The Execution Gap

A Case for Cryptographic Authority Infrastructure

Gary Chigaros
OrgForge, Inc.
orgforge.io

ABSTRACT

Every era of organized human activity has lived with a governance lag: the gap between the moment a rule is articulated and the moment an action is governed by it. For most of institutional history, this lag was bounded by the pace of human action. Detection arrived before consequences compounded beyond recovery. That condition is changing.

Modern organizations execute at machine speed through software agents that hold real credentials, call real APIs, and initiate real payments without per-step human review. The governance apparatus built for human-paced decision-making has not been rebuilt for the pace at which these agents now act. The result is a structural gap between the rules organizations have written and the rules their systems enforce at the moment of execution.

This paper argues that closing this gap requires a new category of infrastructure: one that makes authorization a cryptographic property of an action rather than a social judgment about it. The argument proceeds from the historical failure of institutional governance, through the structural lesson of Bitcoin, to the architectural requirements of a machine-readable organizational constitution, and outward to the federal, sovereign, and international contexts where that architecture becomes consequential. The paper also addresses the genuine limits of this approach, including the boundary between procedural correctness and material truth.

OrgForge is an early working instantiation of this architecture. The category it represents is cryptographic authority infrastructure.

SECTION 1

The Speed of Machines, the Pace of Rules

At 2:32 in the afternoon on May 6, 2010, a mutual fund manager in Kansas initiated a sell order for 75,000 E-Mini S&P 500 futures contracts. Under standard human-mediated protocols, a position that size required roughly five hours of careful execution to avoid destabilizing the market. Instead, an automated algorithm processed the entire order without regard to price or time, saturating the order books in minutes. High-frequency trading systems, detecting the unusual flow, began passing the contracts among themselves at machine speed, creating a feedback loop so intense that a single contract changed hands 27,000 times in fourteen seconds. Within twenty minutes, nearly a trillion dollars in market value had vanished. Then, almost as quickly, the machines recalibrated and prices recovered. The event entered the financial record as the Flash Crash.[1]

The algorithms that caused it held valid credentials. Every access control check passed. The failure was a structural gap between the authority the systems had been given and the rules governing how they were supposed to use it. No mechanism required the execution systems to know what the organization had decided. They acted on their credentials and their code. The governance question went unasked.

Two years later, on August 1, 2012, a technician at Knight Capital Group deployed new trading software to the firm's production servers. He updated seven of the eight servers correctly. The eighth kept a dormant legacy function called Power Peg, dead code from a system decommissioned years earlier. A repurposed flag in the new software accidentally triggered it. For forty-five minutes, Knight Capital's systems executed erroneous orders at a rate that no human could have monitored in real time, let alone stopped. The firm lost $460 million before anyone managed to intervene. Knight Capital did not survive the year.[2]

Both failures share a precise structure. The execution systems checked credentials. They verified identity. They confirmed access. At no point did any system ask whether the actions being taken satisfied the actual rules under

which that access had been granted. The governance layer and the execution layer were separated by a gap that moved at human speed in a world that no longer did.

The gap between rule and enforcement at execution is old. What is new is the speed at which it compounds.

For most of institutional history, the governance lag (the delay between an action and a governance process's ability to respond) was measured in days or weeks. A fraudulent payment required paper, signatures, and physical transfer. An unauthorized procurement required meetings, approvals, and physical delivery. The audit cycle arrived while the situation was still recoverable. Actions were slow. So was harm.

That relationship has inverted. Modern organizational operations execute at machine speed. Payments clear in seconds. Software deploys in minutes. API calls complete in milliseconds. A trading algorithm can place thousands of orders in the time a compliance officer reads an alert. A deployment pipeline can push changes to production before an approval ticket is assigned. The governance apparatus (policies, approval workflows, compliance reviews, audit cycles) was built for human-paced decision-making. It has not been rebuilt for the pace at which organizations now act.

What currently exists as a substitute is largely reactive. Monitoring systems alert after execution. Audit functions review after the fact. Hard-coded filters on individual systems block specific known-bad actions. These are useful, but they are not governance. They cannot evaluate whether an action satisfies the actual conditional rules of an organization at the moment it executes. That evaluation (the governance question) currently happens through social process when it happens at all: a human checks, a workflow routes, a reviewer approves. The execution system then acts on the outcome of that process without being required to verify it occurred.

Artificial intelligence changes the character of this problem, not just its scale.

When a software system is a tool, a human remains the operator. The tool executes what the human instructs. When a software system is an agent, holding credentials, making decisions, calling APIs, initiating payments, and executing multi-step workflows without per-step human review, the agent is the operator. The distinction is structural. An agent with credentials can take real consequential actions at a rate and scale no human operator could match. Its authority is bounded, in practice, by the technical constraints at the interface between the agent and the system it is calling, not by what the organization's policies say, not by what the compliance manual specifies.

In most cases, that constraint is identity-based. The execution system asks whether this credential has permission to call this endpoint. If yes, the action proceeds. It does not ask whether this action satisfies the conditions under which this credential's authority was granted, or whether this action falls within the operational constraints the organization has defined for agents of this class, or whether this action complies with the risk policy updated six weeks ago.

Those are governance questions. The access control system does not ask them. The governance gap has existed for decades. Agents that act at machine speed, with no human review between authorization and consequence, make it unmanageable.

SECTION 2

How Civilizations Have Constrained Power

The problem of constraining authority is as old as authority itself.

Before written law, governance operated through reputation and memory. A chief, a priest, a council: authority was personal, visible, and bounded by the scale of what a single reputation could govern. Violations were social events. Accountability was immediate or it was nothing. The system worked inside communities small enough for reputation to function as enforcement. It failed at every scale beyond that, and it left no durable record. When the authority-holder changed, the constraints changed with them.

Written law was the first structural upgrade. The Code of Hammurabi, the Twelve Tables of Rome, the Magna Carta: each represented the same architectural move. Rules that had previously existed in the memory and discretion of rulers were externalized into a text that existed independently of any individual. You could now point to a document rather than a person. The rules became, in principle, separable from the people who made them.

This was a genuine advance. It made governance legible across geography and time. A legal system that a provincial administrator could consult in the absence of the lawgiver was more consistent than one that depended on the lawgiver's physical presence. Written rules created the possibility of challenging their application by reference to the text, rather than to the authority of whoever was applying them.

Writing rules down did not enforce them. Between the rule as articulated and the action as taken, enforcement still depended on people. The document was inert. The institutions that applied it were not.

Institutions were the second structural upgrade. Courts, bureaucracies, professional guilds, regulatory bodies, legislative oversight committees: the entire apparatus of modern governance was built to solve the problem that written law could not solve on its own. Rules needed interpreters. Interpretations needed adjudicators. Adjudications needed enforcement. Enforcement needed accountability. The institutional layer was an attempt to build chains of accountability that would constrain each level of authority by the level above it.

The economic analysis of this problem identified its structural flaw with considerable precision. The principal-agent problem, formalized by Jensen and Meckling in 1976, describes what happens when one party (the agent) is given authority to act on behalf of another (the principal).[3] Agents pursue their own interests, which diverge from the interests of principals in ways that are difficult to observe and costly to constrain. The court that is supposed to apply the law impartially has judges with careers and ideological commitments. The regulatory body that is supposed to protect the public has staff who rotate into the industries they regulate. The legislative committee that is supposed to

provide oversight has members whose reelection depends on the industries they oversee. The auditor that is supposed to certify the accounts has relationships with the management whose accounts they certify.

Douglass North spent a career demonstrating that the quality of institutional constraints on authority is the primary determinant of long-run economic and political outcomes.[4] Daron Acemoglu and James Robinson extended that framework to show that the difference between inclusive institutions, which constrain those who hold power, and extractive ones, which allow those who hold power to exploit it, explains more about the divergence of national outcomes than geography, culture, or resources.[5] The pattern across centuries and continents is consistent: governance quality is a structural property, determined by the architecture of constraints, not by the intentions of those who hold authority within them.

The institutional solution has produced the best governance outcomes in human history. It has also reproduced, at every scale and in every context, the same underlying failure mode. Every institutional solution creates a new layer of agents who can exploit the gap between rule and enforcement. The auditor can be captured. The regulator can be bought. The court can be slow, corrupt, or simply wrong. The legislature can be dominated by the interests it is supposed to constrain. Institutions constrain authority and then become the new surface through which authority escapes constraint.

The pattern across this history is not a story of villains or insufficient effort. The most serious institutional designers in history understood the principal-agent problem in practical terms even without the formal vocabulary. The separation of powers, the adversarial legal system, the independent judiciary, the free press as a check on government: each represents an attempt to build structural constraints that do not depend on the good character of whoever happens to hold power at a given moment. Each has succeeded partially. None has solved the problem.

Every institutional solution shares a common architecture: rules articulated by people, enforced by people, interpreted by people, and amended by people. The enforcement layer is always a human layer. Human

layers have human failure modes.

The question this paper is building toward is whether that architectural constraint is permanent, or whether there is a class of governance problem for which enforcement can be made native to the execution boundary rather than dependent on a human intermediary standing between rule and action.

SECTION 3

The Structural Lesson from Bitcoin

Start with a problem that has nothing to do with cryptocurrency.

Imagine you want to send someone a document. You make a copy. You keep the original. This is the fundamental property of digital information: it duplicates perfectly. For text and images, this is a feature. For money, it is a structural catastrophe. A digital dollar that copies is a dollar that can be spent twice, three times, indefinitely. It is not a dollar. It is a claim that cannot be verified.

For decades, the solution was institutional. A trusted record-keeper (a bank, a clearinghouse, a payment processor) maintained the authoritative account of who had spent what. Before a transaction could complete, the record-keeper checked the ledger. Trust in the financial system was not trust in mathematics. It was trust in the institutions maintaining the record and the people running them.

This works, mostly. It works the way all institutional solutions work: when the institution is competent and honest, the system functions. When it is not, the system fails. The 2008 financial crisis was, among other things, a demonstration of what happens when the institutions whose trustworthiness underwrites a system turn out to have been operating with interests of their own. The ledger-keepers had positions. Their record-keeping reflected those positions. The proof of authorization was only as reliable as the institution holding it.

In 2008, a paper appeared proposing a different approach: a way to make the institution structurally unnecessary for one specific class of problem.[6]

The key move was this: rather than verifying a transaction by consulting a central record-keeper, design a protocol so that validity is a mathematical property of the transaction itself.

A valid Bitcoin transaction carries a cryptographic signature, a mathematical proof generated using the sender's private key, that this specific transaction was authorized by whoever controls that key. The proof is generated by the sender. It travels with the transaction. Any party receiving the transaction can verify the proof independently, without asking anyone for confirmation. The proof is either valid or it is not. Mathematics decides, not an institution.

The double-spend problem (preventing a digital token from being spent twice) is handled by a shared ledger that no single party controls, maintained through a consensus mechanism that makes falsifying the record computationally prohibitive.[7] The specific mechanism has been analyzed, debated, and critiqued extensively. What matters for this paper is not the mechanism but what it demonstrated: for at least one important class of trust problem, institutional judgment can be replaced by mathematical proof.

Before Bitcoin, authorization was an act performed by an institution about a transaction. The institution decided whether the transaction was valid. After Bitcoin, authorization became a property of the transaction itself. The institution is optional. The proof travels with the action.

This shift, from institutional judgment to mathematical property, is the structural lesson.

It is not, at its core, a lesson about money. Money was the domain in which the lesson was demonstrated. The underlying insight concerns what it means for an action to be authorized and where the proof of that authorization lives.

INSTITUTIONAL AUTHORIZATION	CRYPTOGRAPHIC AUTHORIZATION
SOURCE OF TRUTH Institution holds the record	**SOURCE OF TRUTH** Protocol encodes validity
LOCATION OF PROOF Held by intermediary	**LOCATION OF PROOF** Travels with the action
VERIFICATION Requires institutional access	**VERIFICATION** Any party, public keys only
FAILURE MODE Capture, error, corruption	**FAILURE MODE** Mathematical or protocol failure

Bitcoin's monetary claims are contested in ways outside the scope of this paper. Its architectural contribution is not contested: it proved that for one important class of trust problem, the proof of authorization can be made native to the action rather than held by an intermediary.

The question this paper is building toward is whether that structural move is available one layer up, not at the level of monetary transactions but at the level of organizational authority. Can authorization become a mathematical property of an action rather than an institutional judgment about it?

SECTION 4

The Missing Layer

Bitcoin solved one species of a broader problem. The species it solved was financial: how to verify that a transaction was authorized without depending on a central institution to hold the proof. The broader problem is older and more general: how to verify that any action taken under delegated authority actually satisfied the conditions of that delegation at the moment it occurred.

Financial transactions are one category of action taken under delegated authority. Procurement decisions are another. Software deployments are another. Budget disbursements, regulatory filings, contract awards, agent-initiated API calls: all of them are actions taken by someone operating under authority granted by someone else, subject to conditions that the granting party specified. In every case, execution systems verify that the actor has credentials. In no case are execution systems required to verify that the

conditions attached to those credentials were satisfied.

This gap has a precise character. Access infrastructure answers the identity question: who is this actor, and do they have permission to call this interface? The governance question is different: does this specific action, at this specific moment, under the current conditions the organization has established, fall within the actual scope of authority that was granted? The first question has been solved, repeatedly and well. The second has not been addressed at the execution boundary at all. The infrastructure to provide it has never been built.

Consider what a system would need in order to answer the governance question at the execution boundary.

It would need the organization's rules in a form the system could evaluate. Policies stored in documents, governance frameworks expressed in natural language, approval processes described in procedure manuals: none of these are evaluable by a system in real time against a specific proposed action. The rules would need to be expressed as formal specifications, structured precisely enough that a function could take a proposed action as input and determine whether the rules permit it.

This boundary deserves honest treatment. Some rules resist formal specification entirely. Rules that require contextual human judgment, that admit competing interpretations, that involve weighing incommensurable values: these do not yield to machine evaluation in any meaningful sense. The territory of machine-evaluable governance is narrower than the territory of governance as organizations actually practice it.

It is, however, significantly larger than it might initially appear. Most governance failures occur in the zone of clear, unambiguous rules that simply were not enforced at the moment of execution. The spending limit was clear. The approval threshold was clear. The procurement condition was clear. The agent's operational constraints were clear. These rules failed because no mechanism required compliance at the moment an action executed. The formally specifiable zone covers the overwhelming majority of documented governance failures, even if it does not cover all governance decisions.

The second requirement follows from the first: the evaluation would need to happen at the moment of execution, against the current version of the rules.

Rules change. Governance states evolve. A rule that permitted an action last week may prohibit it this week. An approval threshold that applied under normal operations may be suspended under an emergency governance state declared this morning. Any evaluation that uses a stale version of the governance state can produce a result that does not reflect the organization's current actual rules. The evaluation must be synchronized to the canonical governance state at the moment the action is proposed.

The third requirement is that the result of the evaluation would need to travel with the action to the execution system. This is the same structural move Bitcoin made, applied one layer up. The execution system should not need to know the organization's governance rules. It should receive, along with the action request, a proof that the action was evaluated against the current governance state and found compliant. The execution system's job is to verify the proof. Verification is simpler than evaluation, and it can be performed by any party with the relevant public keys, without access to the organization's internal governance infrastructure.

The proof would need to be unforgeable: mathematically impossible to construct a valid authorization for an action the rules did not permit, to reuse a valid authorization for a different action than the one evaluated, or to present an authorization issued under a previous version of the rules as current.

A reader who has worked through these requirements is already holding the architecture. There is a governance specification: the organization's rules in machine-evaluable form. There is an authorization function: a deterministic evaluation of a proposed action against that specification, producing either a proof of compliance or a rejection. There is a portable proof: a cryptographic artifact that travels with the action and that any execution system can verify independently. There is a verification requirement: execution does not proceed without a valid proof.

One clarification before naming the architecture: it closes the authorization boundary. It ensures that an action cannot execute without a proof that it was evaluated against the current governance rules and found compliant. It does not close the boundary between authorized action and real-world outcome. A governance specification can be wrong. An intent can be constructed to satisfy the formal rules while violating their purpose. The physical execution can fail or cause harm despite procedural correctness. Those limits are real, and they are addressed later in this paper. The authorization boundary and the outcome boundary are different. Closing the first is both achievable and valuable independent of whether the second can ever be fully closed.

SECTION 5

The Machine-Readable Constitution

The architecture derived in the previous section has a name: cryptographic authority infrastructure. The governance specification at its core has a name as well: an organizational constitution in machine-readable form.

The constitutional analogy is precise rather than rhetorical. A constitution is a foundational document defining the scope of authority, the conditions under which actions are permitted, and the processes by which the rules themselves can be changed. It is intended to be the highest governance artifact: the document against which all other governance decisions are measured. The critical word is document. Constitutions, in every political and organizational tradition, are human-language artifacts. They are interpreted by courts, implemented by bureaucracies, amended through political processes, and enforced by people who may or may not enforce them consistently, completely, or at all. The rule lives in the document. The enforcement lives in the institution.

A machine-readable organizational constitution works differently in kind. It is a formal specification of the same governance content: roles, permissions, approval thresholds, operational constraints, the conditions under which each

of these changes, and the processes through which the specification itself can be amended. It is a computable artifact that an authorization function evaluates against proposed actions in real time. The specification and the enforcement mechanism are the same thing.

The authorization function is a deterministic mapping. Given a proposed action (signed by the actor, specifying type, parameters, timestamp, and governance reference) and the current canonical governance state (the committed version of the organizational constitution as of this moment), the function produces either a signed authorization artifact or a rejection. Same inputs, same output, every time. No discretion available to whoever runs the function. The function evaluates the formal specification against the formal intent. The result is a proof, not a judgment.

The authorization artifact carries that proof. It contains a threshold signature over a deterministic hash of the complete intent combined with the governance state commitment at evaluation time. If any parameter of the intent is altered after evaluation, the hash changes and the signature becomes invalid. If the governance specification is updated after evaluation, the state commitment changes and the signature becomes invalid. The artifact is mathematically locked to exactly one action evaluated under exactly one version of the rules. It cannot be reused. It cannot be transferred. It cannot be forged.

The execution system that receives this artifact does not need to know the organization's governance rules. It needs to know how to verify a cryptographic signature, a standard capability present in any security-conscious execution system. The governance layer and the execution layer are architecturally separated. The proof travels with the action. The execution system verifies and acts, or verifies and rejects, without querying the governance system directly.

The five-stage authorization pipeline:

What does it mean for a rule to be encoded in this specification? It means the rule has been expressed as a computable predicate: a formal statement that a function can evaluate as true or false given a specific set of inputs.

Some rules translate directly. A spending limit is a numerical comparison. An approval threshold is a count of valid signatures from a defined set of roles. A rate limit for automated actions is a frequency check against a time window. A prohibition on certain categories of action is a membership test against a defined set. An operational state restriction is a lookup against the current state register. These are ordinary conditional logic, the same kind that runs in every piece of software that has ever checked whether a user has permission to perform an action.

Other rules do not translate directly. Rules that require weighing competing values, interpreting contextual facts, or exercising judgment about novel situations exist outside the domain of formal specification with current techniques. A machine-readable organizational constitution will always be a partial encoding of what an organization wants to govern. The question of whether that partial encoding is valuable depends on where governance failures actually occur. The formally specifiable zone covers the zone in which most failures happen. The residual zone of judgment-dependent rules is handled differently: the organizational constitution can specify who is authorized to exercise judgment in that zone and what threshold of approval that judgment requires. The judgment remains human. The record of the judgment, and the proof that the required approvers actually participated, is mathematical.

This is a pre-execution gate, native to the execution boundary. An action without a valid artifact does not execute. The architecture enforces this, not a policy.

OrgForge is an early working instantiation of this architecture: the first serious attempt to build the authorization primitive in deployable form. The category it represents is cryptographic authority infrastructure. That is what this paper is arguing for. OrgForge is the evidence that the category is not theoretical.

SECTION 6

From Firms to Governments

The machine-readable organizational constitution has no inherent domain. The authorization function evaluates a proposed action against a governance specification and produces a proof or a rejection. What varies across applications is the content of the specification. The function itself is the same.

This domain-neutrality is the architectural property that makes the progression from firms to governments something more than an analogy. The structural gap between rules and their enforcement at execution is the same gap at every level of institutional organization. The architectural response does not change when the institution grows larger or acquires public authority.

Firms

The organizational use case is the one Section 1 established: closing the authorization gap in operations where software agents, automated systems, and delegated human actors take actions at speeds and scales that outpace existing governance processes. Spending controls, deployment approvals, API access under conditional governance states, AI agent operational constraints, role-based thresholds that change with organizational context. The governance specification encodes what the organization has decided its actors are permitted to do. The authorization function makes those decisions enforceable at the execution boundary rather than auditable afterward.

Procurement and public contracting

Procurement is where organizational governance first intersects consistently with public authority. The rules governing public procurement are among the most detailed and extensively documented governance specifications in existence: competitive bidding requirements, conflict of interest provisions, socioeconomic set-aside programs, performance conditions, anti-kickback statutes. They are also among the most frequently violated, not because they are ambiguous but because no execution system is required to verify compliance at the moment a contract is awarded or a payment is made. The authorization gap in public procurement is structural and well-documented. The rules exist. The enforcement is retrospective.

Municipal governance

A city government has budgets, approval hierarchies, procurement requirements, and operational constraints that govern every significant action it takes. Most of these exist in ordinances, administrative codes, and policy documents. None of them are mechanically enforced at execution. A public works payment can be processed without any system verifying that the expenditure was authorized under the current budget allocation, that the approval threshold was met, or that the project was not in a procurement freeze. The authorization gap at the municipal level is structurally identical to the authorization gap at the organizational level. The governance specification is more complex and the public accountability obligations more extensive. The architecture that closes the gap is the same.

State and federal agencies

The federal case receives a full treatment in the following section. The point to carry forward here is that the federal authorization gap is not a special case requiring a different architecture. It is an instance of the same pattern that appears at every level of institutional organization, where the conditions attached to authorized actions live in legal text and administrative guidance rather than in the execution systems through which actions are carried out.

The Estonia precedent

Estonia rebuilt its public administration infrastructure from near-zero following the restoration of independence in 1991. The X-Road data exchange platform, the digital identity system, the e-Residency program, the online voting infrastructure: within two decades, Estonia developed what became a reference implementation of digital governance infrastructure, studied by governments on every continent.[8]

The standard framing treats this as a story about political will and national character. The structural explanation is more instructive. Estonia adopted digital governance infrastructure because it had no legacy systems worth protecting. The cost of the incumbent was zero. Every institution that could have resisted change in defense of its existing architecture either did not exist or had not yet accumulated the organizational momentum to resist. Estonia was exceptional in its lack of encumbrance, not in its ambitions.

Infrastructure of this kind does not penetrate through the political center of established institutions. It penetrates through the edges: jurisdictions rebuilding after governance failures, agencies standing up new programs without legacy constraints, contractors operating under requirements that create adoption incentives. The nations and jurisdictions most likely to adopt cryptographic authority infrastructure first are those for whom the cost of the existing architecture is most visible and the cost of the new architecture is most manageable.

SECTION 7

The Federal Question

The argument that cryptographic authority infrastructure could eventually underlie federal governance requires building from the ground up, in small pieces, before the larger claims are attempted. The place to start is not constitutions. It is payments.

The proof of concept: appropriations execution

The Anti-Deficiency Act, codified in its modern form in 1906, prohibits federal agencies from spending money in excess of amounts appropriated by Congress, or for purposes other than those specified in the appropriation.[9] The law is unambiguous. The compliance record is not.

The GAO's annual compilation of reported Anti-Deficiency Act violations documents the structural persistence of this gap. The Department of Agriculture's Commodity Credit Corporation Fund carries a reported violation of $23,840,132, appearing in federal reporting to GAO as of FY 2024; the underlying events ran from March 23 to May 17, 2022, when CCC exhausted its initial apportionment of $18,994,570 and continued obligating interest expenses before a new apportionment was approved by OMB on May 20, 2022. The agency's own post-incident review identified the cause as structural: its software could not verify remaining interest expense apportionment in real time, requiring a manual monitoring process that failed when interest rates fluctuated and disbursements moved faster than the oversight process could track.[10]

The same GAO compilation includes a Navy violation involving the Marine Corps National Defense Cadet Corps, in which uniforms, personal costs associated with uniforms, and expenses including food and travel were paid from funds that did not authorize those purposes. Amount reported: $5,655,891.07. The underlying violation dates span FY 2011 through 2018, reported to GAO on October 4, 2023 and appearing in the FY 2024 compilation.[11]

These violations are not administrative errors waiting to be fixed by better training. They are the predictable product of a governance architecture in which the rules governing public money live in statutory text and administrative guidance while the systems actually moving the money operate on access credentials and account balances. The payment systems do not check authorization conditions. The IG finds the violations afterward.

What is at stake is not simply lost money. It is democratic intent. When Congress appropriates funds for a specific purpose under specific conditions, the appropriation is the expression of a political decision about how public

resources should be used. When the conditions of that appropriation are violated because no execution system was required to check them, the democratic decision is overridden not by any deliberate political act but by the structural indifference of payment infrastructure to the rules attached to the money flowing through it.

An authorization layer encoding appropriations conditions would not require rewriting the Anti-Deficiency Act or rebuilding Treasury systems. It would require that payment systems verify, before processing a disbursement, a cryptographic proof that the transaction satisfies the applicable appropriations conditions at the moment of execution. The governance specification encodes what Congress appropriated and under what conditions. The authorization function makes those conditions mechanically real at the payment boundary. The IG retains its role, since governance specifications can be wrong and amendment processes matter. But baseline enforcement moves from retrospective audit finding to pre-execution verification. In this domain, the OIG shifts from detecting violations to auditing governance specifications. That is a considerably more useful function.

The hybrid adoption path

The path from current federal systems to cryptographic authority infrastructure does not require a legislative mandate or an agency-wide procurement. It requires what TLS required when it moved from a niche security protocol to universal internet infrastructure: a verification layer at the execution boundary that existing systems could adopt incrementally.[12]

That analogy belongs at the adoption-pattern level. TLS is simpler and far more uniform than governance verification. The instructive parallel is the penetration mechanism: TLS did not require rebuilding web servers. It required that traffic pass through a cryptographic layer before transmission. Applications adopted it incrementally, driven first by security-sensitive domains, then by browser policy, then by universal expectation. Infrastructure penetrates through the execution boundary, through demonstrated value in bounded domains, expanding as that value becomes visible.

Federal contractor systems are the natural entry point. Contractors operating under government authority already carry a compliance culture and

audit infrastructure that creates receptive conditions. They report to contracting officer representatives, maintain compliance documentation, and operate under Federal Acquisition Regulation clauses specifying authorization conditions in detail. An authorization infrastructure requirement for federal contractors, requiring proof at the execution boundary that each disbursement satisfies the applicable contract terms and appropriations conditions, is bounded, high-value, and consistent with existing compliance obligations. The contractor layer demonstrates the architecture before agencies are required to adopt it.

Grant disbursement follows for structural reasons. Federal grants carry allowability, allocability, and reasonableness conditions under the Uniform Guidance.[13] Those conditions are currently enforced through single-audit requirements and OIG reviews. An authorization layer making condition compliance a precondition of disbursement is a direct upgrade with a clear institutional constituency: awarding agencies that currently bear reputational and fiscal cost when grant dollars are misspent.

Cross-agency delegation chains are next. When federal authority moves from a lead agency to a regional office to a contractor to a subcontractor, the authorization conditions attached to each delegation level are maintained through contract language and compliance reporting. Every link in that chain is a point where authority can be exercised outside its intended scope, detectable only in retrospect. Cryptographic authorization makes the conditions at each link verifiable at the execution boundary.

The broader federal substrate follows this path over time, through demonstrated value at the edges and accumulated proof-of-concept in bounded domains.

Sovereign credibility and the architecture of state capacity

There is a version of this argument that concerns operational efficiency: agencies reduce audit findings, contractors demonstrate compliance, grant programs recover funds more reliably. That version is real and worth having. It is not the deeper version.

In 1689, following the Glorious Revolution, the British Parliament asserted control over royal spending. The Crown could no longer raise or spend money without parliamentary approval. Douglass North and Barry Weingast, in their analysis of this transition, showed that this institutional constraint did something economically transformative: it made the sovereign's financial commitments credible to capital markets in a way they had never been before.[14] Interest rates on government debt fell. Borrowing capacity expanded. The fiscal foundation of British imperial power was built not on military strength alone but on the demonstrated inability of the Crown to unilaterally violate its own financial rules. The state gained power by accepting verifiable constraint.

That is the model. The 21st-century equivalent is cryptographic verification of machine-speed execution.

The distinction worth drawing is between asserted sovereignty and verifiable sovereignty. Asserted sovereignty is the traditional model: rules on paper, compliance reported through self-assessment, violations discovered through retrospective audit when discovered at all. Verifiable sovereignty is the mechanical governance of the execution boundary, a government's demonstrated ability to prove that every action taken under its authority satisfied the conditions of the law at the moment of execution. A government that can offer verifiable compliance records to its legislature, to international partners, to capital markets, and to its own citizens is making a different kind of commitment than a government that can only assert it. The commitment's credibility comes from the architecture, not from the promises of whoever currently holds power.

The argument extends further when the focus shifts from human bureaucrats to autonomous systems operating under governmental authority. Federal agencies are increasingly dependent on AI systems, algorithmic decision support, and automated workflows for functions ranging from procurement and benefits administration to defense logistics and financial oversight. A state that cannot mechanically govern the authority it has delegated to these systems has, in a meaningful sense, lost effective control over a growing portion of its own operations. The governance lag between what the state has authorized and what its systems are actually doing expands

as those systems become faster, more autonomous, and more deeply embedded in consequential workflows. Cryptographic authority infrastructure is how a state maintains genuine sovereignty over its own operational layer as that layer becomes less human and more algorithmic.

The geopolitical implications follow from this analysis. States with stronger authority verification will have stronger procurement integrity, cleaner aid disbursement, more credible sanctions and export-control enforcement, more trustworthy treaty-linked administration, and more legible chains of delegated authority. In alliance systems, these properties matter because partners need to trust that a government can actually control what it says it controls. In competitive interstate environments, a state that can prove its internal governance integrity has a structural credibility advantage over a state that can only assert it. The adoption incentive is competitive as much as it is administrative.

Dominant powers with deep investments in existing governance architectures have strong incentives to resist changes that make their current arrangements look inadequate. Early movers will likely be states at the edges: rebuilding after governance crises, standing up new agencies without legacy constraints, operating under multilateral financing conditions that require verifiable compliance rather than self-reported assurances. The path from those entry points to broader federal and interstate adoption follows the same logic as every prior infrastructure transition, beginning with bounded adoption where the benefit is immediate and the political resistance is manageable, then expanding as demonstrated advantage creates adoption pressure elsewhere.

> *A government that can offer verifiable compliance records to its legislature, to international partners, to capital markets, and to its own citizens is making a different kind of commitment than a government that can only assert it. The commitment's credibility comes from the architecture, not from the promises of whoever currently holds power.*

The treaty case

International treaty obligations are among the hardest governance problems that exist, governed by some of the weakest enforcement architecture in operation. Nations make commitments covering non-proliferation, climate, trade, and human rights, and compliance is verified through self-reporting, periodic inspections, diplomatic pressure, and referral to international bodies with limited enforcement power.[15] The mechanisms are slow, politically freighted, and frequently ineffective.

Nuclear procurement is where this argument is most concrete. Non-proliferation commitments include restrictions on acquiring specific materials, technologies, and equipment. In principle, those restrictions bind the procurement systems of signatory states. In practice, enforcement depends on the IAEA's safeguards program: a system built on access, cooperation, and the good faith of the states being monitored.[16] A trusted party reviews the record and issues a determination. The determination is only as reliable as the record the state provides.

Consider what changes if two signatory states operate procurement systems with authorization infrastructure. Procurement transactions touching restricted categories generate cryptographic proofs that each transaction was evaluated against the treaty-specified conditions at the moment it executed. Those proofs are verifiable by any treaty party without requiring access to the other state's internal records or cooperation from the state being verified.

The scope of that claim requires precision. Cryptographic authorization makes one specific class of compliance more legible and more independently verifiable: procedural authorization becomes auditable without institutional intermediaries. A state could operate systems outside the authorization infrastructure. A governance specification could be written to exclude restricted categories from authorization requirements. The architecture addresses the verifiability of what was authorized through the system, not the full scope of what a state does in the world.

Within that scope, the difference between asserted compliance and independently verifiable compliance is consequential. Current treaty verification asks whether the records a state has provided can be trusted.

Authorization infrastructure shifts part of that question to whether the cryptographic record of what the state's procurement systems authorized matches the treaty conditions. Those are different questions, and the second carries evidence that does not depend on the state's cooperation to produce. Institutions, diplomacy, and political will remain essential. The architecture offers one class of compliance claim that is harder to fabricate than a self-reported record.

SECTION 8

The Hard Problems

A paper that argues for a new category of governance infrastructure and does not engage seriously with the ways it can fail is advocacy, not analysis.

Discretion is a feature, not only a vulnerability

The case for cryptographic authority infrastructure rests in part on the observation that discretion is the mechanism through which governance fails: officials who exercise discretion in their own interest, agents who act outside their authorized scope, systems that execute actions no one intended to authorize. This framing is accurate. It is also incomplete.

Discretion is also the mechanism through which governance succeeds in novel situations. A rigid system that cannot respond to circumstances its designers did not anticipate is brittle rather than well-governed. The history of law is full of cases where the literal application of a rule produced outcomes so contrary to the rule's purpose that courts, regulators, or legislatures had to intervene. The ability to exercise judgment, to consider context, to recognize when a rule is failing to serve its intended function: these are necessary features of any governance system operating in a complex and changing world.

Cryptographic authority infrastructure does not eliminate discretion. It encodes the conditions under which discretion is exercised and makes that exercise verifiable. When an organizational constitution requires multi-party approval for a decision involving judgment, the authorization function

enforces that requirement. The approvers apply their judgment. The architecture produces a proof that the required approvers actually participated, with the requisite authority, at the time of the decision. The judgment is human. The record is mathematical.

Emergency provisions work the same way. An organizational constitution can include operational states that suspend normal authorization requirements under specified conditions, subject to specified approvals. A freeze mode, an incident response state, an emergency override: these are formal specifications of the conditions under which governance temporarily operates differently. The discretion available under those states is broader. The authorization for entering those states is itself subject to the architecture. Declaring an emergency to escape governance constraints requires a valid authorization for the declaration. This is authorized discretion, governed by the same system as everything else, with a cryptographic record of when it was invoked and by whom.

The boundary between procedural correctness and material truth

A cryptographically valid authorization artifact means that a proposed action was evaluated against the current governance specification and found formally compliant. It does not mean the action's real-world consequences will be good, intended, or safe.

The Oracle Problem identifies the specific vulnerability here. A governance specification can enforce a rule based on an external variable (a market price, an interest rate, a reported inventory level) but it cannot independently verify the accuracy of that variable if it originates outside the system. If an AI agent is authorized to execute a trade provided market volatility falls below a threshold, and the data source feeding that threshold provides a corrupted or hallucinated value, the architecture will produce a mathematically valid authorization for a materially catastrophic action. The system stops unauthorized actions. It does not stop authorized mistakes.

This is a genuine limit, and it applies to any governance system, including the human-intermediary systems that currently exist. What the architecture provides is an immutable, independently verifiable record of which rule permitted the action, making post-failure accountability cleaner than any

retrospective reconstruction.

Bad rules enforced well

The authorization function enforces whatever is in the governance specification. It enforces unjust rules with the same fidelity it enforces just ones. The architecture is entirely neutral toward the content of what it enforces.

This is a genuine limit. Cryptographic authority infrastructure does not solve the problem of bad governance. It assumes the problem of what rules to write has been addressed elsewhere, and confines itself to enforcing whatever rules exist consistently and verifiably at the execution boundary.

Consistent and verifiable enforcement is nonetheless a precondition for accountability and amendment, regardless of whether the rules being enforced are good ones. You cannot challenge a rule that no one can find. You cannot build a case for changing a rule without a record of how it has been applied. You cannot hold an institution accountable for violating its own rules without evidence that the rules were violated rather than selectively enforced. The architecture provides that foundation. What is built on top of it is a political and social question that the architecture does not answer.

The authoritarian adoption paradox

Authoritarian governance systems benefit from opacity and selective enforcement. The ability to apply rules inconsistently, to create exceptions for allies and enforce strictly against opponents, to exercise discretion in ways that concentrate power: these are primary mechanisms of control, not incidental features.

Cryptographic authority infrastructure, adopted fully, would constrain exactly these mechanisms. Consistent enforcement, publicly verifiable authorization records, governance specifications that apply equally regardless of who submits the intent: these properties are structurally incompatible with authoritarian control.

This creates a predictable adoption pattern. Authoritarian governments will be drawn to the efficiency properties of the architecture (reduced petty

corruption, lower bureaucratic friction, credibility signals to international investors and partners who want verifiable compliance records) while resisting the accountability properties: equal application of rules, publicly verifiable enforcement, governance specifications that cannot be silently modified by whoever holds power at the moment.

The predictable response is selective deployment: authorization infrastructure for workflows that face external scrutiny, while maintaining discretionary domestic enforcement through systems outside the architecture. The infrastructure is adopted where it provides credibility at the lowest political cost. It is not adopted where it would constrain the exercise of power. This is an accurate prediction about how institutions that benefit from the current system will respond to infrastructure that threatens it. The architecture does not force adoption. It offers a verifiable alternative to institutional trust.

SECTION 9

The Decentralization Condition

There is a version of everything this paper describes that is a useful product. There is a version of it that is infrastructure. The difference is a single architectural property.

When a single operator runs the authorization function, the organizations relying on it are trusting that operator. The governance layer is cryptographically structured. The audit trail is complete and verifiable. The enforcement is pre-execution rather than retrospective. These are real improvements over the status quo. They are improvements delivered by a vendor, dependent on that vendor's continued honesty and competence, subject to the same failure modes that affect any trusted intermediary. The vendor can be captured. The vendor can be compelled. The vendor can fail commercially.

One signer is a service. The credibility of the authorization proof is borrowed from the credibility of the institution issuing it. The proof is only as reliable as the issuer.

> *One signer is a service. The credibility of the authorization proof is borrowed from the credibility of the institution issuing it. The proof is only as reliable as the issuer. The infrastructure threshold is crossed at threshold signing.*

In a multi-validator architecture, a valid authorization artifact requires a quorum of independent validators to agree on the authorization decision before the artifact is produced. A 2-of-3 threshold means any single validator, whether compromised, pressured, or simply wrong, cannot produce a fraudulent authorization on its own. A 3-of-5 threshold means an attacker must compromise a majority of independent validators simultaneously.[17] At meaningful quorum sizes with genuine validator independence, this becomes a structural security property rather than a policy commitment. The authorization is reliable because the architecture makes producing a fraudulent proof mathematically prohibitive, not because any particular operator can be trusted.

At threshold signing with genuine validator independence, the authorization layer becomes neutral infrastructure. Any organization publishes its governance specification to a canonical registry. Any qualifying validator set can evaluate actions against it. Any execution system can verify the resulting artifacts. No single party, including the network's founders, controls what gets authorized.

The practical question of why execution systems would actually require a valid artifact, and why independent actors would participate in producing them, has a concrete answer that does not depend on speculation about token economics.

Execution systems require artifacts when the surrounding institutional environment makes that requirement rational. SEC Rule 15c3-5, the Market Access Rule, already codifies requirements for broker-dealers to maintain risk management controls and supervisory procedures for automated market access, including pre-trade risk checks designed to prevent the transmission

of erroneous orders.[18] Requirements of this kind are the regulatory mechanism through which execution systems develop obligations to verify governance compliance before acting. Procurement terms can require valid artifacts as a condition of payment. Internal controls can require them as a condition of releasing credentials. Insurers and auditors can require them as a condition of coverage and certification. Counterparties in high-stakes transactions can require them as a condition of settlement. Treaty-linked compliance regimes can require them as a condition of participation. Each of these mechanisms, operating independently and reinforcing each other, creates the institutional environment in which execution systems have strong reasons to require artifacts rather than act without them.

The incentive for validators follows the same operational logic. Participants whose operations depend on the governance layer's credibility have structural reasons to maintain its integrity. Certificate authorities in the TLS infrastructure are compensated by the services that depend on encrypted communications. DNS operators run infrastructure because their core business depends on internet reliability. Validators in an authorization network occupy the same structural position: organizations that benefit from a trustworthy, neutral governance layer have reasons to contribute to the infrastructure providing it.

The path from hosted service to neutral infrastructure follows a deliberate sequence. The primitive is demonstrated in deployable form. Threshold signing is deployed at meaningful quorum, with validators that are genuinely independent and subject to published participation rules. The validator set is opened to external participants under conditions encoded in the governance specification and enforced by the authorization function. At that point, the governance layer governs itself. The amendment process for the validator participation rules is subject to the same authorization infrastructure as everything else.

An authorization network whose own governance is subject to the same architecture it provides to others is making a structural claim, not a trust claim. That claim can be verified by anyone with access to the governance specification and the authorization record. That is the difference between infrastructure and a well-intentioned service.

SECTION 10

The Governance Lag

Every era of organized human activity has lived with some version of the governance lag: the gap between the moment a rule is articulated and the moment an action is governed by it. In small, slow-moving institutions, this gap was narrow enough to manage. A village council could enforce its decisions because the pace of action was visible and detection was sufficient. A medieval guild could discipline its members because violations were observable, social consequences were immediate, and the scale of damage a single actor could cause before being caught was limited.

The lag widened as institutions scaled. A government operating across a continent could write laws that its agents in distant provinces chose to ignore, with no practical mechanism for the center to know. A corporation operating across multiple business lines could issue policies that different divisions implemented differently, with no system for the parent to verify which version was actually in use. Auditors were invented to close this gap retrospectively. Oversight bodies were invented to close it prospectively, through deterrence. Neither closed the gap. Both reduced it, at substantial cost, in some contexts and not others.

The digital transformation of organizational operation widened the governance lag at the execution boundary while creating the appearance of control elsewhere. Systems generate logs. Logs create the impression that everything is monitored. The impression is accurate in a narrow sense: what happened is recorded. Whether what happened satisfied the organization's rules at the moment it happened remains answered retrospectively, if at all.

Artificial intelligence does not introduce a new version of the governance lag. It changes the relationship between the lag and its consequences.

In previous eras, the lag was bounded by the pace of human action. Detection, however imperfect, arrived within a timeframe that sometimes

allowed recovery. The systemic assumption embedded in every governance architecture built over the past several centuries is that human-speed action and human-speed detection are roughly compatible. Governance systems were not designed to be perfect. They were designed to be good enough given the pace at which things happened.

That assumption fails in a specific class of cases: any operation in which software agents with operational authority act at machine speed. The Flash Crash erased nearly a trillion dollars in twenty minutes before human reviewers could intervene. Knight Capital lost $460 million in forty-five minutes because a single server was running the wrong code. In these cases, consequences compound beyond recovery before any governance process responds. The 2010 and 2012 incidents were warnings. They will not be the last.

The structural response to this condition is governance native to the execution boundary: every action carries a proof that it was evaluated against the current governance specification and found compliant, and the architecture refuses execution without that proof.

The adoption of this infrastructure will not happen uniformly or quickly. Institutions with deep investments in existing governance architectures have powerful reasons to resist changes that make their current arrangements look inadequate. The organizations most likely to adopt first are those for whom the cost of the current governance lag is already legible and already painful: financial institutions managing algorithmic trading risk, federal agencies under OIG scrutiny for recurring appropriations violations, defense contractors operating under procurement compliance requirements they currently satisfy through documentation rather than verification. These are institutions for whom the problem is already acute and the cost of the current answer is already visible.

From those entry points, the architecture propagates through demonstrated advantage. Every organization that can present a verifiable governance record to counterparties, regulators, and auditors has something that organizations operating on asserted compliance cannot match. Every

jurisdiction that makes authorization infrastructure a condition of contractor participation creates adoption pressure in its supplier ecosystem. Every international agreement that requires verifiable compliance rather than self-reported compliance creates demand for the architecture at a level where self-reporting has already been revealed as insufficient.

History gives no particular reason for optimism about the timing of this transition. Critical infrastructure is typically adopted after the failures it would have prevented, rather than before them. The three-point seat belt was designed in 1959 and demonstrated a 50 percent reduction in fatalities. It took nearly a decade of accumulating crash data and persistent regulatory pressure before it became mandatory equipment.[19] The Pure Food and Drug Act of 1906 did not arrive from a legislature seized with prophylactic vision. It arrived because Upton Sinclair's reporting on the meatpacking industry made the cost of the status quo impossible to ignore.[20] The pattern holds across infrastructure categories: the cost of the incumbent becomes legible through a sufficiently large failure, and the infrastructure that would have prevented it is adopted in response.

Systems under the pressures this paper describes (machine-speed execution, autonomous agents holding operational authority, and governance processes running at human pace) tend to adapt toward architectures that close the execution boundary. The pressure is structural. The direction of adaptation, over time and under increasing operational stress, runs toward governance that is native to execution rather than adjacent to it.

The governance lag in machine-speed operations has not yet produced a failure large enough to force institutional response at scale. The incidents that have occurred have been contained, absorbed, or attributed to specific bad actors rather than to the structural gap that made them possible.

What is different about this moment is that the architecture exists before the forcing failure rather than in response to it. The question is whether the institutions whose governance depends on closing the execution gap act on that structural reality while acting is still a choice.

> *The architecture exists before the forcing failure rather than in response to it. The question is whether the institutions whose governance depends on closing the execution gap act on that structural reality while acting is still a choice.*

NOTES

[1] The Flash Crash of May 6, 2010 is documented in the joint CFTC-SEC report: Findings Regarding the Market Events of May 6, 2010, Report of the Staffs of the CFTC and SEC to the Joint Advisory Committee on Emerging Regulatory Issues (September 30, 2010). The 'hot potato' effect and the 27,000 contract-change figure appear on pp. 2-3 of that report. The initiating sell order was placed by Waddell and Reed Financial.

[2] The Knight Capital incident is documented in the SEC Administrative Proceeding, In the Matter of Knight Capital Americas LLC, File No. 3-15570 (October 16, 2013). The Power Peg designation and the eight-server deployment failure appear in the order's factual findings. The $460 million figure is from Knight Capital's August 2, 2012 press release.

[3] Jensen, M.C. and Meckling, W.H. (1976). Theory of the firm: Managerial behavior, agency costs and ownership structure. *Journal of Financial Economics*, 3(4), 305-360.

[4] North, D.C. (1990). *Institutions, Institutional Change and Economic Performance*. Cambridge University Press.

[5] Acemoglu, D. and Robinson, J.A. (2012). *Why Nations Fail: The Origins of Power, Prosperity, and Poverty*. Crown Business.

[6] Bitcoin: A peer-to-peer electronic cash system (2008). https://bitcoin.org/bitcoin.pdf

[7] For a technical treatment of the consensus mechanism and its security properties, see Narayanan, A. et al. (2016). *Bitcoin and Cryptocurrency Technologies*. Princeton University Press.

[8] For Estonia's digital infrastructure development, see Kalvet, T. (2012). Innovation: a factor explaining e-government success in Estonia. *Electronic Government*, 9(2), 142-157. See also e-estonia.com.

[9] Anti-Deficiency Act, 31 U.S.C. Sections 1341, 1342, 1349-1351, 1511-1519.

[10] GAO, Antideficiency Act: Information on Violations Reported to GAO, FY 2024. The USDA Commodity Credit Corporation Fund violation (amount reported: $23,840,132.00) reflects underlying events from March 23 to May 17, 2022, when CCC exhausted its initial FY 2022 apportionment of $18,994,570 and continued obligating interest expenses before a new apportionment was approved by OMB on May 20, 2022. The agency identified the cause as an inability to verify remaining interest expense apportionment in real time. See also USDA 2024 Agency Financial Report.

[11] GAO, Antideficiency Act: Information on Violations Reported to GAO, FY 2024. The Navy violation involving the Marine Corps National Defense Cadet Corps (amount reported: $5,655,891.07) covers improper payments for uniforms, personal costs, and other expenses including food and travel. The underlying violation dates span FY 2011-2018. The report was submitted to GAO on October 4, 2023.

[12] For TLS adoption history and the mechanisms through which it became universal, see Felt, A.P. et al. (2017). Measuring HTTPS adoption on the web. *Proceedings of the 26th USENIX Security Symposium*.

[13] Office of Management and Budget, Uniform Administrative Requirements, Cost Principles, and Audit Requirements for Federal Awards (Uniform Guidance), 2 C.F.R. Part 200.

[14] North, D.C. and Weingast, B.R. (1989). Constitutions and commitment: The evolution of institutions governing public choice in seventeenth-century England. *Journal of Economic History*, 49(4), 803-832.

[15] For a treatment of international treaty enforcement limitations, see Chayes, A. and Chayes, A.H. (1995). *The New Sovereignty: Compliance with International Regulatory Agreements*. Harvard University Press.

[16] IAEA Safeguards Overview: Comprehensive Safeguards Agreements and Additional Protocols. https://www.iaea.org/topics/safeguards-and-verification

[17] For the cryptographic foundations of threshold signature schemes, see Shoup, V. (2000). Practical threshold signatures. *Advances in Cryptology, EUROCRYPT 2000*, 207-220.

[18] SEC Rule 15c3-5, Market Access Rule, 17 C.F.R. Section 240.15c3-5 (2010). The rule requires broker-dealers with market access to implement risk management controls including pre-trade risk checks and supervisory procedures reasonably designed to prevent erroneous orders.

[19] For the history of seat belt regulation and the gap between technical availability and mandatory adoption, see Eastman, J.W. (1984). *Styling vs. Safety: The American Automobile Industry and the Development of Automotive Safety, 1900-1966*. University Press of America. The Nils Bohlin design patent was filed in 1959; mandatory federal installation requirements came into effect in 1968.

[20] Sinclair, U. (1906). *The Jungle*. Doubleday, Page and Company. The Pure Food and Drug Act was signed June 30, 1906.

Gary Chigaros is the founder of OrgForge, Inc. contact@orgforge.io | orgforge.io