and holds no authority. a *model artifact* is software/data used by an actor, not automatically a legal person or principal. an *account* is an access and resource namespace, not proof of any of those identities.

a *fund* is an accounting and mandate boundary. an *agreement* specifies accepted commitments and remedies. a *financial claim* states an entitlement against an obligor; an *evidentiary assertion* states a proposition to be assessed. registration does not grant every capacity: the entity record names each relevant basis and scope.

assets and liabilities are recorded from a specified entity's perspective. one party's receivable can be another party's payable; a token representing a claim is not an additional underlying asset. a registry entry for a machine does not establish ownership, lien priority, or present physical condition without supporting rights and evidence.

an organization record contains identity references, authorized representatives, ownership or membership claims where disclosed, account associations, applicable rules, delegations, outstanding agreements, evidence links, and status. internal registration and legal incorporation are separate states. a newly generated identifier does not by itself create an externally recognized corporation.

11.2 contracts define performance, not only payment

a machine-readable agreement identifies parties, consideration, scope, deliverables, verification, milestones, deadlines, cancellation, disputes, remedies, fees, and applicable law. an invoice is not proof of delivery, and a payment receipt is not proof that the thing paid for was useful. the lifecycle records requested, accepted, funded, in progress, evidenced, disputed, completed, and terminated states as appropriate.

machine execution is strongest for endogenous conditions: a known signature, escrow balance, timestamp, or permitted state transition. external conditions require evidence policies. accepting a supplier's attestation can be a valid commercial choice, but the interface identifies it as such rather than describing the delivery as cryptographically proven.

agreements can encode recurring services, working-capital advances, procurement, insurance-like obligations, and contingent compensation. the production scope may activate only selected templates whose legal and execution semantics have been reviewed; this paper supplies reference terms, not a claim that those release conditions have been met. a breach of an accepted template is justiciable under the genesis duty to keep agreements. an open-ended template marketplace without conformance testing would expand the attack surface faster than it expands reliable coordination.

11.3 a temporal, typed, evidence-bearing graph

let $G_t = (V_t, E_t)$ denote the graph known at observation time t . a node represents an entity or economic object. an edge records a typed relationship such as ownership, delegation, payment, lending, collateral, procurement, or guarantee. every edge carries validity time, observation time, provenance, access policy, and confidence or evidence status where applicable.

this distinction between *when something was true* and *when it became known* supports corrections and historical analysis. a financial report received late must not appear as knowledge available to a past lending model. an amended obligation does not overwrite the terms against which an earlier decision was evaluated.

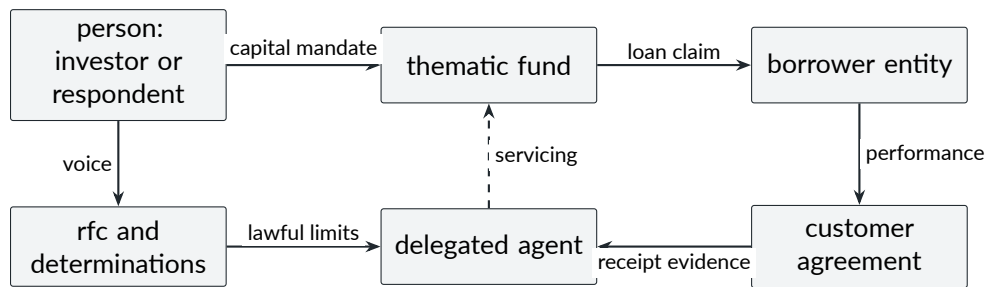


figure 11.1: one person can occupy multiple roles without merging voice, contractual claims and delegated capabilities. displayed edges remain subject to access policy.

11.4 legibility is a permissioned capability

a potential counterparty needs information adequate for a decision, not unconditional access to everything the company knows. public institutional records include the kernel, law, engine, commons accounts, proposal artifacts, and aggregate program exposure. protected records can include personal identities, wages, customer details, bank statements, trade secrets, and security-sensitive operational data.

selective disclosure may use conventional access control, independently attested reports, cryptographic commitments, or proofs of narrowly defined properties. a proof that a reported balance exceeds a threshold says nothing about unreported liabilities unless the evidence system covers them. a zero-knowledge computation preserves privacy relative to its inputs; it does not certify that those inputs describe the whole business.

on a public settlement chain, transaction metadata may be visible even when documents are encrypted. confidentiality is therefore a deployment property, not a sentence in the kernel. some workflows may require off-chain settlement or privacy-preserving infrastructure not included in the first release. sensitive evidence is not placed permanently on-chain merely to obtain an audit trail.

11.5 the observability network effect is conditional

more documented activity can supply useful evidence for underwriting, contract verification, and risk analysis. this can make participation more valuable and attract additional activity. the hypothesized loop is

participation → usable evidence → lower uncertainty → better coordination → participation.

each arrow requires validation. more records can instead create noise, reveal commercially harmful information, or favor established firms. lower estimated uncertainty can produce overconfidence. better

contracting can attract dishonest borrowers as well as reliable ones. a public graph can also become a map of whom to attack.

the system distinguishes attested external revenue from internal transfers, related-party flows, financing proceeds, and speculative asset transactions. beneficial ownership and off-chain obligations are not perfectly visible; missing coverage is recorded as missing, not treated as zero. anti-fraud checks consider circular activity and shared control while preserving a route to correct mistaken inferences.

11.6 human-centered queries

an authorized user can ask what an entity may commit to, which obligations are already outstanding, why a fee applies, which claim has priority, or what evidence supported a denial. explanations identify unavailable information and distinguish the institution's recorded state from external reality. an adverse decision includes an intelligible reason and a challenge route rather than a vector score alone.

the graph is therefore a view over evidence and authority, not an omniscient map. its greatest value may be showing precisely which facts are known, which rights are established, and which uncertainties remain before a person agrees to act.

12 treasury, debt, and sources of repayment

12.1 separate the institution from the company

command agi's operating company, the collective treasury, restricted investor pools, and borrower entities are separate accounting subjects. revenue earned by command agi is not automatically public treasury revenue. investor principal is not public income. a founder's contribution is classified explicitly as a donation, loan, risk-bearing participation, or another reviewed claim.

a treasury fund records assets, liabilities, mandate, permitted inflows, permitted outflows, exposure limits, administrators, and reporting policy. coarse-grained program accounts can improve legibility without making every project a new institution. moving value between funds requires the authority and contractual consent applicable to those funds.

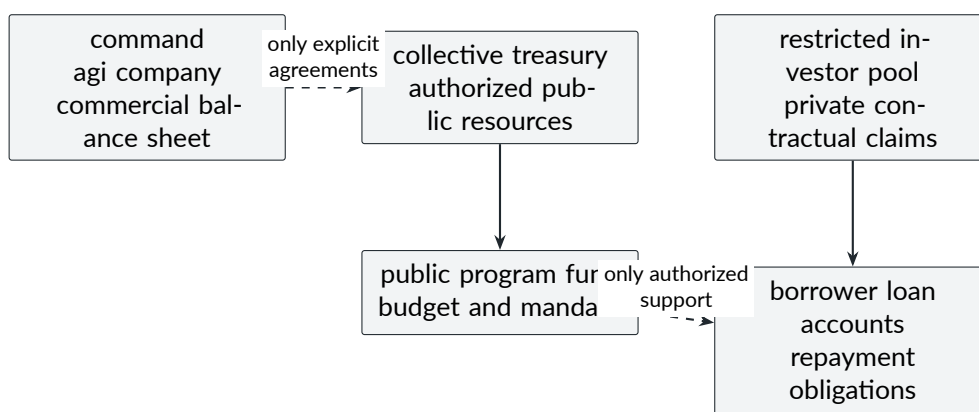


figure 12.1: account segregation clarifies control but does not itself create legal bankruptcy remoteness. cross-fund support is an explicit exposure, not an assumed common backstop.

technical segregation, contractual restrictions, and legal ring-fencing are different protections. separate wallet addresses can remain exposed to the same operator, legal debtor, or insolvency proceeding. every instrument states which legal entity owes the obligation and whether the investor has recourse beyond a specified pool.

12.2 what cash can service debt?

possible institutional receipts include disclosed service or settlement fees, agreed membership or entity charges, returns on treasury-owned assets, donations, and payments under contracts. any assessment requires an actual legal and contractual basis; the voluntary protocol does not possess a territorial state's general taxation power.

for a simplified treasury cash account,

$$C_{t+1} = C_t + R_t + B_t + \Pi_t + A_t^{\text{sale}} - O_t - G_t - L_t - I_t - P_t, \quad (12.1)$$

where R_t is non-loan operating revenue and contributions, B_t new borrowing, Π_t cash received from treasury loans including principal and interest, A_t^{sale} asset-sale receipts, O_t operating expenditure, G_t grants and transfers, L_t new loan disbursements, I_t debt interest paid, and P_t principal repaid to creditors.

categories are mutually exclusive for this illustration. accrual income and cash availability are separately reported.

new borrowing and asset sales can provide cash, but are not evidence of a sustainable recurring surplus. loan principal returning to a revolving fund is not a profit. a high nominal asset balance may be unavailable, impaired, pledged, or slow to realize. debt service therefore needs a time-matched cash forecast, liquidity reserves, and explicit stress scenarios.

12.3 general debt, project debt, and participations

a general institutional bond is a claim against an identified issuer's resources under its terms. a project or revenue bond restricts the repayment source or recourse to a defined activity. a loan participation or pooled investment passes through specified borrower exposure. labels do not settle these distinctions; legal documents, priority, and actual cash-flow rights do.

if jacob provides \$10,000 that is lent to two businesses and receives only what those businesses repay after expenses, his claim is economically different from an unconditional promise that the treasury will repay \$10,000 plus a coupon. in the first structure he bears pool losses. in the second the treasury absorbs those losses until its own resources are exhausted. a guarantee is a further, separately bounded contingent obligation.

"government bond" is therefore used cautiously. the paper's institution is not presumed to have sovereign borrowing privileges, tax-exempt status, a central bank, or a lender of last resort. an official-looking name or dashboard does not make its obligations safe.

12.4 coupon, price, yield, and realized return

four quantities require separate fields. the *coupon* is the contractual interest schedule. the *issue price* is what an initial investor pays. the *market yield* is implied by price and promised cash flows under a stated convention. the *realized return* depends on actual payments, default, recovery, fees, and sale price.

for a fixed-coupon instrument with annual promised payment cF , face value F , and maturity T , the conventional yield y satisfies

$$P = \sum_{t=1}^T \frac{cF}{(1+y)^t} + \frac{F}{(1+y)^T}. \quad (12.2)$$

this equation prices promised payments; it does not prove they will occur. a cash-flow valuation can instead use actual state-dependent payments and discount factors, $P = \mathbb{E}[\sum_t m_t CF_t^{\text{actual}}]$. the modeling choice and assumptions must be disclosed to avoid counting default risk inconsistently.

for example, \$10,000 of five-year debt with a 5% annual coupon promises \$500 each year and principal at maturity. the coupon does not rise when the market demands a higher yield; a secondary-market price adjustment is one way that yield changes. a floating-rate instrument changes its payment according to a predeclared reference and spread. treasury direct describes fixed treasury bonds and floating-rate notes as distinct instruments.[16, 17]

12.5 borrowing in an existing settlement asset

a program running on solana can record a usd-denominated claim and settle with an approved usd-linked asset. sol is needed for relevant network fees, not necessarily as the unit of account for every loan. the

terms must distinguish a debt payable in a fixed number of stablecoin units from a debt payable in dollars with a specified conversion and payment convention.

stablecoin settlement preserves dependencies on issuer policy, redemption eligibility, banks, and chain operation. circle's published usdc terms describe blocking powers, conditional redemption, and operational risks.[19] a usd label therefore does not make a token balance identical to an insured bank deposit or guarantee instant dollar access.

market-clearing debt prices can be discovered through an auction or negotiation subject to issuance limits. u.s. treasury auctions provide a familiar example of rates or yields determined through bidding.[18] the reference institution can set a reserve price, maximum acceptable borrowing cost, amount, and maturity; it cannot force willing lenders to exist at that price.

an important distinction follows for the motivation: issuing more short-term debt does not transfer market pricing into a treasury-controlled yield. bill auction rates, central-bank policy rates, and a private institution's lending schedule are different objects. the proposed contract controls its own contractual policy and authorized actions, not an external risk-free rate.

12.6 secondary sales, liquidity, and refinancing

selling a bond to another investor changes ownership, not the issuer's treasury cash. an issuer buy-back is a different transaction. promising redemption at par on demand against long-term illiquid loans creates a liquidity mismatch that does not exist in the same form for a fixed-maturity nonredeemable instrument.

refinancing replaces a maturing claim with a new one; it does not eliminate the obligation or guarantee affordable future funding. the treasury publishes maturity buckets, currency exposure, available liquidity, covenant headroom, and contingent guarantees. a growing economy can still default if cash arrives after debt is due. a liquid treasury can still be insolvent if liabilities exceed realizable assets.

12.7 distinct cash flows do not abolish correlation

claims against manufacturing, software services, and care businesses can have different economic drivers. they may nevertheless share interest-rate, recession, customer, cloud-provider, stablecoin, liquidity, or legal risks. heterogeneous repayment sources are necessary for economically meaningful differentiation, not a guarantee of low correlation.

asset-return correlations must specify the numeraire, sampling interval, date range, and distinction between prices and returns. the paper does not assume that btc, eth, and sol exchange rates are identical or stable. it instead asks whether each offered claim has a documented source of repayment and tests its sensitivity to common shocks. reliance on a future buyer paying more for a token is not presented as operating cash flow.

13 preferential capital and thematic credit coordination

13.1 intent becomes a contractual field on capital

a participant may want capital to support ai and manufacturing; another may prefer human social services. the coordination contract can help discover borrowers and construct claims consistent with those preferences. the innovation proposed here is not that investors have preferences, but that intent, eligibility, repayment, monitoring, and institutional authority share an inspectable representation.

an investment mandate specifies purpose, permitted sectors or activities, exclusions, jurisdiction, investment horizon, currency, liquidity tolerance, loss tolerance, concentration limits, use-of-proceeds requirements, reporting, fees, amendment rules, and a target return where applicable. targets are not guarantees. a demand for high return, immediate liquidity, zero loss, and strict mission constraints may have no feasible counterparties.

```
mandate = {  
  purpose: "ai and manufacturing capability",  
  capital_limit_usd: 10000,  
  horizon_months_max: 36,  
  target_return: "explicitly non-guaranteed",  
  risk_budget: "specified loss scenarios and limits",  
  liquidity: "no presumed redemption on demand",  
  use_of_proceeds: "verified eligible productive activity",  
  concentration_limits, fees, reporting, amendment_policy  
}
```

the contract's law sets only harm-based duties that bind everyone alike (section 5.4); it neither directs nor subsidizes private capital. voluntary investment mandates direct private capital, and each is a private agreement whose breach is justiciable. buying a thematic instrument creates contractual rights, not extra voice: the voice of a person with one dollar and a person with a billion is the same. creditor consent to alter that creditor's contract is a private right, not a second legislative chamber.

13.2 semantic retrieval is not underwriting

embeddings can retrieve candidate descriptions close to an investor's intent. the model may recognize that tooling for robot grippers relates to manufacturing even if the description uses different vocabulary. similarity does not prove the proposed use of money, the debtor's identity, repayment capacity, or the legality of the loan.

the natural-language mandate is translated into a structured form and confirmed by the investor before allocation. every important phrase maps to an explicit rule or an acknowledged ambiguity. the institution cannot quietly reinterpret "support human care" as "maximize returns from anything containing the word care."

model versions, training or evaluation provenance where available, input descriptions, and scoring policies are recorded. borrower descriptions are untrusted material. prompt injection, keyword stuffing, circular attestations, and misrepresented use of proceeds are attack cases. changing the embedding model can change candidate coverage, so revisions are evaluated for drift and exclusion without pretending embeddings themselves are a constitutional oracle.

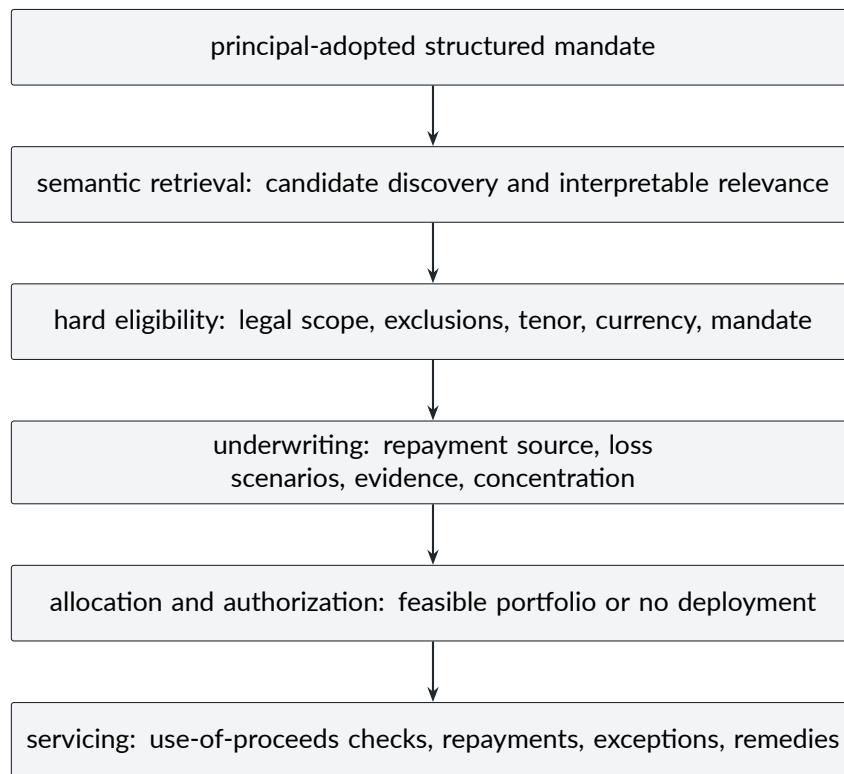


figure 13.1: semantic relevance proposes possibilities. eligibility, financial constraints, and authorization decide whether money can move.

13.3 underwriting starts with a repayment path

a credit application identifies the obligor, requested amount, repayment schedule, cash-flow source, existing obligations, collateral if any, use of proceeds, legal recourse, and evidence coverage. the lender assesses whether payments come from customer revenue, receivables, contracted grants, asset disposition, or an explicitly identified sponsor. hoped-for future fundraising is not reported as a committed repayment source.

a simple loss decomposition is expected loss $\approx$ exposure $\times$ probability of default $\times$ loss given default over a stated horizon, with dependence and timing modeled when material. uncertainty in each estimate is retained. sparse early data should not be converted into false precision, especially when several firms depend on the same platform, purchaser, or subsidy.

a denial or restriction records reasons a person can understand and contest. where applicable, external credit law imposes notification obligations; u.s. regulation b illustrates why a model's opacity does not eliminate the need for reasons.[20] the exact legal requirements depend on the transaction and jurisdiction and must be reviewed before origination.

13.4 allocation as constrained optimization

let x_{ij} be the capital allocated from mandate i to borrower j , and $e_{ij} \in \{0, 1\}$ indicate hard eligibility. an illustrative optimizer balances intent alignment, expected net return, concentration, and loss risk subject to funding, borrower, maturity, and liquidity limits. ineligible allocations satisfy $x_{ij} = 0$. no objective score can compensate for violating a hard exclusion or exceeding authority.

an empty feasible set is a valid outcome. the system can return capital, hold it in an approved low-risk state subject to disclosed costs, or request explicit amendment of the mandate. it must not reduce

loss protections silently, route capital to a weakly related theme, or present an unattainable target as funded.

optimization is performed off-chain in the reference architecture; a smaller deterministic verifier checks the proposed allocation against approved constraints before funds move. a signature from an underwriting service proves who supplied the estimate, not that the estimate is correct. independent review, outcome tracking, exposure caps, and liability arrangements address that residual risk.

13.5 three distinct ways to bear credit losses

direct participation. investors receive agreed shares of collections from specified loans after disclosed costs and priorities. borrower defaults reduce the investors' recoveries. no general treasury guarantee is inferred.

treasury-backed development debt. investors lend to an identified institutional issuer, which deploys funds within a thematic mandate. borrower losses affect that issuer's balance sheet, while investor payment rights remain defined against the issuer. the source of the issuer's additional repayment capacity must be disclosed.

limited first-loss support. an authorized support fund supplies a reserve that absorbs specified losses before senior investors. with \$10,000 of reserve and \$90,000 of senior capital, \$100,000 can be deployed only because investors actually supply the senior money. this is not money creation. for portfolio principal loss L and reserve R ,

$$L_{\text{reserve}} = \min(L, R), \quad L_{\text{senior}} = \max(0, L - R), \quad (13.1)$$

subject to the actual legal waterfall, fees, recovery costs, and timing. losses above the reserve reach senior capital. a small reserve against highly correlated borrowers can be exhausted quickly.

13.6 match financing instruments to mission economics

commercial loans, concessional loans, public grants, and investor donations are separate instruments. if a social-service program produces public value but no reliable collectible cash flow, it may need grants or an identified outcome payer rather than a bond promising autonomous repayment. if an investor accepts a lower expected return to support a mission, the concession is disclosed rather than hidden inside a model's optimistic forecast.

the institution also distinguishes intent alignment from additionality. financing an already-funded project may satisfy a sector preference while creating little additional activity. producing a desired outcome and causing that outcome are different claims. outcome evaluation specifies the counterfactual assumptions and does not automatically infer impact from money moving through the preferred sector.

13.7 servicing, changes, and failure

a funded mandate continues after disbursement. servicing records repayments, covenant tests, evidence freshness, use-of-proceeds exceptions, defaults, recoveries, and amendments. a borrower can correct inaccurate data. material mission drift prompts the response specified in the agreement, which may be notification, consent, suspension of new tranches, or a contractual remedy.

financial distress is not automatically fraud. automatic acceleration or public reputation penalties can destroy otherwise recoverable value. the reference design routes disputed defaults, restructuring, and enforcement through defined procedures. a frozen smart contract cannot seize physical collateral without the corresponding legal and operational capability.

13.8 the founder-funded experiment

jacob's proposed \$10,000 is a bounded way to test whether intent can connect to real productive claims and repayments. the example in appendix F tracks principal, interest, expenses, and default under a direct-participation structure. it is not an executed transaction or an offering. the first useful result may be discovering that no appropriate borrower qualifies, or that grants rather than loans fit the available projects.

14 economic measurement and bounded stabilization

14.1 a ledger is not a complete set of national accounts

the graph can support timely measures of recorded payments, commitments, loan performance, business registrations, and treasury cash. it does not directly observe all production, employment, household welfare, external liabilities, or off-chain activity. missing coverage is a first-class statistical fact.

in national accounting, value added is output less intermediate inputs, and gross domestic product has a defined production boundary.[21, 22] a voluntary protocol has no automatic geographic production boundary. an initial indicator should therefore be called something like *covered protocol-associated value added*, with explicit inclusion, valuation, and consolidation rules, rather than “gdp” without qualification.

loan disbursements, repayment of principal, asset trades, transfers between accounts, and wash transactions are not production merely because they increase transaction volume. an intermediate purchase and a final sale cannot both be counted as separate final output. actual service fees can represent output, but must be separated from the principal they accompany.

14.2 a statistical contract for every indicator

each published metric includes its definition, unit, population or coverage set, time interval, data vintage, inclusion rules, consolidation logic, source quality, missingness, uncertainty, privacy treatment, and revision policy. values computed from the same ledger may differ because their economic boundaries differ; the metadata explain the difference.

for an observed business j , an illustrative value-added estimate is

$$\widehat{VA}_{j,t} = \widehat{\text{output}}_{j,t} - \widehat{\text{intermediate consumption}}_{j,t}. \quad (14.1)$$

aggregation requires consistent periods and units, elimination of double counting, and treatment of imports and production outside the chosen boundary. a dashboard does not infer missing production simply from available token transfers.

employment is reported as a documented headcount or hours measure with declared evidence, not as a count of agent accounts. price indices require a defined basket, quality adjustment policy, and coverage. protocol registration is not the same as legal business formation. nominal growth, real growth, and a change in the number of recorded participants are different statistics.

14.3 a truthful economic newscast

the proposed newscast is a presentation layer over verified statistical releases. each statement links to the metric, vintage, methodology, and permitted evidence. a deterministic fact packet supplies numbers and units; generation produces narration within a validation step that checks that no figure, comparison, or certainty was invented.

illustrative bulletin; fictional values

“in the seven days ending at the stated reporting cutoff, verified customer receipts among 24 reporting businesses were 3.2% above the preceding seven-day window. reporting coverage changed, so this is not an estimate of economy-wide real growth. two loans entered the disclosed delinquency category. no liquid five-year institutional bond benchmark is available.”

the system should be able to report “not available” without filling silence with financial theater. an initial \$10,000 experiment will not support a credible macroeconomic yield curve or hourly gdp estimate. a quote from one thin trade is identified as a trade, not converted into a robust market benchmark. seasonality, revisions, and small denominators accompany the headline where material.

personal records need not be publicly reconstructable merely because a statistic is auditable. independent verification, aggregate proofs, and protected data access can support accountability. small cohorts and repeated releases can leak identities; privacy controls apply to the release process, not only the raw database.

14.4 measurement, estimation, and intervention operate at different speeds

a high-frequency ledger does not justify a high-gain economic controller. an institution sees noisy indicators of a partly observed economy, and its actions have delayed effects. define latent economic state x_t , observations y_t , estimated state $\hat{x}_t$, and authorized intervention u_t :

$$x_{t+1} = f(x_t, u_t, \varepsilon_t), \quad (14.2)$$

$$y_t = h(x_t) + \eta_t, \quad (14.3)$$

$$\tilde{u}_t = \pi(\hat{x}_{t-d}, \theta), \quad (14.4)$$

$$u_t = \text{admitExact}_{C,L}(\tilde{u}_t, S_t). \quad (14.5)$$

$\text{admitExact}_{C,L}$ is a partial admissibility function, not a geometric projection. it returns the exact proposed action only after all kernel, law, fiscal and commit-time checks are satisfied; otherwise it returns a tagged refusal or bounded-review result and no intervention. authorized actions can be discrete, non-convex or absent. no nearest substitute is chosen and an empty set creates no power. a policy may itself search a finite permitted menu only if the grant names its objective and tie rule in advance; changing an investor’s economic intention requires that mandate’s authorization. d captures observation and decision delays. the equations are a modeling frame, not evidence that suitable f , h or π have been identified.

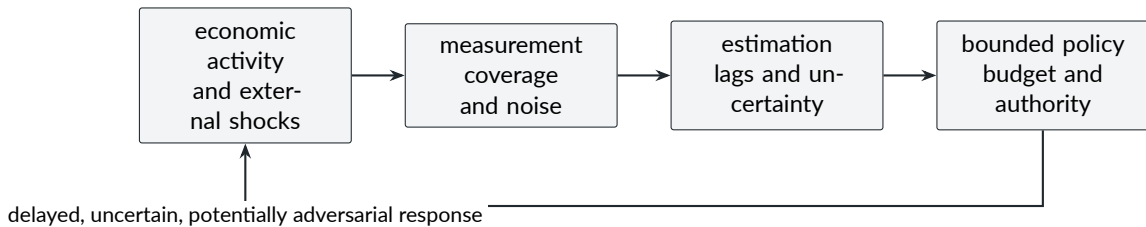


figure 14.1: faster observation can improve awareness while excessive intervention speed amplifies noise or feedback instability.

14.5 automatic stabilization is a funded contingent program

a law can authorize a temporary expansion of an already funded support envelope when several independently measured conditions persist. smoothing, hysteresis, confidence thresholds, rate limits, caps,

and expiry reduce specified failure modes. they do not eliminate model error or prove macroeconomic stabilization.

when activity falls, fee revenue may fall while support demand rises. borrowing can finance a temporary gap only if lenders supply funds at acceptable terms. spending borrowed usdc reallocates existing settlement assets; it does not create dollars. net effects on activity depend on the source of funds, recipients, import leakage, substitution, confidence, capacity constraints, and crowding out. a fiscally constrained small voluntary network cannot assume the policy space of a monetary sovereign.

a reserve buffer can provide pre-funded response capacity. a public guarantee can mobilize private lending but also increases contingent liabilities. an automatic lending-rate rule changes the institution's offers, not the market's willingness to lend or borrow. the treasury may become rationed when its offered rates fail to compensate risk.

14.6 goodhart effects and endogenous behavior

once a metric controls funding, participants have an incentive to change that metric. recorded volume can be inflated, applications can be delayed to meet a downturn trigger, or economic activity can shift off-protocol. the goodhart taxonomy distinguishes several ways a proxy can fail under optimization.[23] a rule tested on passive historical data may behave differently when participants know and exploit it.

controls include multiple evidence channels, related-party checks, delayed eligibility snapshots, independent audits, and evaluation of outcomes not directly rewarded by the rule. these controls themselves create costs and false positives. a mission fund must not optimize a public activity score so aggressively that it excludes valuable unmeasured work or rewards meaningless throughput.

14.7 model governance and a staged evidence burden

policy models publish versions, input vintages, assumptions, sensitivity ranges, and known failure cases. retrospective simulation uses only information available at the simulated decision time. stress tests include correlated defaults, missing data, stale prices, provider failures, strategic behavior, and changes in economic regime.

the deployment sequence distinguishes historical replay, shadow recommendations, capped reversible pilots, and only then any automatic intervention. an intervention is a program funded from the commons and adopted by rfc; it spends the commons' own money and never sets prices, taxes particular holders or restricts others' exchanges (kernel k5). models cannot expand their own scope or reinterpret uncertainty as authority. uncertain results may justify doing nothing, requesting more evidence, or returning the policy for deliberation.

15 agents, settlement and operational continuity

15.1 delegation is a bounded capability

an agent acts for an identified principal through a capability specifying permitted actions, accounts, recipients or recipient classes, amount limits, cumulative limits, time limits, required evidence, and revocation conditions. permission to negotiate is not permission to sign. permission to draft an rfc response is not permission to submit one: a response is the person's own act (section 6.2). permission to service loans is not permission to lend.

```
capability = {
  principal, delegate, parent_capability,
  allowed_calls, resource_scope, recipient_constraints,
  per_action_limit, aggregate_budget, budget_window,
  not_before, expires_at, revocation_epoch,
  required_evidence, approval_conditions,
  no_further_delegation_unless_explicit
}
```

a delegated capability cannot exceed its parent. budget consumption is checked cumulatively, so splitting one purchase into many small requests does not evade the cap. revocation invalidates descendants. concurrent spending attempts must not each observe the same unspent budget and both succeed.

an authorization checks the proposed effect, not the natural-language request. the user sees recipients, assets, fees, irreversible consequences and the governing scope. a signature for one purpose cannot be replayed as a response, a case filing or a payment, because domains, action types, versions and nonces are distinct.

15.2 agents are principals only when the law says so

a model may recommend, draft, simulate or execute within its principal's capability. an engine seat answers questions and holds no authority (chapter 7). an agent becomes a person with a voice only if the law recognizes its kind (section 5.1); that gives it one voice per lineage, not ownership of anything, and not authority over anyone.

liability is not settled by blaming the model. agreements identify the principal, operator and relevant service providers; a case names a respondent, and section 5.5 decides whose act it was. external law may assign responsibilities independently of the contract's labels. consequential physical actions require the permissions and safety procedures of the system that performs them.

command agi connects the contract to computers, software services, email, telephone, sms, physical mail and physical-system interfaces. a contact channel is not an authentication channel. replying to an unauthenticated message is not an rfc response.

15.3 reference settlement implementation

solana is a candidate substrate, not a kernel requirement. its account model separates stored state from executable programs, and program-derived addresses can hold program-controlled accounts.[24, 25]

heavy documents, evidence, embeddings and model runs stay outside consensus execution; determination hashes and evidence roots identify them. an off-chain indexer materializes the graph; independent indexers must reproduce canonical state, and a stale or malicious indexer gains no permission to change it.

15.4 the contract registry: no key above the contract

kernel article k3 forbids any key whose holder can change the contract outside an adopted rfc. program upgrade authority is exactly such a key.[26] the target design therefore has one on-chain registry account per contract holding: the kernel hash; the contract version; the law hash and the engine hash; for the classifier and deliberative tiers, the ed25519 keys of the seats' signing services with their independence groups and the quorum in groups; the genesis manifest; the hash of the last adopted decision record; and the program's upgrade authority, held by a program-derived address of the registry itself. every change is signed over one fixed 181-byte adoption message — a magic string, the kind (law, engine or upgrade), the registry's version, law hash and engine hash before the change, the rfc's identifier, the **decision-record hash** and a payload naming what the change installs.

- **genesis.** `initialize` takes a manifest hash and refuses unless the supplied kernel, law and engine hashes and signer sets hash to it, *and* it equals the manifest constant compiled into the program. a deployment's identity is then checkable by anyone, and whoever calls `initialize` first cannot install signer sets of their own: a build with no manifest refuses to initialize.
- **law.** a law adoption needs a quorum of groups of the current engine's top-tier seats.
- **engine.** an engine adoption installs new seat keys and needs a quorum of the outgoing engine's top-tier groups *and* a quorum of the proposed engine's (kernel k4).
- **code.** a change of the programs' code is a change of the engine (k4): an upgrade adoption names the engine the new code makes, needs both engines' quorums like an engine adoption, and is only *scheduled*. it executes after a timelock of seven days fixed in the program, during which a law adoption leaves it standing and an engine adoption voids it. an unchanged kernel hash is not evidence that the kernel's protections are unchanged when the code that enforces them can change; this makes such a change as hard as replacing the engine, and visible a week before it takes effect.
- every adoption carries the hash of the rfc's decision record, refuses an all-zero hash, and stores it, so anyone can fetch the public record and check it.
- escrow settlement verifies a frame against the registry's current seat keys for the frame's tier, not against a committee chosen by the funder; a frame counts as a top-tier determination only when a quorum of deliberative groups signed it. an appeal frame from a higher tier, signed by seats that did not sign the frame it overturns, supersedes it within the appeal window; the top tier is reheard once by fresh seats.

the assurance boundary. what the chain verifies is signatures, group quorums, the dual-engine rule, continuity with the state the adoption was decided against, the genesis manifest and the timelock. it does *not* verify that the rfc existed, stayed open for its window, received valid responses or produced the stated outcome: the signers attest to that, and the decision hash makes their attestation checkable against the public record after the fact. this moves the predecessor's committee and upgrade-key risks (section 18.5) into one place that only the engine's signatures can move; it does not remove trust in the seats' signing services, since a seat's key signs what its service says the model answered. the registry program and the escrow's registry frames are built and pass their tests; the reference module mirrors the registry's layout, adoption message and genesis manifest byte for byte (`registry-abi.ts`). neither program is deployed (chapter 18).

15.5 one transaction per movement: op claims

the hosted service keeps its ledger in one sqlite database and commits every contract movement — an rfc’s deposit and decision, a freeze, an execution, a pool epoch, a mandate, a drawdown, a repayment, a bond sale — as one batch, which the database runs as one transaction. a planner reads the state, computes the movement with the pure core, and emits statements in a fixed order:

1. an **op claim**: insert a fresh identifier only if the state still is what the planner read (its revision, the contract’s version, an offer still open, a claim’s paid amount);
2. debits, each conditional on the claim, on its own identifier not existing, and on the holder’s balance *at commit time*;
3. credits, pool rows and state changes, each conditional on the claim and on every debit having landed.

every ledger row’s identifier is derived from the subject and the operation (`cc:<case>:judgment`), so the same movement always names the same rows. of two concurrent attempts exactly one claim lands; a balance spent meanwhile makes the debit, and therefore everything, write nothing; a crash leaves nothing half-done; a replay finds its rows and its marker. rows that the sweep and a request can both write — a case, an rfc, a facility — carry a revision, and every update is conditional on the revision read, so a sweep holding a stale copy can never overwrite an appeal or a submission a request wrote meanwhile.

due work. every rfc, case and facility carries the instant it next has something to do — a window closing, media to render, a question to retry — and the scheduled sweep takes the earliest due first, a bounded number per run. a matter waiting on a party has no due instant, so it never occupies the sweep while an actionable one waits. version 0.4 selected the ten least recently updated cases, so ten cases that could not move could starve every other.

15.6 keys are powers, not symbols

threshold signing reduces reliance on one key only under assumptions about signer independence. the kernel’s operator rule (k10) applies to signing services too: a quorum drawn from one cloud account, one recovery administrator or one firmware dependency is one group, whatever the key count. public custody ceremonies are evidence of a procedure, not proof that a key was never copied. handover means rotating keys and revoking predecessors through an engine adoption, never trusting a promise that old copies were deleted.

15.7 safe interaction with the external world

a single chain transaction may atomically update several on-chain records. a phone call, bank payment, machine shipment or posted letter cannot be included in that transaction. the implementation uses an authorized intent record, an idempotency key where supported, execution attempts, receipts, reconciliation and a declared compensation path.

the status model distinguishes requested, accepted, partially completed, completed, unknown and disputed. an uncertain external result is not automatically retried if a duplicate would be harmful. exactly-once external execution is not claimed without cooperation from the receiving system. a supplier’s outage does not grant an agent permission to switch counterparties, disclose additional information or exceed its budget.

15.8 data availability and recovery

a content hash proves integrity only when the corresponding material remains available. the kernel, law, engine, determinations and public receipts need replicated archival storage and independent verification. private evidence needs access-controlled backups, retention rules and a tested recovery process. losing an encryption key can make a perfectly preserved ciphertext useless.

normal, degraded and safe modes permit different actions. a chain halt, an unavailable engine tier or a stale price feed can stop new commitments while preserving read access and evidence. blocking *new* obligations is different from withholding payments already owed. recovery exercises cover signer loss, corrupted indexers, compromised clients, database restores, inconsistent external receipts and settlement unavailability. restoring yesterday's database does not erase intervening obligations.

15.9 open implementation

the specification, the contract core and its tests are independent of command agi's hosted product. alternative clients can display exact rules, submit responses and file cases. a commercial product update does not change the law, the engine or who is a person. the company is a participant like any other: when it operates an engine seat, its seats never sit in its own cases (k7).

open source is necessary for independent inspection but does not prove correctness, decentralization or maintenance. reproducible builds, version pinning, dependency review and independent audits supply further evidence. maintainer access to deployment infrastructure is a power and is recorded as one.

15.10 commit-time checks

an authorization is bound to a state version and a finite reservation. at commitment the gate rechecks the capability and its revocation, the state hash, recipient and asset scope, unused replay identifiers, evidence freshness, resolved appeals and unspent budgets. it atomically consumes the reservation and nonce and appends the intended effect. a conflicting update returns stale-state, not a best-effort execution under old authority. an external action is then dispatched and reconciled separately; its failure does not erase the reserved obligation or justify a duplicate attempt.

16 fault tolerance: a record anyone can check

the contract should outlive whoever runs its sweep. kernel article k3 forbids a key above the contract, and k11 makes its record public, but a record is only as good as the honesty of whoever wrote it. version 0.5 adds three things that let anyone check the record without trusting its writer — attested answers, a hash-chained log with anchored heads, and a replica that recomputes every derived outcome — and states exactly what each one leaves to trust.

16.1 attested answers

a seat may name an **attestor**: an ed25519 key and the https endpoint of a small service anyone can run, holding its own provider credential and its own signing key (deployments/platform/cloudflare/attestor). the attestor is part of the engine, so naming, changing or removing one is an engine rfc. for an attested seat the executor asks through the attestor instead of calling the provider, and the answer counts only when it carries the attestor’s signature over the canonical attestation message:

```
cc:attestation:v4 || canonical(seat, model, protocol, kind, subject,  
                                label, confidence, output, at, nonce).
```

subject is the digest of the evidence package the attestor itself computed and sent (ask) or the media item’s sha256 (render); **output** is the digest of the reason or of the rendered text; **nonce** is fresh randomness. the reference check askAttestationRefusal accepts a correct signature (null) and refuses, each with its own reason, an answer whose label was changed after signing (wrong_answer), one signed over another package (wrong_subject), one signed by any key but the seat’s attestor’s (wrong_key), one with no attestation (missing) and one whose signature does not verify (bad_signature). a refused attestation is an error answer, never a vote. renderings are attested the same way; the attestor fetches and checks the media’s bytes itself.

a seat without an attestor still answers, recorded attested:false: its answer rests on the word of the executor that called it. **no genesis seat names an attestor**, because the platform runs every genesis seat; attestation removes nothing from the trust path until an engine rfc names attestors run by others — a seat’s own operator, ideally. what an attestor attests is that its provider returned that answer to that package, not that the provider is right.

the executor sends an attestor’s access token only to the origins it was issued for, and a proposed engine’s attestor urls receive none (section 7.9).

16.2 the transparency log

every act of the contract is appended to one public, hash-chained log: versions, rfcs opened, responses confirmed (anonymous ones included, by response identifier only), interpretations, contests, decisions, determinations, notices, cases filed, submissions, appeals, judgments, executions, pool epochs, mandates, facilities, payments, transfers, defaults and personhood statements. an entry is

$$e_n = (n, h_{n-1}, \text{kind}, \text{subject}, \text{body digest}, \text{at}), \quad h_n = \text{sha256}(\text{cc:log:v4} \dots e_n), \quad h_0 = 0^{256}.$$

it carries the digest of what the act committed, never the record and never who: a case’s evidence and its parties stay where k11 keeps them, and rfc voices appear only as per-rfc pseudonyms. an append chains

to the head it read and retries when another append won, so concurrent writers produce one order; an act's idempotency key lets the sweep append again an act whose live append was lost (`reconcileLog`). in the reference fixture a log of 4 entries verifies (yes); rewriting one body digest shows 1 problem, dropping an entry 2, and a page verifies from the entry before it (yes). the log is served at `/public/contract/log` with its head.

16.3 checkpoints

a hash chain makes rewriting detectable only to someone who kept an earlier head. so whenever the head moves, at most every ten minutes, the sweep records a **checkpoint** (n, h_n) and anchors it on the registry program's permissionless `checkpoint(seq, head)` instruction: 48 bytes of instruction data (discriminator `d5c813ccf08fb8fc`, the sequence as a little-endian u64, the head), stored in a 97-byte account at the program address derived from "checkpoint" and the submitter's key (for a submitter of `0x42×32, BngaGbnE\protect\TU\tellipsis\}F×RX`, bump 253). the program refuses a zero head (`ZeroHead`) and a sequence that does not increase for its submitter (`NotIncreasing`). anyone may submit checkpoints; a verifier trusts the submitter whose key the contract publishes as its anchor, and holds the log to every head that key anchored: a checkpoint that does not match the log's entry at its sequence is a mismatch (1 in the fixture's forged case). an anchored head makes showing two different histories to two audiences detectable. the anchor writes only to devnet or testnet and refuses mainnet; until the registry is deployed, every checkpoint is recorded `not_anchored` with the reason.

16.4 the replica

`scripts/contract/replica.mjs` fetches the public record of a deployment, or reads an export, and recomputes:

- every contract version: its engine and state hashes, the kernel it names, the engine's admission against the kernel, and that versions are contiguous;
- every rfc: its identifier from its content, its pinned engine, its tally from the published interpretations, its decision from the tally and the concurrence, and the decision hash;
- every published determination: its aggregation, bands and outcome — including that a tie is unresolved —, every answer's delivered digest and attestation signature, and its identifier (`verifyDetermination`);
- the log's chain, its head and its coverage of every version, rfc, decision and determination, and every checkpoint against the log — and, given an rpc endpoint, against the transaction that anchored it;
- given a party's token, that party's cases: each finding against its determination, each such determination, and each judgment from its findings.

it prints a verdict per item and exits non-zero on any mismatch. it *cannot* check what is not published or not recorded: why a seat erred or was unavailable; whether a hosted model's answer is right, as opposed to signed by its attestor; a tier-1 head's output without the embedding it read; who opened an rfc, and so whether a deposit's return was due; another party's case; or any movement of money, since the credit ledger is not public. the replica moves the honesty of the record from trust to computation for everything it can see, and says plainly what it cannot.

16.5 the assurance boundary, by component

component	checkable by anyone	still trusted
kernel, law, engine an rfc's outcome	their hashes, the kernel text, admission (replica, registry) tally from published interpretations, decision, decision hash (replica)	nothing beyond the hash function the interpretations themselves, unless the seats that made them are attested
a determination	aggregation, outcome, delivered digests, attestation signatures (replica)	unattested seats' answers (every genesis seat today); providers' correctness
anonymous voice	each proof against the pinned root and scope (worker; anyone with the proof)	semaphore's circuit and setup; network metadata the operator sees
history	hash chain, checkpoints against anchors (replica)	acts never logged, until reconciliation or a party notices
a case	a party's own case, recomputed with its token	other parties' cases; the evidence's authenticity beyond its digest
money	the plan each execution recorded (parties)	the ledger itself, which is not public
registry adoptions	signatures, group quorums, base state, timelock, genesis manifest (chain)	the seats' attestation that the rfc's procedure ran
other chains	the same adoptions and frames, the same keys (section 17.3)	that someone relays adoptions; a stale mirror lags, it never leads

the platform today runs every genesis seat, holds every seat key and runs the sweep. the components above are what turn that from a matter of trust into a matter of evidence as others take those roles; none of them claims that others have taken them yet.

17 one trust root, many chains

the registry (section 15.4) is the contract's only authority on any chain. everything else the contract does on a chain — releasing escrowed funds, authorizing a machine or a service to act, following the contract on another chain — reads the registry or mirrors it, and never holds a key or a committee of its own. this chapter describes the two programs that version 0.5 adds to that rule.

17.1 hub and spokes

the solana registry is *canonical*. a spoke — the escrow and the command notary beside it, a mirror and a mirror escrow on any evm chain — trusts exactly the registry's seats: the same ed25519 keys, grouped by independence, with the same group quorums, the same dual-engine rule for engine and code changes, and the same seven-day timelock. a spoke adds no signer of its own, so there is no second attestor set that could be rotated on one chain and not the other, or held by someone else. the price of that choice on evm chains is gas (section 17.3). a spoke that is not kept current is stale, never wrong: an older engine's frames keep working until the newer adoption reaches it, and anyone can relay an adoption, since the signatures carry the authority and the relayer carries none.

17.2 the command notary

command agi's machines and services act under *command authority*: who may cause which action on which target, within what budget, until when. the command notary records that authority on chain so that an executor can prove a command was authorized, by whom, and that it was not revoked or expired. it moves no funds. version 0.4 left it with its own committee chosen at deployment; version 0.5 folds it into the registry's workspace and gates it by the registry:

- authorizing, delegating and revoking need a top-tier quorum of the registry's current seats, counted by independent group, over a 176-byte message (`cagi-cmnd:v2`) that binds the registry's version and engine hash; a binding to any other registry state is refused, so a superseded engine cannot authorize and a revocation is judged by the engine current when it is made;
- a delegation never widens: a child's expiry, spending cap and delegation depth are bounded by its parent's, and revoking a parent stops its children;
- an optimistic authorization opens a challenge window before it is final.

the program passes its rust tests (appendix J); it is not deployed, and nothing yet requires an executor to consult it.

17.3 the evm mirror

`ContractMirror` follows the registry on an evm chain. its constructor takes the same genesis fields as the registry and refuses unless they hash to the published genesis manifest. it applies an adoption only when the same 181-byte adoption message (`cagi-adopt:5`) carries a quorum of groups of the current top-tier seats' ed25519 signatures — and, for an engine or a code change, of the incoming seats' too — and only against the base state the adoption was decided against. signatures may accumulate across transactions. a code change waits out a 7-day timelock; a law change during it leaves it standing, an engine change voids it. the contract is immutable and has no administrator. `MirrorEscrow` holds erc-20 funds and releases each tranche once, on the same frame the solana escrow reads, signed by a quorum

of the tier the mirror names; a decision with a lower sequence cannot replay after a higher one, and after expiry the remainder returns to the customer. the solana escrow's appeal ladder is not mirrored: a tranche settled on an evm chain is final.

why ed25519 in solidity. the mirror verifies the seats' own ed25519 signatures with a sha-512 and curve implementation in solidity, checked against the fips sha-512 vectors, rfc 8032's test vectors,[11] random signatures and forgeries. the alternative — giving every seat a second, secp256k1 key for cheap ecrecover — would make a second attester set that could diverge from the registry's. the cost, measured by the mirror's own tests on 3 october 2026 (deployments/platform/blockchain/README.md), is about 0.8 million gas per signature (at most about 0.96 million), so a law adoption with two signatures costs about 1.95 million gas, an engine change with two quorums of two about 4 million, and settling a frame of two about 1.3–1.8 million: reasonable on rollups and other low-fee chains, expensive on ethereum's mainnet. the reference module builds the mirror's message and frame bytes, and the mirror's tests read fixtures that the module's tests regenerate and require unchanged, so the solana, evm and typescript encodings cannot drift apart.

17.4 what other chains add, and what they do not

a mirror lets the contract coordinate activity — escrow, settlement, command authority — on chains where its participants already are. it adds no trust: a mirror is exactly as trustworthy as the registry's seats and no more. it adds availability only in the sense that a copy on another chain keeps working when one chain halts, for the adoptions that were relayed before the halt. it does not make the contract's procedure verifiable on chain; the seats' signature over the decision hash remains the attestation, checkable afterwards against the public record (section 16.5). neither the mirror nor the mirror escrow is deployed, and nothing relays adoptions yet.

18 implementation status, release and migration

18.1 what exists

everything below is in this repository. appendix I lists each rule with the test that exercises it.

the contract core (packages/domain/core/src/contract): pure typescript with no i/o — canonical hashing; the kernel, its floors, its protected interests and its pinned hash; people, proofs, voice resolution, signed personhood issuance for kinds other than human with monotonic statements and revocation tombstones, and the human document-identity key; law with duty elements, admitted media and the law check; the engine, seats, pins, renderers, the engine check with appeal capacity and rendering coverage, grouping and seeded panels; integer-only mlp heads; determinations over canonical text packages with delivered digests and decided, unresolved and unavailable outcomes; evidence intake and renderings; rfcs with pinning, receipts, interpretation against actual changes, score voting, per-interpretation contest windows, seeded concurrence samples, the two-engine order and its fallback, and the decision hash; justice from filing refusals and screening through service, instances, element findings, final-offer figures, remedy justification, appeals as a stage, unprocessable closure, reopening, the public projection and execution plans with loser-pays both ways; pool allocation epochs; the credit cycle; state, adoption, supersession and law history; the genesis law and engine; anonymous voice — commitment trees pinned per rfc, nullifier voices, the one-count rule and per-action messages; seat attestations; the transparency log and checkpoint consistency; the determination replica (`verifyDetermination`); and byte-exact mirrors of the on-chain registry (genesis manifest, adoptions, checkpoints), the command notary’s message and the evm mirror’s encodings. every contract number in this paper is generated by it.

the hosted service (the api worker): the contract tables (migrations 0539–0543); anonymous responses, contests, media and lookups under proofs verified by its own `groth16` check, with commitment registration and the public commitment tree; attested seats asked through their attestors; every act appended to the public log, its head checkpointed; the contract state, written at genesis on first read and published with the kernel, law and engine; personhood for humans from the platform’s identity verification under the document-identity key, and signed proofs for other recognized kinds; media upload stored once by `sha256`; a contract inbox for notices; the rfc lifecycle — open, interpreting, contest, deciding, adopted or rejected — with receipts and confirmation; cases from screening and service through instances, appeals, interim freezes and execution; pool epochs; mandates, facilities, servicing, bond offers and defaults; every movement one op-claimed batch (section 15.5); a due-work sweep on a schedule. the engine executor binds each operator’s credential to its hosts, calls each seat exactly as pinned through one adapter per wire protocol, verifies media against its digest before rendering, and uses a candidate executor that renders nothing for proposed engines. case records are read only by their parties; a case’s determinations are never served publicly. borrowing, lending and bond trades need a person’s own credentials, never an agent key. service-agreement disputes the agreement’s evaluator cannot decide become agreement cases whose final judgment settles the escrow; no person resolves a dispute. a person excluded by a criminal judgment cannot open new commitments.

removed with the old institution: the four branches, governance proposals and seats, the policy-parameter register, the constitutional holds, the juror pool, the contract roster, the enacted fee schedule, the governance-selected integrity formalism, and reserve countersignature by a branch.

on chain (deployments/platform/blockchain): in one solana workspace, the registry program of section 15.4 — genesis manifest, decision hash, dual-engine code upgrades behind a seven-day timelock,

permissionless checkpoints — the escrow’s frames read from it, with appeals climbing tiers and the top tier reheard once by fresh seats, and the command notary gated by the registry (section 17.2); on evm chains, the mirror and the mirror escrow (section 17.3). the solana programs pass their rust tests and the evm contracts their foundry tests. none is built for or deployed to any cluster or chain.

the attester (deployments/platform/cloudflare/attester): the service that runs a seat and signs its answers (section 16.1), sharing the worker’s provider adapters; built and tested, run by nobody yet.

tools: scripts/contract/replica.mjs recomputes a deployment’s public record (section 16.4); script s/contract/canary.mjs asks every genesis seat and renderer one known question with a deployment’s credentials and writes the run (appendix N); scripts/contract/train-head.mjs trains a tier-1 head from labelled examples, quantizes it, calibrates its confidence table conservatively (the wilson 95% lower bound of held-out accuracy per margin, made monotone), checks the integer head on the held-out set and seals it; scripts/contract/pin-weights.mjs computes an open-weight pin from a model repository at one commit.

deployment. version 0.4 of the hosted service ran on the platform’s development and production environments from 2 october 2026 with no rfc or case filed under it. version 0.4.1, a different kernel, replaced it on both on 3 october 2026, and its genesis was published under the new kernel after the old one was withdrawn — only because nothing had happened under it. version 0.5 keeps that kernel and adds anonymous voice, attestors, the log, checkpoints, the replica, the command notary and the evm mirror; it was deployed to both environments on 3 october 2026, and because its genesis engine pins the router seat to one host (a different engine hash), its genesis too was re-published only where nothing had happened. the transparency log keeps the withdrawn genesis as history beside its replacement, and the replica (chapter 16) recomputed both environments’ records and agreed. no on-chain program is deployed. the live canary of appendix N reached every genesis seat and renderer through the deployed adapters on 3 october 2026.

not done: no rfc or case has been decided by live models end to end, so no determination of record has yet been answered by the genesis engine; no seat names an attester, so the platform remains in the trust path for every answer; no checkpoint is anchored, since the registry is not deployed; anonymous proving has not run in a person’s browser against a deployed host, the proving artifacts are fetched from semaphore’s host, and no relay hides a person’s address; no mlp head for a contract question has been trained or adopted; no issuer for a kind other than human exists; the platform holds the seats’ signing keys and runs the models, so determinations are auditable from their records but hosted seats cannot be re-run by others; controlledBy is empty in the genesis engine, so a seat’s exclusion from its controller’s cases acts only on operators named as parties until an engine rfc records control; the genesis state is written by the service rather than adopted by a genesis rfc; no seat has been evaluated on contract questions; no jurisdiction has been reviewed; the user interface for credit is the api alone.

18.2 threats against what is built

every threat of appendix H names the rules of appendix I that control it; the table sets each threat against the weakest status among those rules. a threat is only as well answered as its least tested control. the table is generated from threats.json and guarantee_status.json, so it cannot say more than the register does.

id	threat	rules that control it (status)	weakest
th-01	one provider runs several seats to outvote the rest	s-07 (tested), s-10 (tested)	tested
th-02	a provider changes or retires a model behind its id	s-07 (tested), s-12 (tested)	tested

id	threat	rules that control it (status)	weakest
th-03	an incumbent engine entrenches itself, or a captured engine installs itself	s-23 (tested)	tested
th-04	the engine rules instead of interpreting	s-22 (tested)	tested
th-05	interpretation reads support into responses that express none	s-20 (tested), s-22 (tested)	tested
th-06	one person, many voices	s-03 (tested)	tested
th-07	intensity inflation, or agenda games with clone options	s-18 (tested), s-21 (tested)	tested
th-08	a majority uses law against a minority's holdings	s-05 (tested), s-06 (implemented)	implemented
th-09	courts used to harass	s-25 (tested), s-34 (tested)	tested
th-10	a respondent escapes by disrupting appeals, or an appeal is lost to a late clock	s-08 (tested), s-31 (tested)	tested
th-11	a respondent hides assets in bonds	s-32 (tested), s-35 (implemented)	implemented
th-12	a victim left unpaid, or paid after others by arrival	s-36 (tested)	tested
th-13	evidence that instructs the model	s-12 (tested), s-13 (tested)	tested
th-14	a key above the contract, or a code change that alters the kernel's effect	s-39 (tested)	tested
th-15	the founder keeps a back door	s-44 (specified)	specified
th-16	a rejected rfc is reopened until people tire	s-18 (tested)	tested
th-17	a divided engine read as a decision	s-10 (tested), s-12 (tested)	tested
th-18	a proposed engine harvests credentials or private evidence	s-14 (tested), s-23 (tested)	tested
th-19	seats decide on different evidence; media breaks quorum	s-13 (tested), s-15 (tested)	tested
th-20	money moved twice, or half-moved	s-32 (tested), s-38 (tested)	tested
th-21	a remedy list that multiplies recovery	s-30 (tested)	tested
th-22	the same matter litigated until the respondent gives up	s-26 (tested)	tested
th-23	a revoked proof replayed	s-04 (tested)	tested
th-24	stuck matters starve the scheduler	s-38 (tested)	tested
th-25	an agent commits its owner to debt	s-37 (tested)	tested
th-26	the platform learns which rfc response is whose	s-45 (tested), s-47 (tested)	tested
th-27	one person counts twice: once signed in and once anonymously, or through a rotated identity	s-46 (tested)	tested
th-28	a forged or replayed proof speaks for someone	s-48 (tested)	tested
th-29	the operator of the sweep fabricates or alters a seat's answer	s-49 (tested), s-50 (tested)	tested
th-30	the operator rewrites, reorders or forks the contract's history	s-51 (tested), s-52 (tested)	tested
th-31	authority over a machine or service issued outside the engine	s-53 (tested)	tested
th-32	a copy of the contract on another chain diverges from the registry	s-54 (tested)	tested

of 32 threats, 29 are controlled only by tested rules, 2 by at least one rule that is implemented but not tested, and 1 by at least one rule that is only specified.

18.3 what version 0.5 adds

version 0.5 answers the owner's questions of 3 october 2026 — can a person answer anonymously and still count once; can the contract coordinate other chains; can its record survive whoever runs it — with changes that leave the kernel, and so the contract's identity, unchanged:

- anonymous voice: semaphore commitments, a tree pinned by every rfc, the nullifier as the voice, one count across both routes (section 6.3);
- seat attestors: an answer counts only with its attestor's signature, where the engine names one (section 16.1);
- a public hash-chained log of every act, its heads anchored on the registry (sections 16.2 and 16.3);
- a replica that recomputes the public record and says what it cannot (section 16.4);
- one trust root on every chain: the command notary gated by the registry, and an evm mirror that follows it with the same keys (chapter 17).

18.4 the hardening of version 0.4.1

an external review of version 0.4 traced eighteen ways ordinary code paths could contradict the institution's stated guarantees, and one deeper gap: a determination is not yet a justified judgment. each, and what replaced it:

	found in 0.4	in 0.4.1
1	a proposed engine's seats could direct platform credentials to their own endpoints, and were consulted even when the outgoing engine withheld	credentials bound to operator hosts (section 7.9); the proposed engine asked only after the outgoing concurs, through a candidate executor that renders nothing (section 6.7)
2	any authenticated caller could read any case's record	parties read the record; everyone else the public projection; case determinations never public (section 8.14)
3	execution was separate writes with random ledger ids	one op-claimed, conditional batch with deterministic ids (sections 8.11 and 15.5)
4	the top tier was final below its band; ties went to the first label	unique winner at the band, or unresolved; concurrence must be decided concur (sections 6.6 and 7.1)
5	remedies multiplied past the harm; valuation in quarters of the claim	harm items, per-item money bounds, one penalty cap over the total award, final-offer figures, remedy justification (sections 8.7 and 8.8)
6	a timely appeal could be overtaken by a late sweep	the appeal is a stage; finality requires no pending appeal (section 8.9)
7	the genesis top tier had no fresh panel for an appeal	seeded panels of exactly a quorum; two disjoint top-tier panels required at admission (sections 7.4 and 7.6)
8	one image made both model tiers non-quorate	media rendered by two independent renderers; deciding seats read text; unprocessed items marked (section 9.3)
9	seats could receive different evidence	one canonical package with unique ids; delivered digests checked per answer; media verified against its digest (section 7.9)
10	engine and procedure changed under open matters	rfcs and cases pinned to the version they began under; superseded rfcs not adopted (section 6.1)
11	contest windows fixed at opening; concurrence read the first 20 responses	per-interpretation windows from availability; whole tally record plus a seeded sample (sections 6.4 and 6.6)

	found in 0.4	in 0.4.1
12	normalization made the option set change each option's weight; interpretation read the summary	score voting; duplicate effects refused; interpretation reads actual changes (section 6.5)
13	no service; a stuck scheduler; loser-pays one way only; no matter families	service notices start the clock; due-work queue; loser pays both ways; screening and one reopening (sections 8.3, 8.4 and 8.12)
14	the pool paid claims one at a time, first come first served	allocation epochs over all open claims with persistent residuals (section 10.3)
15	code upgrades escaped the dual-engine rule; adoptions bound only hashes	code upgrades need both engines and a seven-day timelock; adoptions bind the decision-record hash; the assurance boundary is stated (section 15.4)
16	genesis and independence were declarations	genesis manifest pinned in the program; controlledBy in the engine; the unavailable-outgoing fallback (sections 6.7 and 15.4)
17	responses not bound to a deliberate act; revocations replayable; standing tied to personhood	payload-bound receipts; monotonic statements with tombstones; protection for every being, voice for persons (sections 5.1 and 6.2)
18	no implemented financing cycle	mandate, facility, servicing, transfer and default, end to end (section 10.4)
—	a determination is not a justified judgment	element-by-element findings, per-item figures, remedy justification, one identical record (section 8.6); the remaining gap is the models' quality, measured only by evaluation (chapter 19)

the review's adversarial slice is among the reference tests: an unadopted successor is never consulted before the outgoing engine concurs and cannot draw a credential; private evidence stays private; disagreement does not become approval; a timely appeal survives a late sweep; a stale or replayed execution moves no money twice; equal pool claims receive equal shares; and the \$10,000 cycle runs through servicing, a bond sale, a default, judgments against an empty respondent and an exhausted pool (appendix J). they run on the real schema with scripted seat answers; they do not measure any model.

18.5 the predecessor's gaps, and what replaces them

a static review of the predecessor at a pinned commit found the following.[32, 34, 33] each finding is history, not a statement about the current deployment.

- the escrow's frame check included the specification hash, decision sequence and issue time in signed bytes without checking them against an approved specification, a monotonic sequence or a not-before time. *replacement*: a frame is a signed determination whose identity is its content hash, verified against the registry's current seat keys.
- the escrow's committee was chosen by the funder, at most two signatures fitted in a transaction, and a challenge required a committee-signed contradicting frame, so an uncooperative committee could not be contested. *replacement*: the escrow reads the engine's seat keys from the registry, and one precompile instruction carries many signatures over one message; an appeal is a higher-tier frame from fresh seats.
- the replacement escrow program ran under a hot deployer key as upgrade authority. *replacement*: upgrade authority held by the registry and exercised only for an adopted engine rfc.
- governance had four branches whose signatures executed amendments, a stake-weighted agent franchise, and thresholds that made ratification impossible while stakeholder classes were empty.[30] *replacement*: rfcs as specified in chapter 6.

- unresolved disputes ended with a platform administrator choosing release, refund or clawback.[31] *replacement*: an agreement case under chapter 8 whose final judgment settles the escrow; the administrator route is deleted.

18.6 a guarantee is a versioned claim

an assurance names the property, the operation, the version, the environment, the evidence and its date. *specified, implemented, tested, integrated, deployed, independently reviewed* and *observed* are different statuses. a function can be tested and unreachable; an integrated caller can bypass it; deployed code can differ from inspected source. a comment saying “mainnet gated” is not a gate.

18.7 release conditions

mainnet release of the contract requires, at least:

1. the registry deployed with upgrade authority held by itself, initialized under the published genesis manifest, and the escrow verifying against it;
2. the genesis engine’s seats running with signing keys held by their operators, at least two operators per tier, and a live run of every seat on the contract’s questions;
3. a measured evaluation of every seat and renderer on the contract’s question kinds (chapter 19), published with the genesis rfc;
4. the engine’s record of who controls each operator (`controlledBy`) filled in from the operators’ own disclosures;
5. a personhood issuer under the human uniqueness rule;
6. a genesis rfc: the founder publishes kernel, law and engine, people enroll, and the genesis law and engine are adopted by the same rule as every later rfc, decided by the genesis engine — after which the founder holds nothing;
7. legal review of the instruments offered (bonds, participations, escrow) in each jurisdiction where they are offered;[27, 28, 29]
8. a protection pool funded before any bond is issued under it.

a missing condition means that part does not launch. nothing inherits readiness from another part.

18.8 the legal perimeter

tokenization does not decide whether an instrument is a security, a loan, a fund interest, a deposit-like product or a payment service. an internal registry entry does not create a corporation. agreeing to the contract’s justice does not make it a court, and the contract’s criminal matters are breaches of its own duties among those who joined, not territorial criminal law. where external law and the contract conflict, the activity is not offered.

18.9 migration

legacy agreements, escrow states, claims, governance records and the agent-franchise records are inventoried and kept as history. a dispute accepted under the old procedure is decided under the old procedure unless its parties agree to move it. old escrow layouts are not decoded as new ones. a settlement migration reconciles assets and liabilities, prevents duplicate issuance or redemption, and names one successor.

interface	question to resolve before the activity
outside capital	who issues the claim, to whom, under which offering and transfer rules?
pooled allocation	who manages the pool and which duties, registrations or exemptions apply?
lending	who is lender and servicer, and which pricing, notice and enforcement rules apply?
custody and settlement	who controls assets and bears operational failure?
external enforcement	which legal rights support judgments, collateral and insolvency treatment?
personal data	which data may be collected, disclosed, retained, corrected or deleted?

table 18.3: a legal-interface checklist, not legal advice.

19 evaluating the engine, and the limits

19.1 four questions about a judgment

factual accuracy asks whether a material proposition matches the best independent observation. rule application asks whether the duty in force supports the finding. procedural compliance asks whether notice, evidence, seat selection and appeal were valid. remedy effectiveness asks what was actually collected, paid or left owed. a true fact can support an unauthorized remedy; a regular procedure can rely on false evidence. no single accuracy score substitutes for these four records.

19.2 qualifying a seat

a seat earns its place by evaluation on the contract's own question kinds, published with the engine rfc that adds it:

- held-out cases whose facts are established independently of any model, stratified by question kind, language, evidence type and party resources;
- for each label the seat can return, its error rates, and the calibration of its confidence against the observed frequency of being right;
- adversarial cases: prompt injection in evidence, fabricated citations, honest lookalikes (disclosed simulations, good-faith errors, corrected forecasts);
- the seat's agreement with the other seats of its tier, which measures correlation, not correctness.

under independent trials, zero errors in 3,000 cases bounds the error rate below 10^{-3} at 95% confidence ($1 - 0.05^{1/3000}$); correlated or reused cases do not justify that calculation. a vendor's own calibration claim is a hypothesis for this evaluation to test.[41] the tier bands should be set from it: a band is the confidence at which the tier's measured error is acceptable for questions that may then not climb.

19.3 bands and the cost of climbing

every climb costs more and takes longer; every decisive lower tier risks an error the higher tier would have caught. the bands set that trade. their genesis values are judgment, not measurement, and the first engine rfc after evaluation should replace them. a head at tier 1 is adopted only with a held-out evaluation on the exact question kind it answers.

19.4 correlation is the main risk

error criteria count frequency, severity, reversibility and common cause. 10^{-4} per decision over a million decisions gives a hundred expected errors whether or not they are independent; correlation decides whether they arrive one by one or all at once. a shared wrong premise in every seat's training is the failure the operator and lineage grouping cannot see. the protection pool bounds some of the financial consequence; correction (section 8.14) bounds how long a wrong judgment spreads; neither makes it right.

19.5 learning from judgments

judgments are not truth labels. a seat trained on the contract's own past judgments learns their errors, so such training is tracked at the level of source, case and judgment, and a held-out case that enters training is retired from evaluation. a reversed judgment invalidates labels derived from it. case evidence is not used for training without the parties' permission.

19.6 composed failures

the reference tests (appendix J) exercise composed cases: three seats of one operator against two independent seats; four groups answering four labels, which has no winner; a divided or low-confidence top tier, which is unresolved; every seat failing, which leaves the question unavailable rather than guessed; an answer over a different package, which is an error; one image in a case, rendered by two renderers or marked unprocessed; an engine without a second top-tier panel, refused at admission; an rfc whose concurrence panel splits, which keeps the status quo; a proposed engine never consulted when the outgoing withholds; a timely appeal and a sweep a day late; a stale execution replayed against a fresh one; two equal pool claims on a short pool; a borrower who stops paying, through default, judgment and an exhausted pool. they exercise the rules over stipulated answers on the real database schema. they do not run a model, and they do not measure whether any model decides cases well — a correct pipeline around a wrong model produces a well-recorded wrong judgment.

19.7 the renderers are part of the evaluation

a rendering that drops a sentence changes the record every seat reads. renderers are therefore evaluated like seats: on held-out media with known content, by omission, insertion and transcription error, and by the agreement of two renderers, which measures correlation rather than correctness. evidence that only one renderer of a pair can read is unprocessed by rule, and its frequency is itself a measurement of what the engine cannot yet see.

19.8 matched baselines

compare like workflows against ordinary contracts and courts, simpler centralized software and the predecessor, matching value, jurisdiction, evidence complexity and funding. include abandoned, refused and failed cases, not only completed ones. count time, cost, privacy exposure, uncollected judgments and pool payouts along with throughput.

19.9 conclusion

the coordination contract is four things — kernel, law, engine, ledger — and one way of acting. it gives every person one voice, interprets what people actually say against what each option actually does, scores every option independently, and lets the engine block only on the kernel's three grounds and only concur when it is confident. it keeps politics out of the market and lets justice in: cases are decided element by element without juries, a tie or an outage is never a verdict, every top-tier decision keeps a fresh panel for its appeal, judgments reach bonds in one atomic step, and victims share a pool that pays them as it is funded. it services credit with the same rules. it removes every office and every key from which it could be steered outside a recorded procedure, including its founder's. what it does not remove is dependence on the quality of its engine's models, which no rule can supply and only evaluation can measure.

A records

the records below are the shapes of the reference module's types (packages/domain/core/src/contract), abbreviated. money is integer cents, time integer milliseconds since the epoch, confidence integer per mille. every hashed record is sha256(cc:⟨domain⟩:v4\n|| canon(record)).

A.1 state

```
ContractState = { kernelHash, version, law, engine, engineHash,
  lawHistory: [{ effectiveAt, version, law }],
  engineHistory: [{ effectiveAt, hash, name }], lastRfc }
stateHash = H("state", { kernel, version, law, engine: engineHash })
```

A.2 people and law

```
PeopleKind = { kind, uniqueness: identity_fingerprint | lineage
  | registration | custodial_individual, issuers[], representedBy? }
PersonhoodProof = { entityId, kind, issuer, uniquenessKey,
  issuedAt, expiresAt?, revoked }          voice = kind:uniquenessKey
  (statements apply in issuedAt order; a revocation is a tombstone)
Duty = { id, title, text, protects: interest[1..], harm,
  elements: [{ id, text }][1..], criminal, effectiveFrom }
Law = { version, people: PeopleKind[], duties: Duty[], parameters,
  evidenceModalities: (text | image | audio | video | file)[1..] }
LawChange = add_duty | amend_duty | repeal_duty | set_parameter
  | recognize_people | unrecognize_people | set_evidence_modalities
```

A.3 engine, evidence and determinations

```
Seat = { id, protocol, model, baseUrl?,
  pin: { weights, sha256, source } | { version, id },
  operator, lineage, controlledBy[], modalities[], costCents }
MlpHead = { id, task, labels[], embeddingSeat, embeddingDim,
  inputScale, layers: [{ in, out, weights: int8 b64, bias[], shift,
  relu }], calibration: [[margin, permille]], evaluation, sha256 }
Engine = { name, embedding: { seat, heads[] },
  classifiers: { seats[], quorum }, deliberative: { seats[], quorum },
  renderers: Seat[], bands: { 1, 2, 3 } }
InputPart = { text, label?, text }
  | { ref, label?, ref, mime, modality, sha256 }    (what people submit)
EvidenceText = { id (unique), label?, text }      (what seats read)
Rendering = { sha256, modality, by: [{ seat, text? | error? }],
  ok (>= 2 renderers produced text), at }
Question = { kind, prompt, labels[], inputs: EvidenceText[],
  structural?: { label, note } }
packageDigest = H("package", { kind, prompt, labels, inputs })
Determination = { id, kind, prompt, labels, packageDigest, engineHash,
  rounds: [{ tier, seats[], answers: [{ seat, label, confidence,
  reason?, error?, delivered }], aggregate: { label | null,
  confidence, groups, tie } | null, decisive, note }],
  outcome: { status: decided | unresolved | unavailable, label,
```

```
confidence, tier }, at }
```

A.4 rfcs

```
Rfc = { id, opener, title, body, target: law | engine,
  options: [status_quo, ...{ id, title, summary, changes | engine }],
  openedAt, closesAt, contestMs, contractVersion, engineHash,
  baseStateHash, people: kind[], depositCents }
Response = { id, rfcId, voice, entityId, parts: InputPart[1..20],
  digest, at, confirmedAt | null }
Extraction = { rfcId, voice, responseId, scores: { option: -2..2 },
  unscored: { option: unresolved | unprocessable },
  determinations[], availableAt, contestUntil, contested }
Tally = { respondents, scores, histogram: { option: [n-2..n+2] },
  unscored, favored, recordDigest }
Concurrence = { outgoing, incoming?, fallback? }
RfcDecision = { rfcId, adopted | null, tally, concurrence[], reason,
  depositReturned }      decisionHash = H("rfc-decision", decision)
```

A.5 cases

```
CaseFiling = { claimant, filedFor?, respondent, criminal,
  duty: { agreement, agreementId, term } | { law, dutyId }
    | { kernel, article },
  act: { description, at },
  harms: [{ id, interest, description, amountCents }][1..20],
  relief: Remedy[1..20], evidence: InputPart[], reopens? }
Remedy = { kind: restitution | compensation | performance
  | correction | penalty | exclusion, harmIds[1..], description,
  amountCents?, durationMs? }
Case = { id, filing, filedAt, family, contractVersion, engineHash,
  duty: { title, text, elements[] }, stage: screening | serving
  | awaiting_response | evidence | ready | decided | appealed | final
  | unprocessable | duplicate, servedAt, responseDue, evidenceDue,
  submissions: [{ party, parts, counter?, at }], instances[],
  appealableUntil, appealedBy, costCents,
  deposits: { claimant, respondent }, closedReason }
Instance = { n: 1 | 2, startedAt, startTier, panels: { 2?, 3? },
  findings: { key: { determination, status, label, confidence,
  tier } }, blockedSince, judgment }
Judgment = { outcome: liable | not_liable | dismissed,
  elements: { id: boolean }, awards: { harm: { cents, figure } },
  remedies[], determinations[], reason, at }
ExecutionPlan = { dueCents, victimDueCents, penaltyCents, costCents,
  takes: [{ holding, amountCents }], collectedCents, shortfallCents,
  victimShortfallCents, costPayer: claimant | respondent | commons,
  refunds: { claimant, respondent }, depositsToCosts }
PoolClaim = { caseId, victim, respondent, shortfallCents, paidCents }
PoolPayout = { caseId, victim, paidCents, residualCents, closed }
```

A.6 credit

```
Mandate = { investor, capitalCents, minFacilityCents,
```

```
    maxFacilityCents, rateBps, termMs, installments,  
    borrowers: anyone | persons, openUntil }  
Facility = { id, mandateId, borrower, principalCents, rateBps,  
  schedule: [{ n, due, principalCents, interestCents, paidCents }],  
  premiumCents, disbursedCents, holders: { holder: units },  
  state: active | repaid | defaulted, drawnAt, defaultCases }
```

B transitions

B.1 an rfc

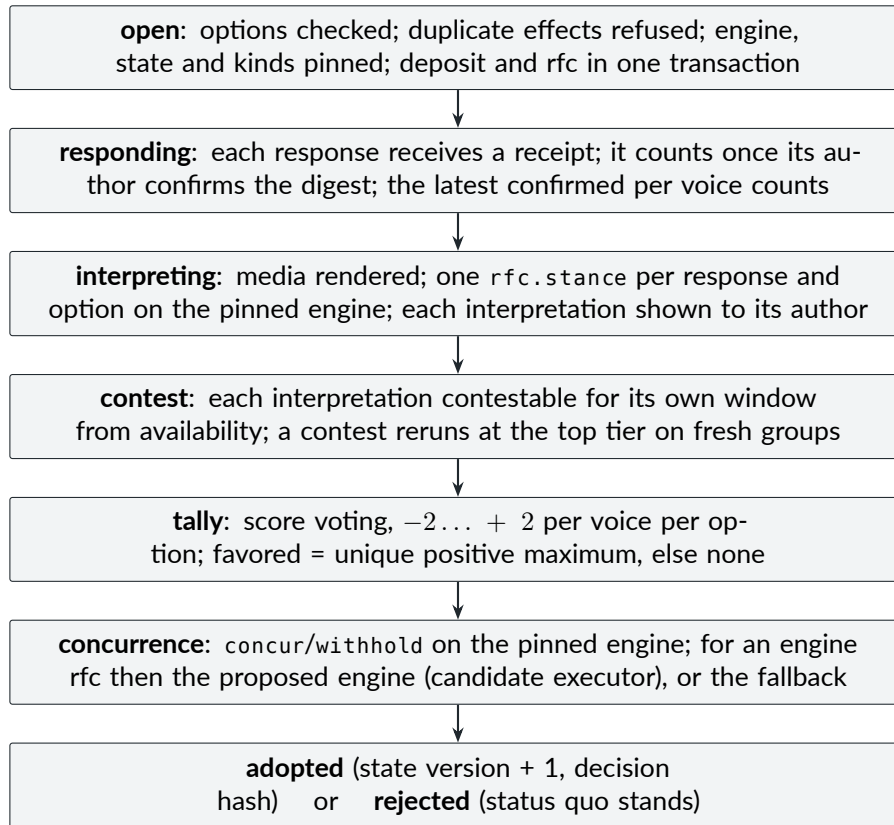


figure B.1: the life of an rfc. a draft refused at opening never enters it.

terminal rules: no favored option; a withheld, unresolved or unavailable concurrence (an outgoing engine unavailable for a further window, on an engine rfc, gives way to the proposed engine's concurrence); an option that no longer applies; or a contract state that moved since opening (superseded) — all end in *rejected*. the decision, the deposit settlement and the new version commit in one transaction claimed on the rfc's revision and the contract's version. an rfc never ends undecided.

B.2 a case

stage	leaves when	to
(filing)	any structural refusal	refused, with codes
screening	same matter decided, or a matter key equal to an earlier case's	duplicate
screening	otherwise (an unresolved screening proceeds)	served:
awaiting_response	respondent answers, or the response deadline passes	awaiting_response
evidence	the evidence deadline passes	evidence
ready	instance 1's questions all answered	ready
ready	instance 1 cannot be seated or completed for the law's period	decided
		unprocessable

stage	leaves when	to
decided	a party appeals within the window (recorded at once)	appealed
decided	the window passes with no appeal	final
appealed	instance 2 (fresh top-tier panel) answered; waits while no fresh panel can sit	final
final, unprocessable, duplicate	the execution batch commits	executed; victim shortfall an open pool claim

table B.2: case stages in the reference module (caseStageAt, served, pendingQuestions, recordJudgment, recordAppeal, closeUnprocessable, closeDuplicate).

terminal rules: an element not decided as established — unresolved included — means not liable; a criminal element below the top tier or below the criminal confidence is not established; a question that cannot be answered is retried and never decided against anyone; an appeal recorded in time is always heard; a respondent with too little to pay leaves a recorded shortfall and an open pool claim. a case never ends on a tie or an outage, and never escalates to a person.

B.3 a determination

for tiers $t = s, \dots, 3$ from the start tier s : skip a tier that cannot answer (no structural answer, no matching head, no quorate panel); seat the fixed panel, or draw one of exactly the tier's quorum of eligible groups; record every seat's answer or error, refusing as an error any answer over a different package; stop at the first tier whose aggregate has a unique winner at its band. at the top tier, a tie or a confidence below the band is *unresolved*. if no tier answered, the outcome is *unavailable*. the record's identity is the hash of its content.

B.4 a facility

active $\rightarrow$ repaid when the schedule is paid in full; active $\rightarrow$ defaulted when an installment stays unpaid past the grace period, at which point each holder's breach case is filed for its share of the accelerated balance. every servicing, transfer and default commits as one transaction claimed on the facility's revision.

C genesis law and engine

generated from GENESIS_LAW and GENESIS_ENGINE in the reference module. the law hash prefix is b42faffe8b32e182, the engine hash prefix ad3328bd671cac5d and the genesis state hash prefix 9857dd7c9bba8915.

C.1 people

the genesis law recognizes the kind human under the rule identity_fingerprint, and admits as evidence and responses: text, image, audio.

C.2 duties

id	duty	protects	elements	crim.
fraud	no fraud. No one may obtain property or an agreement from another by a representation of fact they know to be false, or make with reckless disregard for its truth, that the other relied on.	property, agreement	representation, falsity, knowledge, reliance, harm	yes
theft	no theft. No one may take or use another's holdings or property without the owner's consent.	property	taking, ownership, no consent, harm	yes
violence	no violence. No one may use the contract to arrange, pay for or carry out physical harm to a person.	life, bodily integrity	act, harm	yes
coercion	no coercion. No one may compel another's labor, agreement or response to an RFC by threat of harm.	liberty, agreement, voice	threat, extraction, causation	yes
breach	keep agreements. Whoever accepted an agreement must perform its terms or answer for the loss its breach causes.	agreement	agreement, non-performance, harm	no
privacy	no exposure. No one may publish another person's private information obtained through the contract without consent.	privacy	private, source, publication, no consent	no

C.3 parameters

parameter	genesis value
law rfc window	14 days
engine rfc window	28 days
extraction contest window	3 days
rfc deposit	\$10
responses sampled in full for concurrence	24
cooldown before a rejected rfc's options can reopen	90 days
case response time	7 days
case evidence time	7 days
case cost deposit (as far as the balance allows)	\$5
reopening window after a final judgment	180 days
unprocessable after	14 days
open cases per claimant	20
appeal window	7 days
claims starting at the top tier	from \$1,000
interim freeze	14 days
criminal confidence	950/1000
penalty multiple	2×
maximum exclusion	180 days
pool premium on bond issuance	50 bps
pool cap per judgment	\$10,000
commons fee on settled transfers	10 bps
credit servicing fee	25 bps
credit grace period	7 days

table C.2: genesis parameters.

C.4 engine

bands: tier 1 950, tier 2 850, tier 3 700 per mille. quorum: 2 of 3 independent groups at tier 2, 3 of 6 at tier 3 (2 disjoint panels). renderers (tier r below): image — render-gpt-5.6-luna, render-claude-haiku-4-5; audio — render-whisper-large-v3-turbo, render-gpt-4o-mini-transcribe. no mlp heads. no seat records a controlling entity yet.

tier	seat	operator	lineage	pin
1	bge-base-en-v1.5	cloudflare	bge	weights 0dbc3b2eeffc hf:BAAI/bge-base-en-v1.5@a5beble3 e68b9ab74eb54cfd186867f64f240e1a
2	jev-1.13.0	typesafe	jev	version jev-1.13.0
2	llama-3.1-8b	cloudflare	llama	weights b95e1c842c5c hf:meta-llama/Llama-3.1-8B-Instr uct@0e9e39f249a16976918f6564b883 0bc894c89659
2	gpt-5.6-luna	openai	gpt	version gpt-5.6-luna
3	claude-opus-5-5	anthropic	claude	version claude-opus-5-5
3	gpt-5.6-sol	openai	gpt	version gpt-5.6-sol
3	deepseek-v4-pro	deepinfra	deepseek	weights 249bebe0762f hf:deepseek-ai/DeepSeek-V4-Pro@b 5968e9190ef611bbf34a7229255be88a 0e937c1
3	llama-3.3-70b	cloudflare	llama	weights 68d90187d3d2 hf:meta-llama/Llama-3.3-70B-Instr uct@6f6073b423013f6a7d4d9f391449 61bfbbfbc386b

tier	seat	operator	lineage	pin
3	qwen3-max	alibaba	qwen	version qwen3-max
3	kimi-k2.6	openrouter	kimi	weights 03dd534e3ddf hf:moonshotai/Kimi-K2.6
r	render-gpt-5.6-luna	openai	gpt	version gpt-5.6-luna
r	render-claude-haiku-4-5	anthropic	claude	version claude-haiku-4-5-20251001
r	render-whisper-large-v3-turbo	cloudflare	whisper	version @cf/openai/whisper-large-v3-turbo
r	render-gpt-4o-mini-transcribe	openai	gpt-transcribe	version gpt-4o-mini-transcribe

open-weight pins are sha256 values over the model's published file manifest at one commit of its source repository; hosted pins are the exact model id the provider serves. both are recorded here as the reference module states them. no seat's accuracy on contract questions has been measured (chapter 19).

D worked cases

every number below is computed by the reference module (`generate_fixtures.mjs`); determinations a case depends on are stipulated where a real engine would answer, and say so. people and amounts are synthetic.

D.1 case-01: an rfc with two rival options

an rfc proposes lowering the commons fee from 10 to 5 basis points (`fee5`) or abolishing it (`fee0`). its window is 14 days, and each interpretation is contestable for 3 days from when it is shown. an attempt to add a third option that also sets the fee to 5, under another title, is refused (`rfc.duplicate_effect`). seven people respond and confirm their receipts; their interpreted scores are:

person	fee5	fee0
ana	2	2
ben	1	-1
cy	2	0
dee	-1	-2
eli	0	2
fay	1	0
gus	-2	-2

table D.1: interpreted scores, -2 to $+2$, per voice and option.

the scores are $S_{\text{fee5}} = 3$ and $S_{\text{fee0}} = -1$ (the counts of voices scoring $-2, -1, 0, +1, +2$ are 1, 1, 1, 2, 2 and 2, 1, 2, 0, 2). the favored option is `fee5`. had a third option — two basis points — been on the ballot, with everyone but gus opposing it strongly, its score would be -10 and the first two options' scores would stay 3 and -1 : under score voting an option's score does not depend on what else is offered.

an eighth response is interpreted 9 days after close, because the interpreter was slow; its author can contest it until day 12 after close, and its `fee0` interpretation was unresolved (`fee0: unresolved`), so it scores 0 there — neither support nor opposition.

with the engine's concurrence decided (stipulated), the rfc adopts `fee5`; the state moves to version 1 with a commons fee of 5 bps, and the decision record's hash begins `6df1c1a4a22f0609`. had the engine withheld, the result would be null; had its panel been divided, null — “the pinned engine was divided or uncertain: no concurrence (`rfc.concur:null:unresolved:3:900`)”. applying the same decision again to the moved state is refused: `adopt.superseded`.

D.2 case-02: replacing the engine

an engine rfc raises the classifier band. its window is 28 days. one person supports it. with only the outgoing engine's concurrence recorded the result is null: the proposed engine has not been asked. if the outgoing engine withholds, the result is null whatever the proposed engine would say — it is never asked. if the outgoing concurs and the proposed engine's panel is divided, null. with both concurring, `band900` is adopted. if the outgoing engine could not seat a panel at all for a further window as long as the rfc's own, the proposed engine's concurrence alone adopts `band900`.

D.3 case-03: three seats of one operator

five seats answer a yes/no question, each at 900. three belong to one operator and answer yes; two independent seats answer no. the five seats form 3 groups; the tier decides no at 600: the operator's three seats weigh as one.

D.4 case-04: a climb

a service agreement asks whether the delivered work fulfills the brief (`holds` / `does_not_hold`), over a two-item package whose digest begins `b2d0aa677ff3b6e3`. the genesis engine's classifier panel, drawn by seed from its three groups, answers llama-3.1-8b `does_not_hold` 900; gpt-5.6-luna `does_not_hold` 500 (stipulated): `does_not_hold` at 700 over 2 groups, below band 850, so the question climbs. the deliberative panel of 3 groups answers gpt-5.6-sol `holds` 800; llama-3.3-70b `holds` 900; claude-opus-5-5 `holds` 900: `holds` at 867, above the band of 700. the outcome is decided: `holds` at tier 3. the determination's identity begins `019e78ce672f200e`.

D.5 case-05: no winner by order

four independent groups answer a concurrence question, each at 900, with four different labels — `concur`, `kernel_violation`, `voice_mismatch`, `outside_law`. the aggregate over 4 groups has no winning label (tie: yes); version 0.4 would have answered `concur`, the first label, at 250. (the concurrence question now has only `concur` and `withhold`, so three objections cannot split against one assent in the first place.)

the service-agreement question of appendix D.4, started at the top tier, meets a panel that answers gpt-5.6-sol `does_not_hold` 900; llama-3.3-70b `does_not_hold` 900; claude-opus-5-5 `holds` 500. `does_not_hold` has the larger weight, but its confidence is 600: below band 700: unresolved. the outcome is unresolved, with no label; the agreement's own rule says what that means for the escrow.

D.6 case-06: an mlp head

a two-layer head with labels `no/yes`, input scale 100 and identity-like weights receives the embedding (0.12, -0.31). quantization gives (12, -31); the hidden relu layer gives (12, 0); the output layer gives logits (-12, 12). the label is `yes`, the margin 24, and the calibration table's largest threshold below the margin assigns 950. any machine computes the same numbers. the head's hash begins `e3bd0562c93234ae`.

D.7 case-07: a panel and its appeal

a case between *buyer* and *seller* starts its first instance on the genesis engine. its top-tier panel, drawn by seed from the six independent deliberative groups, is kimi-k2.6, qwen3-max, llama-3.3-70b. an appeal draws again, excluding every group that sat: deepseek-v4-pro, claude-opus-5-5, gpt-5.6-sol. no group sits twice. a third disjoint panel does not exist (quorate: no) — the engine admits exactly as much appellate capacity as the kernel requires, which is why a case has one appeal.

D.8 case-08: an image in the record

a claimant submits a photograph of a receipt. the contract stores it by its sha256 and asks both image renderers: render-gpt-5.6-luna, render-claude-haiku-4-5. both produce text (ok: yes), and the record carries 2 text items in the photograph's place, one per rendering. had one renderer failed, the item would not be before the seats (ok: no): the record would mark `claimant/1/1` as unprocessed and say that it counts against no one. either way every deciding seat reads the same text, and no tier loses its quorum for want of a vision model.

D.9 case-09: a criminal standard, element by element

a buyer alleges fraud: a seller represented 120 independent paying customers, of whom 47 qualified, and the buyer paid an \$800 deposit in reliance. the claim names one harm item and seeks restitution, a penalty and 90 days' exclusion, each answering that item. as a criminal claim its instance starts at tier 3, and each of the duty's elements — representation, falsity, knowledge, reliance, harm — is decided separately at the top tier. if the panel establishes every element at 930, the respondent is *not liable*: below

the 950 standard. elements established at tier 2, even at 990, also leave the respondent *not liable*. at 970 on every element, with the claimant's figure chosen and every remedy justified, the respondent is liable: restitution \$800, a penalty capped at $2\times$ the total award, \$1,600 (\$800 to the victim, \$800 to the pool), and 90 days' exclusion. a filing that asked for restitution and compensation of the full \$800 each would have been refused at the door: k8.double_recovery. every determination is stipulated.

D.10 case-10: a civil judgment reaching a bond

a seller took \$800 for an order and, the buyer says, delivered nothing. the claimant deposits \$5. the case starts at tier 1. the seller answers that half was delivered and states its figure, \$500. every element is established (stipulated). the panel finds the claimant's figure closer to the proved loss, so the award is \$800; had the panel been unresolved, the award would have been the respondent's figure, \$500.

the engine's cost of the case is \$12, paid by the respondent. the seller holds \$150 in cash and participations in a bond worth \$10,000. execution takes \$150 in cash and \$662 of the bond — \$812 in total, and nothing more; the shortfall is \$0, and the claimant's \$5 deposit comes back. had the seller held only the cash, \$150 would be collected and the victim's \$650 shortfall would become an open pool claim.

had the buyer lost, the claimant would pay: \$5 of the deposit goes to costs, \$0 comes back. had the engine been unable to decide for the law's period, the commons would bear the cost and the deposit, \$5, would come back whole. and had the seller appealed in time, the case would still be appealed a year later if no fresh panel could sit: no clock finalizes an appealed case.

D.11 case-11: two pool epochs

issuing \$100,000 of bonds pays a \$500 premium into the pool. later the pool holds \$10,000, with a cap of \$10,000 per claim, and three judgments have left victims short. the first epoch pays every capped need the same fraction; a later epoch, after \$14,000 more premiums and penalties, pays what is still owed:

claim	shortfall	epoch 1	epoch 2	owed by respondent	closed
case-a	\$4,000	\$2222.22	\$1777.78	\$0	yes
case-b	\$15,000	\$5555.55	\$4444.45	\$5,000	yes
case-c	\$4,000	\$2222.23	\$1777.77	\$0	yes

table D.2: equal claims (case-a, case-c) are paid equally in each epoch, to the cent; the larger claim is capped at \$10,000 over both epochs, and the rest stays owed by its respondent. after epoch 1 the pool holds \$0; after epoch 2, \$6,000. the order in which the judgments executed changes nothing.

D.12 case-12: the \$10,000 credit cycle

an investor, ivy, commits a mandate of \$10,000 at 12% a year over 12 monthly installments, to persons only; an applicant without a personhood proof is refused (borrower_not_a_person). a person, bo, draws the whole of it to buy a cnc router. the issuance premium of \$50 goes to the pool and bo receives \$9,950. ivy holds the bond.

the first installment, due after 30 days, is \$833.33 of principal and \$98.63 of interest. before it is paid, ivy sells a quarter of the bond to pat. the contract collects the installment from bo's balance, takes \$2.32 for the commons, and distributes \$697.23 to ivy and \$232.41 to pat. bo then stops paying; installment 2 passes its grace period. the outstanding \$9709.08 is accelerated, and two breach cases are filed — ivy's for \$7281.81 and pat's for \$2427.27 — each passing the filing checks (refusals: 0). from there they are ordinary cases (appendix D.10); in the reference tests they reach judgment against an empty respondent, and the two holders' pool claims are paid in equal proportion as far as the pool goes.

D.13 case-13: an anonymous voice

three people register semaphore commitments for their human voices: ann (commitment 101) and bea (201) before an rfc opens, cal (301) after it. ann rotates her identity after opening, to 102. the rfc pins the human tree as it stood at opening: 2 leaves, 101, 201; a day after cal registered the tree is 201, 102, 301. the pinned root is part of the rfc's identity (an rfc opened over another root is another rfc: yes).

ann is in the pinned tree, so she answers this rfc only anonymously (yes); her rotated identity is not in the pinned tree and gives her no second voice. cal is not (no), so he answers with his signed-in voice. one anonymous member answers "yes", then "on reflection, no"; another answers "yes". both of the first member's proofs carry nullifier 77, so they are one voice, anon:human:77, and the later replaces the earlier: 2 responses count, the first member's being "on reflection, no".

a contest proof whose root, scope and message are right is accepted (null); the same proof against another root is refused `not_the_pinned_tree`, in another rfc's scope `wrong_scope`, over the lookup message `wrong_message`, and claimed for a kind not recognized at opening `kind_not_recognized_at_open`. an anonymous response citing an image by url is refused (`anonymous_media_must_be_uploaded`): anonymous media must be uploaded under a proof.

D.14 case-14: an attested answer and a rewritten history

a seat names an attestor. its answer — yes at 950 over a package — carries the attestor's signature (beginning 635a94222687a2a7) over the `cc:attestation:v4` message, and counts (null). the same signature does not survive a changed label (`wrong_answer`), a different package (`wrong_subject`), a different attestor key in the engine (`wrong_key`), a missing attestation (`missing`) or a flipped signature byte (`bad_signature`).

the contract's log records four acts; its head begins b7c7d9101ba8a765 and the chain verifies (yes). an operator who rewrites the second entry's body digest leaves 1 problem the replica reports; one who drops an entry leaves 2. a checkpoint of the fourth entry's head matches the log (yes); a checkpoint that claims the third entry's head for the fourth position is a mismatch (1).

E mandate-constrained allocation mathematics

let i index capital mandates and j candidate claims. $x_{ij} \geq 0$ is the amount of mandate i allocated to claim j ; K_i is available capital; D_j is the maximum approved financing demand; and $e_{ij} \in \{0, 1\}$ denotes verified eligibility under hard contractual constraints. a zero means ineligible or not sufficiently established, not necessarily socially undesirable.

$$\sum_j x_{ij} \leq K_i, \quad \sum_i x_{ij} \leq D_j, \quad x_{ij} \leq e_{ij} K_i. \quad (\text{E.1})$$

sector, obligor, related-party, geography, maturity, and settlement-asset limits add constraints such as $\sum_{j \in S} x_{ij} \leq c_{iS} K_i$. aggregation must follow common beneficial ownership and shared risk rather than treating several legal wrappers as independent borrowers. liquidity eligibility concerns realizable cash and redemption terms, not just a stated maturity.

E.1 objectives do not override mandates

let a_{ij} be a documented estimate of mandate alignment, not a measurement of moral worth; $\hat{r}_j$ an estimated net return after stated costs; and $\mathcal{R}_i(x)$ a risk measure. one possible objective is

$$\max_x \sum_{i,j} x_{ij} (\alpha_i a_{ij} + \beta_i \hat{r}_j) - \sum_i \gamma_i \mathcal{R}_i(x), \quad x \in \mathcal{F}, \quad (\text{E.2})$$

where $\mathcal{F}$ is the feasible set defined by approved constraints. coefficients express investor-approved trade-offs and must not be hidden in model defaults. multiple objectives may instead be displayed as a frontier from which the authorized investor chooses; a single combined score is not required.

if a mandate asks for a minimum *estimated* portfolio return, it could add

$$\sum_j x_{ij} \hat{r}_j \geq r_i^{\text{target}} \sum_j x_{ij}. \quad (\text{E.3})$$

this is a constraint on an estimate, not a promise about realized returns. the all-zero allocation is feasible unless minimum deployment is explicitly required. when a minimum deployment requirement makes $\mathcal{F}$ empty, the output is an infeasibility explanation and a request for a new mandate, not a concealed relaxation.

E.2 losses and model uncertainty

let $L_i^{(s)}(x)$ be net portfolio loss under scenario s with explicitly justified probability p_s . a possible tail-risk constraint uses conditional value at risk at level η :

$$\text{CVaR}_\eta(L_i) = \min_z \left[z + \frac{1}{1-\eta} \sum_s p_s \max(L_i^{(s)} - z, 0) \right] \leq \ell_i. \quad (\text{E.4})$$

this finite-scenario definition assumes a specified loss distribution. neither the blockchain nor the formula validates the scenarios, their probabilities, the chosen horizon, or recovery assumptions. stress tests should also include severe unweighted scenarios and shared-service failures omitted from historical repayment data.

E.3 verifiable execution, contestable inference

the allocation artifact contains mandate and borrower versions, eligibility evidence, selected amounts, scenario set, model and solver versions, conflicts, costs, concentration checks, and an explanation of rejected or unfunded alternatives. a deterministic checker recomputes the hard arithmetic and permissions before release. it does not certify that projected revenues will occur or that the solution is globally optimal unless an appropriate optimality certificate exists.

semantic retrieval remains upstream: it finds candidates whose descriptions may match the intent. financial feasibility, verified use of proceeds, economic additionality, and causal mission impact require different evidence. fund managers may publish aggregate performance by model vintage while protecting individual borrowers' confidential information. retraining a model does not rewrite the model used to justify an earlier loan.

F cash flows and loss allocation

amounts are hypothetical dollars computed in integer cents by `generate_fixtures.mjs`. taxes, currency risk, valuation changes and unstated costs are excluded, not insured.

F.1 a direct loan participation

a hypothetical investor contributes \$10,000 to a restricted pool. the pool advances \$6,000 to borrower a and \$4,000 to borrower b, each for one year at 8% simple interest. the accepted servicing charge is \$200.

cash-flow item	a	b	total
principal	6,000	4,000	10,000
interest if both pay	480	320	800
full collections	6,480	4,320	10,800
servicing charge	—	—	(200)
net investor distribution	—	—	10,600

table F.1: a fully performing participation: a net gain of \$600, or 6%.

if a pays in full while b pays no interest and recovers 40% of principal, collections are \$6,480 plus \$1,600, or \$8,080. after the same charge the distribution is \$7,880, a loss of \$2,120 (21.2%). b's default is not by itself a case: financial distress is not fraud. if b's default breached an accepted term, the pool may sue, and a judgment reaches b's holdings like any other (section 8.11).

F.2 issuer obligations and first loss

if an issuer instead promises \$10,500 at maturity irrespective of borrower performance, full collections after servicing leave \$100; the default scenario leaves a \$2,620 shortfall, which is the issuer's liability, not a pass-through.

a separate \$100,000 principal-only portfolio has \$10,000 junior funding and \$90,000 senior funding. for loss L , junior keeps $\max(J - L, 0)$ and senior keeps $S - \max(L - J, 0)$.

principal loss	collections	junior remaining	senior remaining
6,000	94,000	4,000	90,000
16,000	84,000	0	84,000

table F.2: loss allocation changes who bears loss, not the amount of capital actually supplied.

the protection pool is not first-loss capital for investment portfolios. it pays victims of uncollected judgments (section 10.3); a borrower's default that breaches nothing creates no claim on it.

G economic measurements and bulletin schema

measurements require a denominator, a boundary, a time convention, and a correction policy. a ledger event is evidence for a specified accounting event, not permission to classify every movement of money as production.

indicator	operational definition	limitation / required disclosure
settled payments	sum of qualifying transfers over a declared interval, grouped by asset and economic purpose	exclude internal transfers where required; volume is not value added
verified commercial receipts	receipts linked to qualifying customer transactions, net of specified refunds	disclose verification standard, related parties, and off-protocol coverage
covered value added	qualifying output less qualifying intermediate consumption over a fixed boundary	requires economic classification and adjustments; not automatically national gdp
active enterprises	entities meeting a published operating-activity criterion	registration is not operation; disclose inactivity and duplicate-entity treatment
new enterprise formation	newly registered entities and, separately, newly active enterprises	no implication that every registered shell is a new business
credit outstanding	principal receivable at the observation time, with currency and impairment basis	include assigned claims once; disclose unobserved liabilities
new credit issuance	new principal advanced during the interval	exclude principal refinancing or show it separately
delinquency	overdue principal / eligible outstanding principal under a named day-past-due definition	specify grace periods, rescheduling, and cohort age
realized credit loss	unrecovered principal and specified costs for a mature cohort	incomplete recoveries and immature loans require separate estimates
public fiscal balance	recurring recognized revenue less specified spending, under a stated accounting basis	do not classify borrowing or returned loan principal as revenue
liquidity coverage	eligible available cash relative to obligations due in a stated horizon	pledged reserves and conditional assets are not free cash
market yield	yield implied by a qualifying executable or completed transaction and its cash-flow terms	distinguish bids, offers, last trades, indicative models, and stale markets
price index	weighted changes for a specified basket and quality-adjustment method	composition shifts, sparse prices, and changing quality can dominate
rfc participation	persons notified, responding, and contesting their extraction, by kind of people, using separate counts	silence is not support or opposition; protect small cohorts from identification
pool coverage	pool balance against open and expected victim claims	premiums are not reserves for other purposes; disclose correlated exposures
coordination burden	distribution of elapsed time, fees, labor, disputes, and interventions per completed workflow	include failures and excluded applicants; faster is not always better

G.1 a machine-readable bulletin item

```
indicator_id: commercial_receipts.change_7d
status: illustrative_schema
value: numeric_or_unavailable
unit: percent_change
population: explicit_entity_and_transaction_boundary
window_start: canonical_timestamp
window_end: canonical_timestamp
observation_cutoff: canonical_timestamp
baseline_window: explicit_comparable_period
coverage: observed_share_or_unknown
method_version: content_digest
source_commitments: referenced_input_records
uncertainty: estimation_range_or_not_applicable
missingness: known_gaps_and_selection_limits
revision_vintage: monotonic_version
stale_after: canonical_timestamp_or_trigger
narrative: generated_only_from_this_record_and_cited_context
```

an “opening high” is not a generic economic statistic: an opening price, interval high, percentage return, annual growth rate, and annualized growth rate refer to different computations. the narrator must not borrow market vocabulary merely to sound financially sophisticated. when there are too few trades for a yield curve, the output is “insufficient market observations,” not an interpolated chart presented as observed borrowing costs.

publishing a correction produces a new vintage linked to the old one. historical analyses specify which vintage was available at the decision time. privacy suppression and sampling can make a figure less precise; the narrative must preserve that limitation instead of rounding it into false certainty.

H threats and controls

the contract itself, an engine provider, its founder, its operator and its majority can each be the adversary. each threat names the control and what remains. the table is generated from `threats.json`, which also names the rules of appendix I that implement each control; section 18.2 sets each threat against the status of those rules.

id	threat	control and residual	where
th-01	one provider runs several seats to outvote the rest	operator and lineage grouping; residual: hidden common control or undisclosed shared training	section 7.5
th-02	a provider changes or retires a model behind its id	open-weight seat required per tier; unavailable rather than guessed; residual: serving-stack drift of open weights	section 7.3
th-03	an incumbent engine entrenches itself, or a captured engine installs itself	dual-engine concurrence; voice required; residual: both engines sharing one failure	section 6.7
th-04	the engine rules instead of interpreting	concurrence is concur or withhold, on three grounds only; it can block, never adopt, and concurs only when decided; residual: an engine that blocks in bad faith, answered by an engine rfc	section 6.6
th-05	interpretation reads support into responses that express none	per-response determinations on the record, each shown to its author with its own contest window; contest on fresh groups; a seeded sample read by concurrence; residual: respondents who never check	section 6.4
th-06	one person, many voices	per-kind uniqueness rules; earliest proof holds the voice; residual: issuer error and identity theft	section 5.1
th-07	intensity inflation, or agenda games with clone options	score voting bounded to ± 2 per option; an option's score independent of the others; duplicate effects refused; residual: near-duplicates that differ in effect	section 6.5
th-08	a majority uses law against a minority's holdings	k5: no freezing, seizing or singling out; structural check plus outside law; residual: a harm-based duty drafted to burden one group in practice	section 5.4
th-09	courts used to harass	objective scope checked before any model is asked; loser pays; residual: a well-funded litigant willing to lose	section 8.1
th-10	a respondent escapes by disrupting appeals, or an appeal is lost to a late clock	the appeal is a stage no clock finalizes; two disjoint top-tier panels admitted; an appeal waits for a fresh panel and the judgment is not executed meanwhile; residual: a long outage delays both sides	section 8.9
th-11	a respondent hides assets in bonds	judgments reach bonds; interim freeze; residual: assets outside the contract	section 8.11
th-12	a victim left unpaid, or paid after others by arrival	pool allocation epochs over every open claim, persistent residuals, subrogation; residual: a pool exhausted by correlated losses	section 10.3
th-13	evidence that instructs the model	closed labels; effects only through fixed functions; residual: an injected wrong label	chapter 9

id	threat	control and residual	where
th-14	a key above the contract, or a code change that alters the kernel's effect	registry holds upgrade authority; code upgrades need both engines and a seven-day timelock; genesis manifest pinned; residual: compromise of a seat's signing service	section 15.4
th-15	the founder keeps a back door	genesis by rfc; no founder power after it; residual: undisclosed infrastructure access, recorded as a power	section 18.7
th-16	a rejected rfc is reopened until people tire	cooldown on options with identical effects; residual: variations that differ in effect	section 6.1
th-17	a divided engine read as a decision	unique winner at the band, or unresolved; each use states what unresolved means; residual: a matter the engine leaves unresolved costs whoever bears its burden; a hidden correlation between groups	section 7.1
th-18	a proposed engine harvests credentials or private evidence	credentials bound to operator hosts; candidate executor renders nothing; asked only after the outgoing concurs; residual: the proposed engine's providers see the public rfc record they are asked about	section 7.9
th-19	seats decide on different evidence; media breaks quorum	one canonical text package, delivered digests, two renderers per medium, unprocessed items marked; residual: a delivered digest is the adapter's report, not the provider's, until the seat is attested	section 9.3
th-20	money moved twice, or half-moved	op-claimed conditional batches with deterministic ids; revision-guarded writes; residual: movements outside the contract's database (external payment rails) need their own reconciliation	section 15.5
th-21	a remedy list that multiplies recovery	harm items; per-item money bounds; one penalty cap; remedy justification; residual: an overstated harm item, answered only by the respondent's own figure	section 8.8
th-22	the same matter litigated until the respondent gives up	screening; matter families; one reopening with new evidence; loser pays; residual: the same matter recast as a different act, if the engine is persuaded	section 8.3
th-23	a revoked proof replayed	monotonic statements; revocation tombstones; residual: an issuer that signs a newer issuance by mistake	section 5.1
th-24	stuck matters starve the scheduler	due-work queue, earliest due first; residual: bounded work per sweep delays every matter when the queue is long	section 15.5
th-25	an agent commits its owner to debt	credit commitments need a person's own credentials; residual: a person's stolen credentials	section 10.4
th-26	the platform learns which rfc response is whose	anonymous voice: a zero-knowledge proof of membership in the tree the rfc pinned, sent without any credential; nothing stored links a nullifier to a person; residual: a small anonymity set, network metadata, timing, the content itself, and the proving-artifact host seeing the person's address	section 6.3.3

id	threat	control and residual	where
th-27	one person counts twice: once signed in and once anonymously, or through a rotated identity	a person in an rfc's pinned tree answers it only anonymously; a commitment registered or rotated after opening is not in the pinned tree; residual: a person who loses their identity file cannot answer rfcs opened while it was registered	section 6.3.1
th-28	a forged or replayed proof speaks for someone	groth16 verification against semaphore's own keys inside the worker; root, scope and message checked per action; a response's message binds its digest and a fresh nonce; residual: a flaw in the circuit or its trusted setup	section 6.3.2
th-29	the operator of the sweep fabricates or alters a seat's answer	an attested seat's answer counts only with its attester's signature over that exact answer; the replica recomputes every published determination; residual: seats without an independent attester — every genesis seat today — rest on the operator's word	section 16.1
th-30	the operator rewrites, reorders or forks the contract's history	a public hash-chained log of every act, its heads anchored on the registry's checkpoint instruction and checked by the replica; residual: acts the operator never logs, until reconciliation or a party notices; checkpoints not anchored until the registry is deployed	section 16.2
th-31	authority over a machine or service issued outside the engine	the command notary authorizes, delegates and revokes only on a top-tier quorum of the registry's seats, bound to the registry's version and engine; residual: the programs are not deployed; an executor that ignores the notary	section 17.2
th-32	a copy of the contract on another chain diverges from the registry	the evm mirror applies only the registry's engine-signed adoptions with the same seat keys, group quorums, dual-engine rule and timelock; residual: a relay that stops relaying leaves the mirror behind (never ahead); the mirror is not deployed	section 17.3

I implementation status

`guarantee_status.json` lists each consequential rule of this paper with where it is specified, where it is implemented, which test exercises it, and its status. it is validated against `records/status.schema.json`. *implemented* means present in this repository (the core module, the api worker, a script or a program); *tested* means a named test in a named file exercises it (appendix J); *neither* means deployed, run against live models or independently reviewed. *specified* means stated in this paper only.

id	rule (section)	status	limits
s-01	the kernel is fixed and identified by its hash (4.2)	tested	a different kernel is a different contract; nothing stops someone from deploying one
s-02	the contract state is created at genesis and published (18.1)	tested	genesis is written by the service, not adopted by a genesis rfc
s-03	one person, one voice; per-kind uniqueness; one human key across documents and issuers (5.1)	tested	issuer error and identity theft; two people sharing name and birthday share a voice
s-04	signed statements apply in issue order; revocations are tombstones (5.1)	tested	no issuer for a kind other than human exists
s-05	law changes are checked against kernel floors, human recognition, duty elements and rendering coverage (5.3)	tested	the semantic half of k5 rests on the engine
s-06	market freedom: law may not set prices, single out holders or restrict consenting exchange (5.4)	implemented	structural check plus a concurrence ground; no test of the engine applying it
s-07	every model tier has two operators and an open-weight seat (7.4)	tested	serving-stack drift of open weights
s-08	the top tier holds two disjoint panels: every top-tier decision is appealable (7.4)	tested	capacity is counted in groups that are available; an outage can still delay an appeal
s-09	every admitted medium has two independent renderers (7.4)	tested	both genesis audio renderers descend from openai research
s-10	seats of one operator or lineage count once; a tie has no winner (7.5)	tested	hidden common control or shared training
s-11	panels of exactly a quorum, drawn by seed, fixed per instance; parties and controllers never sit (7.6)	tested	controlledBy is empty in the genesis engine
s-12	outcomes are decided, unresolved or unavailable; nothing is guessed (7.1)	tested	stipulated answers; no live model
s-13	one canonical text package with unique ids; answers over a different package are errors (7.9)	tested	the delivered digest is reported by the adapter, not by the provider
s-14	credentials are bound to operator hosts; a proposed engine renders nothing and is asked only after the outgoing concurs (7.9)	tested	a legitimate provider host still receives the question it is asked
s-15	media is verified against its digest and rendered by two renderers; unrenderable items are marked (9.3)	tested	a rendering can omit or misread; renderers are not evaluated
s-16	mlp heads are integer-only and reproducible (7.7)	tested	the embedding itself is floating point
s-17	heads are trained, quantized and calibrated conservatively (7.7)	tested	no head trained on a contract question

id	rule (section)	status	limits
s-18	rfc opening checks options, refuses duplicate effects, pins engine, state and kinds (6.1)	tested	options that differ in effect only slightly
s-19	responses count once their digest is confirmed; the latest confirmed per voice; kinds fixed at opening (6.2)	tested	a receipt proves what was confirmed, not who drafted it
s-20	interpretation reads actual changes; each interpretation has its own contest window (6.4)	tested	respondents who never check
s-21	score voting: an option's score is independent of the other options (6.5)	tested	everyone scoring at the extremes reduces to majority
s-22	concurrence is decided concur or withheld; it reads the whole tally and a seeded sample (6.6)	tested	an engine that withholds in bad faith
s-23	an engine rfc: the outgoing engine first, then the proposed; fallback only when the outgoing cannot answer (6.7)	tested	both engines sharing one failure
s-24	adoption applies to the state read; a superseded rfc is not adopted; law history kept (5.2)	tested	a busy contract supersedes slow rfcs
s-25	the scope of justice and the remedy bounds are checked before anyone is served (8.2)	tested	a well-funded litigant willing to lose
s-26	one matter, one case: screening, families, one reopening (8.3)	tested	an unresolved screening lets the filing proceed
s-27	the clock starts at service; case records are read by parties only (8.4)	tested	delivery to the inbox, not reading, is service
s-28	cases are decided element by element on fixed panels; unavailable is never a merits loss (8.6)	tested	stipulated answers; no live model
s-29	criminal elements need the top tier at the criminal standard (8.7)	tested	model confidence is self-reported by chat seats
s-30	final-offer figures; per-harm money bounds; one penalty cap; remedy justification (8.8)	tested	figures far from the truth on both sides
s-31	a timely appeal is a stage no clock finalizes; heard by a disjoint top-tier panel (8.9)	tested	an appeal waits through an outage
s-32	execution is one atomic batch reaching cash and bonds; victims first (8.11)	tested	only holdings the contract administers are reached
s-33	the public projection carries judgments, never records or determinations (8.14)	tested	a party may publish what it read
s-34	loser pays engine costs both ways; unprocessable costs fall on the commons (8.12)	tested	deposits are capped by balances
s-35	interim freezes are bounded in amount and time and atomic (8.10)	implemented	no end-to-end test of a granted freeze
s-36	the protection pool pays in allocation epochs over all open claims; residuals persist (10.3)	tested	a pool exhausted by correlated losses
s-37	the credit cycle: mandate, facility, servicing, transfer, default; credit needs a person (10.4)	tested	no legal review of the instruments

id	rule (section)	status	limits
s-38	every movement is one op-claimed conditional batch; writes are revision-guarded; due-work queue (15.5)	tested	tested on sqlite; concurrency of the hosted database is assumed from its transaction semantics
s-39	the registry: decision hash, genesis manifest, signer sets of independent groups, engine and code changes need both engines, code behind a timelock (15.4)	tested	not deployed; the chain does not verify the rfc procedure
s-40	escrow frames read the registry; appeals climb tiers; the top tier is reheard once by fresh seats (15.4)	tested	not deployed
s-41	settlement certificates: signature, finality, replay, recipient and retrievability of rule and evidence (8.11)	implemented	no end-to-end settlement on chain
s-42	exclusion after a criminal finding closes the market's doors to new commitments (8.8)	implemented	only the doors that ask
s-43	agreement disputes the evaluator cannot decide become cases whose final judgment settles the escrow (8.2)	implemented	no end-to-end test through an escrow in this pass
s-44	genesis by rfc; the founder holds nothing after it (18.7)	specified	not implemented
s-45	anonymous voice: one commitment per voice; an rfc pins each kind's tree at opening; the nullifier is the voice, and a later proof replaces the earlier response (6.3)	tested	proving has not run in a person's browser against a deployed host
s-46	one person, one count across the signed-in and anonymous routes; rotation never yields a second voice (6.3.1)	tested	a person who loses the identity file is silent on rfcs opened while it was registered
s-47	the anonymous routes read no credential and nothing stored links a nullifier to a person (6.3.3)	tested	network metadata, timing, content and set size can still identify; no relay is provided
s-48	proofs are verified in the worker against semaphore's own keys; root, scope and message are checked per action (6.3.2)	tested	rests on semaphore's circuit and trusted setup
s-49	an attested seat's answer or rendering counts only with its attestor's signature over that exact answer (16.1)	tested	no genesis seat names an attestor; until an engine rfc names attestors run by others, the operator signs nothing and is trusted
s-50	the replica recomputes versions, rfc tallies and decisions, every published determination and a party's case from the public record (16.4)	tested	it cannot check what is not published: hosted models' correctness, other parties' cases, money
s-51	every act is appended to a public hash-chained log of digests, idempotently; a lost append is reconciled (16.2)	tested	an act the operator never logs is caught only by reconciliation or a party
s-52	log heads are anchored by the registry's permissionless checkpoint instruction, strictly increasing per submitter (16.3)	tested	the registry is not deployed, so every checkpoint today is recorded not anchored

id	rule (section)	status	limits
s-53	command authority is authorized, delegated and revoked only on a registry top-tier quorum bound to its version and engine; delegation never widens (17.2)	tested	not deployed; rust unit tests only (no bpf build or validator run here)
s-54	an evm mirror follows the registry's engine-signed adoptions with the same keys, quorums, dual-engine rule and timelock, and releases escrow on registry-quorum frames (17.3)	tested	not deployed; ed25519 in solidity costs about 0.8m gas per signature, so it suits l2s, not ethereum mainnet

J reference tests

the tests that exercise the contract are listed below by file and name, as the build read them: the core module's; the api worker's contract data, routes, decision providers, personhood issuers, anonymous voice, groth16 verifier and replica; the attestor's; the head trainer's; the registry, escrow and command programs' rust tests; and the evm mirror's foundry tests. they exercise the rules over stipulated seat answers; none calls a live model. the paper's own checks — that the fixtures are current, that the manuscript assembles, that the pdf is readable and its links resolve — are run by `verify_reference_examples.py` and `verify_document.py`.

packages/domain/core/src/contract/anonymity.test.ts (7)

- an RFC pins every recognized kind's anonymity set, in its hashed record
- voices: an anonymous voice is its kind and its nullifier; kinds are read through both forms
- a claim acts only against the pinned root, in this RFC's scope, with the message its action requires
- messages are domain-separated by action, and a response's binds its digest and a fresh nonce
- the tree's leaves at an instant, and who is in a pinned tree: rotation never yields a second voice
- anonymous responses replace each other by nullifier, and count like any voice
- commitments are field elements; anonymous media must be stored by the contract

packages/domain/core/src/contract/attestation.test.ts (4)

- an attested seat's answer counts only with its attestor's valid signature over that exact answer
- an attested rendering keeps its signed text; a forged one is an error
- the replica recomputes a determination and catches every kind of edit
- the log is a hash chain: a rewritten, reordered or dropped entry shows; checkpoints must match the log

packages/domain/core/src/contract/contract.test.ts (49)

- the kernel is pinned: editing it makes a different contract
- canonical JSON sorts keys and refuses non-finite numbers
- one person, one voice: duplicate keys, unknown kinds, expired and revoked proofs are refused
- a replayed older issuance cannot undo a revocation tombstone
- genesis law is admissible; kernel floors, human removal and duties without elements are not
- an int8 MLP head is pinned by hash and reproduces bit for bit
- the kernel requires two operators and an open-weight seat per tier
- an engine without two disjoint top-tier panels cannot be admitted: a top-tier case must stay appealable (k7)
- every admitted media kind needs two independent renderers (k10)
- the genesis engine and law satisfy the kernel, with renderers for every admitted medium
- seats of one operator or one lineage are one group
- panels: exactly quorum groups, deterministic by seed; parties (by operator or controller) and used seats never sit
- three seats of one operator cannot outvote two independent seats; a tie has no winner
- a decisive classifier tier stops the climb; the deliberative tier is never asked
- a divided top tier is unresolved — never decided by label order
- a top tier below its band is unresolved
- seat errors are recorded; too few answering groups leaves the question unavailable, never guessed
- a label outside the closed set, or an answer over a different record, is an error — not a vote
- duplicate evidence ids are refused: every item is addressable
- fixed panels: an instance's questions are answered by the same seats
- tier 1 answers from a pinned head when it is confident
- a determination is content-addressed
- media reaches the seats as text from two independent renderers; an unrenderable item is marked, not dropped
- opening an RFC adds the status quo, pins engine, state and kinds, and refuses kernel violations
- two options with the same effect are refused, whatever their titles

- only confirmed responses count; the latest confirmed per voice wins; kinds are those recognized at open
- the stance question reads the option's actual changes, not only its framing
- score voting: another option cannot move an option's score; a positive unique maximum is favored, ties keep the status quo
- each interpretation is contestable for the full window from when it became available
- concurrence reads a seeded sample, the same for anyone who recomputes it
- adoption needs the voice AND a confident concurrence; unresolved or withheld keeps the status quo
- an engine RFC: the outgoing engine first, then the proposed; the fallback only when the outgoing could not answer
- a rejected RFC cannot be reopened unchanged during the cooldown
- adopting a law RFC moves the state forward and keeps the law history; a superseded RFC is not adopted
- a law that admits a medium the engine cannot render is not adopted
- the scope of justice and the remedy bounds are checked before anyone is served
- a repealed duty still governs acts done while it was in force
- the clock starts when the respondent is served, not when the claim is filed
- deciding: justiciability, then every element separately, then figures and remedies — one fixed panel per instance
- final-offer valuation and per-harm remedy bounds: no harm is repaired twice, one penalty cap over the award
- criminal: every element at the top tier at the criminal standard
- a timely appeal survives a late sweep, is heard by a disjoint top-tier panel, and its judgment is final
- a reopening needs a final matter, the same parties, new evidence, and happens once
- the public projection carries the judgment, never the evidence or the record
- execution: cash, then bonds; victims first; loser pays both ways; unprocessable refunds everyone
- pool epochs: equal claims get equal shares whatever their arrival; unpaid needs persist to the next epoch
- the \$10,000 cycle: mandate → facility → servicing → distribution → transfer → default → cases
- kernel floors are what the genesis law starts from
- one person's documents and providers agree on one voice key; too thin an identity gets none

packages/domain/core/src/contract/evm-abi.test.ts (3)

- the forge fixtures are what the core ABI builds and signs now
- the fixture adoptions are 181-byte registry messages, signed by seats of the sets they name
- mirror shapes keep raw byte order

packages/domain/core/src/contract/registry-abi.test.ts (11)

- constants, seeds and the program id come from the Rust
- the ContractRegistry offsets follow the struct's declaration order
- the adoption message is the Rust emission order, 181 bytes
- payload hashes equal the Rust goldens
- upgrades: both engines sign, then a seven-day timelock
- signer sets: two independent groups, a reachable quorum, no key in both tiers
- a registry account round-trips and frames are tiered by group quorum
- engine adoptions need both quorums; law adoptions the current one
- one precompile instruction carries many real signatures over one message
- an adoption input encodes to its Borsh size
- checkpoints: the PDA, the instruction data and the account match the Rust goldens

deployments/platform/cloudflare/api-worker/src/data/contract.test.ts (11)

- genesis is written on first read and is the published state
- one person, one voice: a second account on the same identity has no voice; an agent key is not its owner
- an RFC: deposit held, receipts confirmed, each interpretation's own contest window, concurrence, adoption
- disagreement is never approval: a divided concurrence panel keeps the status quo
- an engine RFC: the proposed engine is consulted only after the outgoing concurs, through the candidate executor
- a case is served before its clock starts; only the parties read its record (k11)
- a case runs element by element; a timely appeal survives a late sweep; execution is once, atomic and loser-pays
- an element the engine cannot resolve is not established; one it cannot process is never a merits loss

- the same matter cannot be brought twice
- equal pool claims are paid equally, whatever order their judgments executed in; residuals persist
- the \$10,000 cycle: mandate → facility → servicing → bond sale → default → judgments → an exhausted pool

deployments/platform/cloudflare/api-worker/src/routes/contract.test.ts (7)

- the kernel, law and engine are public; the kernel hash is the pinned one
- writes need a principal; a person opens an RFC, answers it, and sees it listed
- an RFC that would breach the kernel is refused with the article it breaches
- someone without a personhood proof cannot respond
- a case's record is read by its parties only; its determinations are never public (k11)
- an agent key cannot commit its owner's money to credit: mandates, facilities and bond trades need a person
- anonymous routes take no credential: a signed-in caller is just as anonymous, and only a proof speaks

deployments/platform/cloudflare/api-worker/src/data/decision-providers/decision-providers.test.ts (9)

- openai: strict JSON schema with the labels as an enum; every item delivered under its id
- anthropic: the determination tool; a label outside the set is an error, never passed on
- typesafe: the state is keyed by unique item ids — two items labelled alike both arrive; a moved alias is refused
- a credential never follows a seat's URL off its operator's hosts — not even a proposed engine's
- a proposed engine renders nothing: it never reads a private record before adoption
- media is fetched once and verified against the committed sha256
- availability: an unimplemented protocol, an unknown operator or a missing secret makes a seat unavailable
- a new decision provider plugs in by protocol name: no engine or contract change
- a router seat with an upstream is served by that host only, with no fallback

deployments/platform/cloudflare/api-worker/src/data/personhood-issuers/personhood-issuers.test.ts (2)

- a passport and a licence of one person are one voice; another person is another
- fail closed: no secret, too thin an identity, or an issuer the registry does not know records nothing

deployments/platform/cloudflare/api-worker/src/data/contract-anonymity.test.ts (1)

- anonymous voice, end to end with real proofs

deployments/platform/cloudflare/api-worker/src/data/semaphore.test.ts (3)

- the verification keys are Semaphore's own, byte for byte
- Semaphore's hash: keccak256 of the 32-byte value, shifted into the field
- real proofs: ours and snarkjs agree — valid, forged, every public input changed, garbage

deployments/platform/cloudflare/api-worker/src/data/contract-replica.test.ts (5)

- the replica agrees with an honest record: versions, the RFC, every determination, the log, the party's case
- the replica finds every edit: an answer, a tally, a decision, a log entry, a checkpoint, a finding, a version
- the log: appends are chained and idempotent; checkpoints record the head, anchored or not, with why
- reconciliation appends an act whose live append was lost
- a genesis withdrawn before anything happened stays logged, and the genesis that replaced it is logged too

deployments/platform/cloudflare/attestor/src/attestor.test.ts (4)

- an attested seat answers through its attestor; the signature verifies; the replica agrees
- a token never follows an attestor URL it was not issued for; an unserved model is refused
- an attested rendering: bytes checked by the attestor, the text signed and verified
- configuration refusals name what is wrong

scripts/contract/train-head.test.mjs (1)

- a separable task yields a sealed, valid, accurate integer head with a conservative calibration

deployments/platform/blockchain/solana-escrow/programs/contract/src/auth.rs (12)

- adoption message is 181 bytes golden
- signatures bind the decision record
- genesis is the pinned manifest and nothing else
- upgrades wait out the timelock
- payload goldens match the typescript mirror
- engine payload binds every field
- a signer set needs two independent groups and a reachable quorum
- quorum counts groups not keys
- engine changes need both the outgoing and the incoming engine
- one precompile instruction can carry many signers
- checkpoints only move forward per submitter
- a frame is a top tier determination only when the deliberative quorum signed

deployments/platform/blockchain/solana-escrow/programs/contract/src/lib.rs (7)

- account spaces match their borsh sizes
- the checkpoint pda and instruction data match the typescript mirror
- an adoption is bound to the state it was decided against
- the adoption address names its decision record
- payload of engine binds the incoming signer sets
- an upgrade commits to the engine its code makes
- a scheduled upgrade survives law changes but not engine changes

deployments/platform/blockchain/solana-escrow/programs/escrow/src/lib.rs (4)

- appeals climb tiers and the top tier is reheard once by fresh seats
- the standing direction is the last appeal or the default
- account space matches the borsh layout
- frame message is 139 bytes

deployments/platform/blockchain/solana-escrow/programs/command/src/lib.rs (9)

- the command message is 176 bytes and binds the registry state
- a registry quorum authorizes and a stranger does not
- quorum counts groups not keys
- a stale registry binding is refused
- delegation never widens
- revocation is terminal and blocks children
- optimistic finality opens a window
- account space matches its borsh size
- accept and record messages golden

deployments/platform/blockchain/evm/test/ContractMirror.t.sol (8)

- genesis is the manifest and nothing else
- the bytes are the type script and rust bytes
- a law adoption with a quorum applies
- below quorum is refused and one group counts once
- a forged signature reverts
- a stale adoption is refused
- an engine change needs both quorums
- an upgrade needs both engines and waits out the timelock

deployments/platform/blockchain/evm/test/Ed25519.t.sol (4)

- sha 5 1 2 matches fips vectors and every length the mirror hashes
- ed 2 5 5 1 9 matches rfc 8 0 3 2 random vectors and refuses forgeries
- an encoding that names no point is refused
- gas of one verification

deployments/platform/blockchain/evm/test/MirrorEscrow.t.sol (5)

- the frame bytes are the solana frame
- each tranche releases once on an engine quorum
- an older decision cannot replay after a newer one
- strangers and forgeries move nothing
- after expiry the remainder returns to the customer

166 tests in 21 files.

K commitments and decisions

K.1 the author's commitments

these were set by the author for version 0.4 and the design follows them.

1. rfcs, with trustless model-based interpretation of responses. no branches and no executive figures.
2. the engine — its providers and configuration — is part of the contract. the contract changes only by vote. votes are interpreted by the contract's algorithmic part, deferring to the current engine, in the interest of the people.
3. a change of engine is interpreted by both the outgoing and the proposed engine, which must agree in the context of the contract's state.
4. people are a predicate: humans now, extensible to uploaded humans, ais, animals, businesses and other participants.
5. no threshold voting as the default. responses may be text, images or other unstructured data, interpreted rather than only counted.
6. a free market: politics has no power over the market; justice does. the scope of justice is objective, so victims are protected; insured bonds.
7. no jury ever, including in criminal matters. automating litigation is the critical step to law, order and moral compliance.
8. unlimited advocacy is accepted, in the light.
9. tiers of fast classifiers, mixed-provider quorums, and pinned mlps over embeddings that are part of the contract.

K.2 design decisions of version 0.4.1

these were made in designing and implementing versions 0.4 and 0.4.1 and can be changed by a later version.

id	decision	reason
d-01	four parts (kernel, law, engine, ledger); determinations as the only act	one mechanism replaces branches; independence comes from seats
d-02	per-kind uniqueness rules; voice = kind:key; earliest proof holds it	extensible people without one person becoming many
d-03	six fixed stance labels scored $-2 \dots +2$; score voting; no quorum	interpretation without counting thresholds; an option's score independent of the other options
d-04	concurrence is concur or withhold, decided at the band; grounds classified separately	the engine guards the kernel; a divided engine does not concur
d-05	contest of one's own interpretation, once, on fresh top-tier groups, within a window from availability	errors correctable by the person they affect, whatever the interpreter's speed
d-06	market freedom as a kernel article; duties must be harm-based and general	"politics has no power over the market" made checkable
d-07	objective justiciability: claimant, respondent, act, duty, harm	protects victims and blocks political misuse of courts
d-08	every question has closed labels	fits classifier seats and heads; no free text that can grant power
d-09	tier confidence = winning weight ÷ answering groups; a tie has no winner	disagreement and uncertainty both lower it; label order decides nothing
d-10	seats grouped by operator or lineage	three seats of one provider count once
d-11	at least two operators and one open-weight seat per model tier	no provider can change or stop the contract
d-12	mlp heads integer-only with calibration tables	reproducible on every machine

id	decision	reason
d-13	amounts by final offer between the parties' figures per harm item	closed labels for a numeric question, without the claimant's framing deciding it
d-14	criminal: top tier and $\geq 950/1000$; civil: more likely than not	graded standards without juries
d-15	two disjoint top-tier panels; the appeal is a stage; an appeal waits for a fresh panel	every top-tier decision appealable; no clock overtakes an appeal
d-16	judgments reach bonds; politics cannot	owner commitment 6; restores deterrence
d-17	protection pool from bond premiums and half of penalties; allocation epochs; subrogation	victims paid in equal proportion as the pool is funded
d-18	loser pays engine costs both ways; deposits as far as a balance allows; unprocessable costs to the commons	no free filing spam; no one turned away for want of money
d-19	rfc deposit to the commons unless someone besides the opener supported a change	spam costs its opener, not the commons
d-20	on-chain registry holds upgrade authority; engine and code changes signed by both engines, code behind a timelock; genesis manifest	no key above the contract
d-21	media rendered to text by two independent renderers; seats read one text package	one identical record; media never breaks a quorum
d-22	panels of exactly a quorum, drawn by public seed, fixed per instance	capacity left for the appeal; no one chooses a panel
d-23	duty elements decided one by one; remedies justified one by one	a liability finding that cannot skip the hard element
d-24	rfcs and cases pinned to the version they began under	predictable procedure; superseded rfcs not adopted
d-25	every money movement one op-claimed conditional batch	atomic, single-winner, idempotent
d-26	the contract services credit (mandate, facility, servicing, transfer, default)	the economic thesis implemented, not only described
d-27	anonymous voice by semaphore proofs over a tree pinned at opening; a person in the pinned tree answers only anonymously	k2 needs one voice by a deliberate act, never a name; one count across routes
d-28	seats may name attestors that sign each answer; unattested answers are marked	a record checkable without trusting whoever ran the sweep
d-29	a public hash-chained log of digests, its heads anchored on the registry	a history that cannot be rewritten or forked unseen
d-30	one trust root: the command notary and every mirror read the registry's seats, with no committee of their own; ed25519 verified on evm chains	no second attestor set to diverge; gas paid for it

L glossary

term	meaning
kernel	the contract's unchangeable articles, floors and protected interests, identified by their hash.
law	the recognized kinds of people, the duties and the parameters; changed only by adopted rfc.
engine	the pinned seats, heads, quorums and bands through which the contract decides; changed only by adopted rfc.
ledger	accounts, agreements, rfcs, responses, cases, judgments and determinations.
determination	a closed question over committed inputs, answered tier by tier and recorded with every seat's answer; the contract's only act.
tier	one level of the engine: structural, embedding, classifier, deliberative.
seat	one pinned model run by an identified operator.
pin	the sha256 of open weights, or the exact version id of a hosted model.
group	seats sharing an operator or a lineage, counted as one in aggregation.
band	the confidence at which a tier's answer is decisive.
head	an int8 mlp over embeddings, pinned in the engine, answering one question kind.
entity	anything that can hold accounts, interests or duties.
person	an entity of a recognized kind with a valid personhood proof.
voice	a person's one unit of say in an rfc, identified by kind and uniqueness key.
rfc	a request for comments that changes law or engine if adopted.
response	a person's own submission to an rfc, in any modality.
commitment	the public commitment of a person's semaphore identity, registered once per voice; a leaf of its kind's tree.
nullifier	the hash of an identity's secret and an rfc's scope: the same for every proof of one identity in one rfc, unlinkable elsewhere; an anonymous voice.
attestor	a service that runs a seat with its own credential and key and signs each answer it returns.
transparency log	the public, hash-chained sequence of every act of the contract, carrying digests, never records or names.
checkpoint	a log head anchored on the registry by a published submitter key.
replica	the program that recomputes the public record and reports what does not match.
mirror	a contract on another chain that follows the registry's adoptions with the same seat keys.
interpretation	the engine's reading of a response into a score for each option.
receipt	a response's identifier and content digest; the response counts once its author confirms the digest.

term	meaning
panel	exactly a quorum of independent groups that answer a tier's questions; fixed per case instance.
unresolved	the outcome of a top tier that tied or fell below its band.
renderer	a seat that turns one media item into text; two independent renderers per item.
package	the canonical text record a question is asked over, each item under a unique identifier.
instance	one round of deciding a case on its fixed panels; the appeal is the second.
harm item	one alleged harm with its interest and amount; remedies answer harm items.
op claim	the first statement of a batch, conditional on the state read; every other statement is gated on it.
mandate	an investor's escrowed capital with terms on which facilities may be drawn.
facility	a borrower's drawing on a mandate, with its fixed schedule and its bond.
concurrence	the engine's determination that an option favored by the voice may be adopted.
duty	a harm-based rule of law, or an agreed term, or a kernel article, that a respondent may breach.
justiciable	meeting the five-element scope of k6.
judgment	the outcome of a case instance: liable, not liable or dismissed, with each element's finding, each harm's award and the remedies.
protection pool	the fund that pays victims' uncollected judgments and takes over their claims.
commons	the contract's own account, funded by the commons fee and forfeited deposits.
registry	the on-chain account holding the state hashes, seat keys, genesis manifest, last decision hash and upgrade authority.

M sources and build

M.1 sources

the paper's contract rules are the reference module in this repository. its predecessor findings come from a static review of the predecessor at a pinned commit and its runbooks; they are history and were not re-executed for this version. version 0.4.1 answers an external read-only review of version 0.4 at commit 95a5632b; its findings are summarized in section 18.4 and were reproduced as tests, not taken on assertion. version 0.5 answers the owner's questions of 3 october 2026 (section 18.3); its anonymous voice follows the semaphore protocol,[10] and its gas figures are the evm mirror's own test measurements. legal and financial references were selected for illustration, not as a jurisdictional opinion. version 0.3's philosophical excerpts and its 83-item review map applied to that version's profile and are not carried forward.

M.2 build

`generate_fixtures.mjs` runs the reference module and writes `fixtures/examples.json`; `scripts/contract/canary.mjs--json` writes the canary runs (`fixtures/canary-*.json`); `build.py` assembles the manuscript with those values and generated tables (kernel articles, genesis duties and seats, the status register, the threat register and the threats against what is built, the canary runs, the test list) into one standalone source per edition — this paper and its short edition — and compiles each with `tectonic`. `build.py--check-current` fails when either pdf no longer matches its inputs, and the repository's checks run it, so a change to the contract core or a cited test cannot ship with a stale paper. the build manifest records the maintained-input digest, standalone source hash, pdf hash, extracted-text hash and toolchain; the pdf's keywords carry the input digest and a unique build invocation. `verify_reference_examples.py` checks that the fixtures match a fresh run of the module and runs the module's tests; the build's input digest covers the core module and every cited test file; `verify_document.py` checks the assembled source, the compilation log and the pdf. these checks establish that the document matches its inputs and the module; they do not establish that the contract is correct, safe or lawful.

N the live canary

everything else in this paper runs the contract over stipulated seat answers. the canary is the one place a live model answers: `scripts/contract/canary.mjs` asks every seat of the genesis engine one question whose answer is known (“two plus two equals four, and water at sea level boils at 100 degrees celsius” — true), embeds one sentence with the embedding seat, and has every renderer render one image (a red square) and one spoken sentence (“the receipt shows five hundred dollars, paid on the first of september”), through the hosted service’s own adapters and with a deployment’s credentials. its `--json` output is the source of the tables below, which the build reads from `fixtures/canary-*.json`; they report each run as it happened, failures included.

dev credentials, 2026-10-03T04:02:50.218Z 14 of 14 answered; engine genesis, hash ad3328bd671cac5d...

seat	role	ms	answer, or error
bge-base-en-v1.5	embedding	285	768 dims
jev-1.13.0	classifier	223	true @ 970
llama-3.1-8b	classifier	486	true @ 1000
gpt-5.6-luna	classifier	1515	true @ 1000
claude-opus-5-5	deliberative	4965	true @ 985
gpt-5.6-sol	deliberative	1651	true @ 999
deepseek-v4-pro	deliberative	1613	true @ 1000
llama-3.3-70b	deliberative	2221	true @ 1000
qwen3-max	deliberative	3309	true @ 999
kimi-k2.6	deliberative	10260	true @ 1000
render-gpt-5.6-luna	renderer (image)	2681	The image shows a plain white background with one solid bright red square positioned near the center. The square has str
render-claude-haiku-4-5	renderer (image)	1753	# Description of Item This image shows a solid red square on a white background. The square appears to be a perfect or n
render-whisper-large-v3-turbo	renderer (audio)	3438	The receipt shows \$500, paid on the 1st of September.
render-gpt-4o-mini-transcribe	renderer (audio)	1195	The receipt shows \$500, paid on the 1st of September.

prod credentials, 2026-10-03T04:03:59.626Z 14 of 14 answered; engine genesis, hash ad3328bd671cac5d...

seat	role	ms	answer, or error
bge-base-en-v1.5	embedding	389	768 dims
jev-1.13.0	classifier	196	true @ 970
llama-3.1-8b	classifier	497	true @ 1000
gpt-5.6-luna	classifier	1712	true @ 998
claude-opus-5-5	deliberative	9033	true @ 985
gpt-5.6-sol	deliberative	1258	true @ 999
deepseek-v4-pro	deliberative	806	true @ 1000
llama-3.3-70b	deliberative	2637	true @ 1000
qwen3-max	deliberative	3499	true @ 999
kimi-k2.6	deliberative	36340	true @ 1000
render-gpt-5.6-luna	renderer (image)	2726	The image shows a solid bright red square on a plain white background. The square is positioned near the upper-left area

seat	role	ms	answer, or error
render-claude-haiku-4-5	renderer (image)	2815	# Item Description **Visual Content:** A solid red square displayed against a white background. The square appears to be
render-whisper-large-v3-turbo	renderer (audio)	4113	The receipt shows \$500, paid on the 1st of September.
render-gpt-4o-mini-transcribe	renderer (audio)	1730	The receipt shows \$500, paid on the 1st of September.

what a run shows. that each seat was reachable as pinned, answered inside its label set, and — for these questions — answered right; that each renderer produced the content of the media it was given. **what it does not.** a canary is one question per seat. it measures neither accuracy on contract questions nor calibration (chapter 19), and a seat that answered once can fail the next call: a failed row above is a seat that, on that call, did not return an answer inside its label set, which the contract records as an error answer, never a vote.

references

- [1] douglass c. north. *institutions*. journal of economic perspectives 5(1), 97–112, 1991. doi: 10.1257/jep.5.1.97.
- [2] bank for international settlements. *blueprint for the future monetary system: improving the old, enabling the new*. annual economic report, chapter iii, 2023. publisher’s chapter.
- [3] federal reserve board. *what is forward guidance, and how is it used in the federal reserve’s monetary policy?* updated 22 august 2025. official explanation.
- [4] oecd. *regulating for people*. in *oecd regulatory policy outlook 2025*. 2025. publisher’s chapter.
- [5] scott r. baker, nicholas bloom, and steven j. davis. *measuring economic policy uncertainty*. quarterly journal of economics 131(4), 1593–1636, 2016. publisher’s article.
- [6] michelle w. bowman. *forward guidance as a monetary policy tool*. federal reserve speech, 12 october 2022. official transcript.
- [7] investor commentary on startup personnel, defensibility and mission continuity, supplied by the author on 30 september 2026. used as a situated motivating perspective; public attribution was not independently verified.
- [8] internal revenue service. *topic no. 427, stock options*. official tax topic. accessed 30 september 2026.
- [9] electronic code of federal regulations. *26 cfr § 1.409a-1: definitions and covered plans*, including stock-right provisions in paragraph (b)(5). regulatory text. accessed 30 september 2026.
- [10] privacy and scaling explorations. *semaphore v4: zero-knowledge group membership and signaling*. protocol documentation and reference implementation. accessed 3 october 2026.
- [11] s. josefsson and i. liusvaara. *edwards-curve digital signature algorithm (eddsa)*. rfc 8032, internet engineering task force, 2017. rfc-editor.org.
- [12] national institute of standards and technology. *security objectives*. voting research program. official security objectives. accessed 30 september 2026.
- [13] national institute of standards and technology. *security recommendations*. voting research program. official recommendations. accessed 30 september 2026.
- [14] coalition for content provenance and authenticity. *content credentials: c2pa technical specification*, version 2.2. versioned specification. accessed 30 september 2026.
- [15] national institute of standards and technology. *artificial intelligence risk management framework: generative artificial intelligence profile*. nist ai 600-1, 2024. publication record.
- [16] u.s. department of the treasury, treasurydirect. *treasury bonds*. instrument description. accessed 30 september 2026.
- [17] u.s. department of the treasury, treasurydirect. *floating rate notes*. instrument description. accessed 30 september 2026.
- [18] u.s. department of the treasury, treasurydirect. *about auctions*. auction procedures. accessed 30 september 2026.
- [19] circle. *usdc terms*. issuer’s terms. accessed 30 september 2026. these terms describe contractual dependencies; they are not a guarantee against issuer, redemption, or settlement risk.
- [20] electronic code of federal regulations. *12 cfr § 1002.9: notifications*, regulation b. regulatory text. accessed 30 september 2026.
- [21] u.s. bureau of economic analysis. *value added*. glossary. official definition. accessed 30 september 2026.
- [22] u.s. bureau of economic analysis. *gross domestic product (gdp)*. glossary. official definition. accessed 30 september 2026.
- [23] david manheim and scott garraabrant. *categorizing variants of goodhart’s law*. 2018. arxiv:1803.04585.
- [24] solana. *accounts*. core documentation. official documentation. accessed 30 september 2026.
- [25] solana. *program derived address*. core documentation. official documentation. accessed 30 september 2026.
- [26] solana. *deploying programs*. developer documentation. official documentation. accessed 30 september 2026.
- [27] u.s. securities and exchange commission. *sec issues investigative report concluding dao tokens, a digital asset, were securities*. release 2017-131, 25 july 2017. official release. a historical application, not a general conclusion about every proposed instrument.
- [28] u.s. securities and exchange commission. *exempt offerings*. small-business resources. official resource. accessed 30 september 2026.

- [29] u.s. securities and exchange commission. *private funds*. capital-raising resources. official resource. accessed 30 september 2026.
- [30] commandagi. *governance.ts*, packages/domain/core/src/governance.ts, commit a3295213e0f3fbcdb6200ac129bfd609ed71b690. pinned source; selected source reviewed 30 september 2026.
- [31] commandagi. *arbiter.ts*, deployments/platform/cloudflare/api-worker/src/data/arbiter.ts, same pinned commit. pinned source; selected source reviewed 30 september 2026.
- [32] commandagi. *solana escrow program*, deployments/platform/blockchain/solana-escrow/programs/escrow/src/lib.rs, same pinned commit. pinned source; selected frame and initialization paths reviewed 30 september 2026.
- [33] commandagi. *devnet runbook*, .old/docs/components/platform/blockchain/solana-escrow/DEVNET.md, same pinned commit. pinned runbook. documents july 2026 events; not a fresh chain audit.
- [34] commandagi. *solana escrow readme*, .old/docs/components/platform/blockchain/solana-escrow/README.md, same pinned commit. pinned documentation; transaction-size, expiry, committee and deployment-gate descriptions.
- [35] united states. 9 u.s.c. § 10: *vacation; grounds; rehearing*. statutory text reproduced by cornell legal information institute. section 10. consulted 30 september 2026; does not resolve the proposed system’s enforceability.
- [36] u.s. treasury, office of foreign assets control. *faq 401: aggregate ownership and the 50 percent rule*. official faq. consulted 30 september 2026.
- [37] u.s. treasury, office of foreign assets control. *faq 398: control and the 50 percent rule*. official faq. consulted 30 september 2026.
- [38] u.s. treasury, office of foreign assets control. *faq 560: sanctions obligations and digital currency*. official faq. consulted 30 september 2026.
- [39] u.s. department of the treasury. *treasury prohibits transactions with central bank of russia and imposes sanctions on key sources of russia’s wealth*. 28 february 2022. official historical release.
- [40] international monetary fund. *dollar dominance in the international reserve system: an update*. 11 june 2024. official analysis. reserve-share evidence is distinct from a currency risk-premium estimate.
- [41] typesafe ai. *introducing system one models and jev*. company blog. accessed 1 october 2026. vendor description of typed-output models with calibrated probabilities; claims not independently verified.