

CODE AS LAW

0.0.1

A Brief Introduction to Coding Contracts

Table of Contents

Foreword

Chapter 1: *Contracts*

Chapter 2: *Accounts*

Chapter 3: *Owners*

Chapter 4: *Auth*

Chapter 5: *Lex*

Coda

Foreword

This book shows where code works as law under classical contracts. At the time of writing, it finds itself at an interesting inflection point for those in the so-called *Code is Law* camp.¹ Just last week, a French criminal court acquitted two defendants accused of exploiting a decentralized finance (DeFi) protocol, *Platypus*, for millions of dollars on account of the case more simply involving drafting errors² and matters of civil and contract interpretation rather than the more dubious accusations of criminal theft.³

The goal of this short book therefore is simple, especially now in light of the obvious collision course of blockchains and the law: Show how the policy goals of the legal tradition and commercial customs can be met by this kind of code. Through examples and explanations. This exercise should be familiar to law students and coders alike. Specifically, by highlighting how smart contracts and familiar principles of equity work together under the settlement security of blockchain networks, governing billions of dollars, we can fairly point out where code already *is*, or really *should be*, the prevailing law among participants.

For lawyers, the private law of agreements should be an exciting space for revamping ownership—where participants can now make their own laws with real force, subject to the intuitive restraints of public policy (see further, *Contracts*). To stretch the banks of English, an *à la carte omakase* version of contract law is not only manifest destiny based on its own historical arc, but convenient to the risks of our times, where software and digital assets can represent critical commitments and business logic that need native, unambiguous security guarantees. Overall, this transition of agreements and law to code speaks to the values of the internet and the classical liberal tradition of embracing freedom of association and contract. So, let's strap in and seize the task.

At the end of the day, regardless of the situation or debates about justice norms in handling disputes, the contracts themselves should “just work” and escape the

¹ “Code Is Law,” Lawrence Lessig, <https://www.harvardmagazine.com/2000/01/code-is-law-html>, “The law of cyberspace will be how cyberspace codes it[.]”; “Unregulability is a function of code, but the code can change.”

² *Cf.* the legal doctrine of *contra proferentem* (latin, “against the drafter”) in which judicial interpretation of contracts places risks of contract drafting errors and misunderstandings over terms against the offering party since they are often in the best position to avoid risks and harms related to mistakes in the offer itself.

³ “Code is Law Question Raised Again as French Court Rules ‘Yes’” *The Defiant*, <https://thedefiant.io/code-is-law-question-raised-again-as-french-court-rules-yes>.

more common footfaults of error, lazy neglect and principal-agent problems. Blockchains currently offer the best technology we have for this.⁴

I therefore welcome you on this brief journey and hope you can start coding smart contracts yourself working from these iterative concepts, as well, defend their primary role in expanding market choice in settlement providers. Choice after all is good. Even if blockchain adoption stopped today, a segment of the economy will always embrace the tools that work natively with the things they own. We can safely expect more people to own more things on the internet and for this reason alone, I encourage you to read further.

⁴ "He who will not apply new remedies must expect new evils," Francis Bacon, *Of Innovations*, THE ESSAYS OF FRANCIS BACON 59 (1920).

Chapter 1: Contracts

Is a smart contract a contract-at-law? Before going into the rote elements of legal agreement formation, namely, those of *Offer*, *Acceptance* and *Consideration*, let us briefly consider the origin of the form itself: *Promises*, and why contract law came into being as a tool to manage expectations and extend public governance to private affairs. In short, because many smart contracts imply special duties of care and manage material amounts of value, they should be considered legally significant artifacts. But first, let's review the basic philosophy for what kinds of promises should be enforced in the first place, regardless of whether they are expressed by code or on the back of napkins. (Do you for example actually know why you are punished for breaking certain commitments?)

Promises imply *Duty*. This should occur easily enough to most people. As it speaks to fundamental fairness, and in our formulation of natural legal standards that should apply to code, it is a reasonable expectation that most people have entered social contracts as citizens enforced by a governance system that prioritizes their fair treatment. After all, history easily teaches us that, no fairness means no workable contracts, as people will refuse to coordinate and honor rules that they cannot trust to work in their interest or follow the terms of their underlying social contracts.

Turning back to *Duty* and its primordial role in instructing judicial instincts on legal agreements, it is expected that if someone relies on your promise, they might come to harm if you fail to hold up your end of the bargain. For example, if you tell someone you will pick them up from the airport for a sum of money (*i.e.*, Uber), they will be harmed if you don't show up. Since we can all likely agree that good governance should act to prevent and remedy harms, we can logically see how it should therefore enforce duties of care that are created either publicly, under the concept of social contracts for things like crime, negligence and other torts, or privately, through legal agreements that give rise to opt-in special duties to prevent their particular harms-induced-by-reliance. After all, we all rely on each other to not put slippery oil onto sidewalks. We do not all rely on each other to deliver oil, but reasonably would, it was promised in exchange for something of value (consideration) and not merely a gift or casual suggestion (*i.e.*, an illusory promise or agreement-to-agree).

This overarching context of the social contract and fairness in promises is important for the meaningful discussion of blockchain-based agreements. Before we even begin entertaining the notion that *Code is Law* in a primarily political sense, where it might seem illegitimate to restrict freedom of contract or redistribute property rights under principles-of-equity, it must be assumed there will be exceptions and factions that arise on the spectrum of *How Much Code is Law*.

As *The DAO*⁵ hardfork on the Ethereum blockchain (and attempts to invoke its reasoning through the *Parity Lobby*) easily demonstrated, there will be recurring disputes over economically-significant data entries (or, *who owns what*) through fork-choice on the valid state of the blockchain, much like legacy governance has entertained secession and civil war as legitimate ways to sever relationships. Overall, these public governance decisions form a sort of case law and arbitral jurisprudence, accelerating the resolution of property by letting the market decide their history and the platonic state of promises-made-and-promises-kept without requiring formal legal interventions. (*This is The Way*, and offers much greater legal access and affordability around enforcement.)⁶

In terms of bare legal utility (*Code is Law* is not the object of this book, but an extension of it), blockchains provide a credibly neutral medium to record and enforce the terms of their promises (agreements).⁷ In many cases, they can allow people to incorporate code and performance programs to meet their obligations into the records of the agreements themselves. This uniquely sets them apart in two ways from existing legal infrastructure.

First, blockchains have essentially commoditized arbitration of data through provable economic incentives to maintain accurate logs among peers (or *state changes*). The essence of this game is easy enough: Pay to update the ledger, and pay more if you try to corrupt it. As we will discuss further (see *Lex*), this raw technical feature secures an agreeable source of truth and settlement authority that resolution systems (courts) usually provide, but without the high friction and opportunity costs which can make their search for truth more like the *search-for-the-deepest-pocket*. Second, by supporting functional programming, blockchains like Ethereum have enabled many peers to incorporate the essential terms of property rights, such as the ownership right to exclude and assign, directly into their agreements as machine-readable bytecode.

In this new medium and suggestive choice-of-law, if the terms or property changes implicated by an agreement can be digitized as tokens (themselves ruled by smart contract source code), the entire operation of the agreement can be conducted online and settled through related smart contract state changes. Contract drafting in this mode is the task of the smart contract coder, contract administration, the role of the smart contract, and resolution and enforcement of terms, the work of the blockchain validators. This is really the *smart* part of smart contracts, in that they resemble other automated digital processes, such as computer programs, but in this

⁵ "The DAO," Securities and Exchange Commission, *Release No. 81207 / July 25, 2017*, <https://www.sec.gov/files/litigation/investreport/34-81207.pdf>.

⁶ "The Unlawed," Anthony D'Onofrio, <https://distributed-autonomous-society.quora.com/The-Unlawed>.

⁷ "Credible Neutrality As A Guiding Principle," Vitalik Buterin, *Nakamoto*, <https://nakamoto.com/credible-neutrality/>.

case, the computers process things that have real economic value and are not merely instructions to trust agents, like banks, to make such updates and observe parties' performance.

The problem of involving others in a bargain is also self-evident and the practical appeal of smart contracts speaks to solving the core agency problems in agreements: The more people who have to make decisions and respond to the occurrence or non-occurrence of events to honor and enforce terms, the more likely an issue or censorship will arise.

Censorship-resistance, to be certain, is not just an important concern for political organizations but essential for high-stakes business logic. Apple would not want Microsoft to handle its finances, and for similar reasons, especially as markets and software increasingly merge, many legal consumers and corporate departments would not want their competitors or reactionary governance to halt their ability to operate and perform on their commitments.

So, in light of these intuitive commercial considerations, it seems only right and fair that when we enter into a promise that has a degree of seriousness that overarching social contracts would expect them to be enforced: We realize a legal agreement has been made.

This normative ideal around enforceable promises or agreements is more important to understand than the basic elements of classical contracts, but for sake of completeness, we can demonstrate shortly how smart contracts can have similar legal effect and intention. As we have shown, smart contracts do not escape special duties of care, and even more so, can be hard records of classical contracts as well.

For example, a smart contract can be said to contain a legally-binding *Offer* in cases where it invites the public to deposit their funds in exchange for digital tokens, such as in the case of a decentralized exchange or initial coin offering (ICO). *Acceptance* of such an *Offer* can also be seen in transactions signed with private keys owned by those with the capacity to enter into a legal agreement. *Consideration* can be specified by the smart contract as digital tokens or merely through the act of publishing the interaction transaction itself, as blockchain publication fees, like a peppercorn,⁸ can be seen as a sufficient detriment to invoke reasonable reliance.

So, if blockchain consumers, like public addresses that post and exchange value under smart contracts, can be said to be entering into legal agreements, who are they entering them into with? The code itself or legally-cognizable entities?

⁸ See, "The Peppercorn Theory and the Restatement of Contracts," Edmund Polubinski Jr., *William & Mary Law Review*, <https://scholarship.law.wm.edu/cgi/viewcontent.cgi?article=2867&context=wmlr>, which explains the trend in judicial treatment of issues like adequacy of consideration to form legally-binding agreements as deferential to the bargain struck by the parties, such as that in the case of certain dealings, a mere "peppercorn" should suffice.

For the moment, international trends in agency law would likely associate and imply relations among consumers posting transactions and those who manage a smart contract's state and receive such communications—such as by deploying the blockchain bytecode to initialize their *Offer* and terms of *Acceptance*, or by controlling upgrades and state through related maintenance tasks like updating software and fixing bugs. These may be individuals, incorporated entities, or, as more often is the case, general partnerships and unincorporated associations derived from organizational tokens and related smart contracts (DAOs).

And so yes, smart contracts can be legal agreements and create legal associations. Well-designed smart contracts with small attack surfaces operate much like vending machines and present the most optimizations over past forms, giving a stated digital output predictably for a digital input. The risk here is minimized to the contract itself (contract risk) and the network hosting its data (chain risk).

Hybrid arrangements with human agency may leverage the blockchain purely to record evidence of signatures (or hashes of the human-readable version of agreements) to anchor their related duties (such as tokenized representations of real estate that still require off-chain management to perform value transfers and property title), but more advanced and complete representations will exist purely on the internet and settle performance on the blockchain (onchain). In this version, they resemble commercial receipts and clickwrap agreements which streamline the classical elements of contract formation (see above) into the UX of applications.

Under these circumstances, the legal duty of the offeror should be to have created a valid onchain contract that does not contradict their statements and marketing, as a faulty smart contract can be likened to fraud-in-the-inducement and breach of their duty of care originated from their Offer. If maintenance is involved to fulfill their promise and related duties (again, as suggested in the comments of the smart contract code or their statements around release or in offered user applications), their related duties around these tasks can be implied, similar to how regulators insist that marketing can make certain smart contract transactions invoke investment contracts that attach ongoing reporting obligations.

So, like legal agreements can be formed casually where the essential elemental intents are shown, such as orally or silently through conduct, smart contract and blockchain transactions fit familiarly into classical contracts as another medium for markets and consumers to express and enforce their preferences and establish economic relationships. While these may be disclaimed, if expectations are fairly and reasonably formed around performance and blockchain data state changes, legal duties of care should apply and give rise to claims to perform or repair damages.

Altogether, these core theories of contract law will support our journey into other aspects of law and cryptography (Crypto Law), as we move now from settling the novel (but admittedly mundane) question of whether smart contracts are legal contracts. Next, we approach the key instruments of this internet-native medium for law and the completion of performance: *Accounts*, *Ownership*, *Auth*, and for finality, *Lex*.

For our immediate purposes, and what can be expected in no short amount of time (while imminent), is that what we call smart contracts will be more easily understood by legal systems as just contracts, with some conventional ways to interpret whether they give rise to claims not otherwise handled by their code.

Going forward, we will denote the difference between those legal contracts interpreted and enforced by legacy governance (governments) and incentives as *classical*, and those that can be enforced entirely by computers and private arbitration, as just that, *Contracts*.

With a sufficient share of the economy cryptographically ordered into systems like blockchains, we can also expect a shift in how lawyers and other legal professionals understand such Contracts, starting first with the leap of faith (but satisfaction of reason) that cryptographic value can give rise to legal claims just like any other value. So long as these systems matter enough to people to rely on and plan their lives around them.

Chapter 2: Accounts

An *Account* is technically the state associated with a blockchain address.⁹

This state consists of a *nonce*, which is the number of its last signed transaction (by an entity that has constructive possession of this state by controlling the public key address derived from its private key), a *balance*, which is the amount of the blockchain's native asset the Account has the right to pay for transaction validation or as consideration to other Accounts, as well as *code*, which is a static array of data that other Accounts can interact with by sending it a transaction that follows its particular calldata format (interface) and its rules against reverting.

Account

State associated with an address.

```
class Account
  nonce:ethereum.base_types.Uint
  balance:ethereum.base_types.U256
  code:bytes
```

This is all to say in so many words, *Not Your Keys, Not Your Coins*.¹⁰

But the state of an Account as holding code, or not, is what essentially differentiates a typical user account (*Externally-Owned Account*, or EOA) and a Contract, which can have arbitrary rules around how it possesses and updates its state (*i.e.*, not just knowing the secret forming a private key).

Code also makes an Account interoperable with other Accounts and Contracts. *Note:* The offchain conditions for assembling a private key and signing a transaction to make a valid Account state change by sending assets or calling a Contract (such as seed phrases or multiparty computation (MPC)), cannot be trustlessly accessed by other Accounts in terms of the data made publicly available onchain. For example, if a user tells another that they will make a particular transaction from their EOA if the

⁹ See *Execution Specs*, <https://github.com/ethereum/execution-specs/>.

¹⁰ "Not your keys, not your wallet, as they say," Elon Musk, X, <https://x.com/elonmusk/status/1732861163271057507?s=20>.

other does something with their EOA or an offchain task, the conditions are such that classical contracts are more suited to help translate the intent of the parties (the oft-cited meeting-of-the-minds) into performance obligations both parties can trust to run their course (and more importantly, that judges can actually read) regardless of knowing the character and quality of the other. However, in the purely encoded context of a Contract, nobody has to trust or know anyone (on the internet, nobody knows you're a dog, or an AI!) with regard to expected Account state changes. They will occur exactly as programmed subject to the usual Contract and chain risk.

So, for purposes of legal effect, while an EOA more readily demonstrates the intent needed to validly make or accept an offer and grant legal consideration (after all, a person or persons or entity has to first access their key and pay a fee to make such transactions as well as prepare custom data), a Contract can receive these intents to make legally-binding duties.

It is to be expected with the proliferation of industry standards and software to abstract away holding private keys from the typical user experience of Account ownership, that Contract Accounts (or Smart Accounts) rather than EOAs will become more commonly used to both represent conditional logic and custody of funds (the stuff of agreements) as well as the actual parties to an agreement, given their flexibility, and unique modular abilities to designate and authorize certain transactions based on the occurrence or non-occurrence of events onchain (see *Auth*). More or less, *It's Contracts All the Way Down*.

Chapter 3: Owners

As we can now appreciate that Contracts themselves can be an extension of Accounts, it follows that Account ownership would be similarly handled.

In classical contracts and property law, ownership is expressed as the right to exclude others from, and enjoy, a given asset.¹¹ While often proof of such legal rights, possession alone is not enough. It must be legally defensible as to title or beneficial ownership to survive pleadings to an authority to redistribute it.

To better understand how this all may work in the context of Contracts and Accounts, let's first describe what it actually means to hold a digital asset like a token onchain.

When an Account has a token balance under a Contract interface standard like ERC20¹², this simply means that, in the virtual ledger supported by the token bytecode, there is a mapping from that Account's public key address to a numerical value that it has the right to spend. There isn't anything to read from the Account's associated data that would suggest this relationship: It is entirely handled by the external Contract and understood by its public functions and state. It therefore is a contractual arrangement, not implied or presided over by the blockchain, other than ensuring the Contract operates according to its code.

```
Examples.sol
1 // SPDX-License-Identifier: AGPL-3.0-only
2 pragma solidity ^0.8.19;
3
4 contract Token { . . .
5
6     mapping(address account => uint256 balance) public balanceOf; // Public balances.
7
8     . . . }
```

No trust agent is required to settle token transactions that are governed by a Contract. When the owner signs calldata with instructions to update their tokens, it will settle after being confirmed unless its logic and terms are violated. For example, without a sufficient balance, calls to spend and transfer tokens by an Account will just revert and be impossible to complete, following the simple rules of arithmetic. This is

¹¹ See, "The Value of the Right to Exclude: An Empirical Assessment" (2017), Klick, Jonathan and Parchomovsky, Gideon, https://scholarship.law.upenn.edu/faculty_scholarship/1637, finding that even minor incursions into private rights to exclude were viewed very negatively by property owners following the passage of the Countryside and Rights of Way Act in England and Wales in 2000 which expanded public rights to roam.

¹² "ERC-20: Token Standard," Fabian Vogelsteller & Vitalik Buterin, *Ethereum Improvement Proposals*, <https://eips.ethereum.org/EIPS/eip-20>.

effective, real-time enforcement among all Contract participants that they can all trust as open-source code and math to prevent fraud. Inserting additional context or rules should be resisted under more modern contract law, which prioritizes this kind of freedom-of-contract (*cf.*, the “four-corners rule”).¹⁵

```
Examples.sol
1 // SPDX-License-Identifier: AGPL-3.0-only
2 pragma solidity ^0.8.19;
3
4 contract Token { . . .
5     function transfer(address to, uint256 amount) public returns (bool success) {
6         balanceOf[msg.sender] -= amount; // Revert if not enough tokens.
7         balanceOf[to] += amount; // Settle transfer.
8         emit Transfer(to, amount); // Log event.
9         return true; // Return tx success.
10    }
11    . . . }
```

Contract ownership also resembles *fee simple*, in that it grants complete control and title subject to the host validation rules. This is where conflicts-of-laws and related theories around jurisdictional competition come more readily into play. If the primary authority, for purposes of token ownership, is the blockchain the token smart contract is actually on, the determination of state is the practical determination of property rights and the ability to enjoy and own associated assets.

While Contract participants may invoke their local jurisdictions or apply for remedies under tort or theories of contract breach to recover lost or stolen balances from others, the inherent nature of the blockchain as a machine to perform digital accounting for Contracts makes it difficult to justify the application of other jurisdictions’ laws to override the Contract itself, except in cases where repudiation would be a logical and equitable remedy and something that would be assumed as an implied or customary term.

Further, since Contract transactions settle immediately, attempting to void their operations through an order to validators to hardfork state would not be a productive or legally feasible way to put parties into their *status quo ante*. If the recipients of stolen or mistaken balances can identify defendants and demonstrate sufficient grounds for recovery, they could receive liquidated remedies in the form of

¹⁵ “A familiar and eminently sensible proposition of law is that, when parties set down their agreement in a clear, complete document, their writing should as a rule be enforced according to its terms. Evidence outside the four corners of the document as to what was really intended but unstated or misstated is generally inadmissible to add to or vary the writing” *WWW Assocs. v. Giancontieri*, 77 NY 2d 157 - NY: Court of Appeals 1990, https://scholar.google.com/scholar_case?case=3099552382612485863&q=four+corners+of+the+contract&hl=en&as_sdt=fffffffffffe04.

offchain currencies, and that would not interfere with Contract or chain state. These kinds of judgment solutions therefore make the most practical sense and would be efficient ways to apply equitable theories to Contracts while honoring the tradition around freedom-of-contract.

Ownership of Accounts and digital assets like tokens therefore is familiar ground to tread in contract and property law. The chief innovation here is the acceleration of interpretation and settlement of Contract instructions with blockchain settlement authority. Given that this power resembles private arbitration, and given also the legacy and historical arc to respect the bargain that parties assert for themselves as objects of legal force, regardless of whether this promise takes the form of an oral or encoded commitment, judges should recognize that even if the laws of participants' jurisdictions may apply to customary issues like fraud or neglect, the blockchain ultimately resolves Contract state. This is something that participants have opted into and it would be beneficial public policy and consistent with precedent to understand such choice of law and Contract to hold their property as disclaiming judicial protection in favor of blockchain enforcement over related property disputes.

Let us now consider more specifically who owns an Account and its underlying assets, which as discussed, can be understood as onchain rights to update Contract state. Those who know the secret or seed phrase to create a private key can be regarded as its owners for purposes of Contract validation. There, of course, are cases where private keys may be coerced and their use would be regarded as criminal or violating other duties. These considerations are not presented to, and would not be fair objects of concern for, blockchain validators who merely are tasked with judging the sufficiency of calldata to match the requirements of Contracts. The slippery slope for adding additional merits should be obvious: Once blockchain validators are asked to preside over the equity or social validity of certain transactions, Contracts on that chain simply will not be trusted by the market as receiving credibly neutral support, and if not through a hardfork, another blockchain may compete for network activity and market share. This disrupts the normal expectations of participants and should be taboo in most judicial circles as revoking the bargain in favor of paternal overrides.

Through updating the state of a Contract, an Account may also be jointly or partially owned by multiple entities. Multisignature authority schemes, for example, may require a certain number of approving signatures to pass an explicit threshold (such as the common, "3-out-of-5 multisig" security setup for protocol managers). Similarly, the right to call and update an Account or other Contracts may rest with a quorum of token holders who coordinate to propose and vote on transactions that are executable by those Contracts.

Ownership in these cases can be regarded as associational by means of holding tokens and committing to common onchain schemes, with acceleration on joinder and liquidation of remedies as seen in DAO Contracts like Moloch—or explicitly crafted to operate much in the same fashion as an onchain company or formal

partnership. Enforcement actions such as those levied against Ooki DAO¹⁴ saw more pointed theories of general partnership law to token holders, above and beyond the DAO's original operators, who had the ability to fund and manage onchain contracts that the CFTC had labeled as requiring regulatory compliance and reporting obligations.

Therefore, as in all things that involve tangible value and risk, Contract ownership—understood as the right and practical ability to update Contract state—implies risks where it may conflict with or create new duties of care. If such authority is split, careful consideration should be applied to how this agency is managed among peers, and whether it makes sense to incorporate individually or as a collective to help avoid the consequence of partnership liability (where partnership Contract damages can be assigned in full to any individual partner). Onchain operating agreements, more commonly seen with LLCs that utilize tokens to record and manage membership, can be expected to cover these and more edge cases particular to the unique opportunity of using a blockchain as the group's settlement authority.

¹⁴ *Statement of CFTC Division of Enforcement Director Ian McGinley on the Ooki DAO Litigation Victory*, <https://www.cftc.gov/PressRoom/PressReleases/8715-23>.

Chapter 4: Auth

As we have discovered so far, the right to update a Contract is much akin to owning it. A wholly-owned Contract, such as a Contract Account, is fully enjoyed by its key holders, which may consist of a private key or another Contract that validates group consensus. But certain functions or “sticks” in the classical conception of ownership of property (as a “bundle of sticks”) can be separately owned and assigned.

ERC173,¹⁵ for example, specifies an ownership standard (Ownable.sol) that has popularly been received as a way to granularly handle recovery and material authority roles in DEX and other protocols that require onchain agents to update state or act to prevent critical security events. Specifically in this case, if a function is marked as *onlyOwner* (or has similar caller-authorization method), it follows that only the owner Account may call it:

```
Examples.sol
1 // SPDX-License-Identifier: AGPL-3.0-only
2 pragma solidity ^0.8.19;
3
4 contract Owned { . . .
5     modifier onlyOwner {
6         require(msg.sender == owner, "Not Owner");
7     }
8
9     function transferOwnership(address newOwner) public onlyOwner {
10        owner = newOwner;
11        emit OwnershipTransferred(msg.sender, newOwner);
12    }
13    . . . }
```

Tokenizing such roles and specific permissions is a nascent space that aims to increase their discoverability and ability to compose with other Contracts. “Soul”¹⁶ or “token-bound” concepts such as extending ERC173-like authorization checks (ERC6551)¹⁷ to holding a specific non-fungible token (NFT) will increase their expressiveness and ability to unlock different experiences (or ways to enjoy such

¹⁵ “ERC-173: Contract Ownership Standard,” Nick Mudge & Dan Finlay, *Ethereum Improvement Proposals*, <https://eips.ethereum.org/EIPS/eip-173>.

¹⁶ “Soulbound,” Vitalik Buterin, <https://vitalik.eth.limo/general/2022/01/26/soulbound.html>.

¹⁷ “ERC-6551: Non-fungible Token Bound Accounts,” Jayden Windle, Benny Giang, Steve Jang, Druzy Downs, Raymond Huynh, Alanah Lam, Wilkins Chung, Paul Sullivan, Aurnyn Macmillan, Jan-Felix Schwarz, Anton Bukov, Mikhail Melnik, Josh Weintraub, Rob Montgomery, vectorized, *Ethereum Improvement Proposals*, <https://eips.ethereum.org/EIPS/eip-6551>.

property). The right to enter digital spaces, like chatrooms, for example, has been one of the most successful ways that authority and access rights have been translated into pure digital properties that can be easily obtained for other digital assets. It is easier than ever in this format to enter and exit organizations and this should be understood as a welcome innovation to contracting.

Layered into all of this, and only limited by certain technical considerations such as hardcoded Contract data limits,¹⁸ can be multifaceted roles, with programmable timelines and administration. This level-of-detail around property authorities exceeds the comprehension of classical contracts, which necessarily are limited by the cognitive and accounting capacity of judicial enforcement.

Likewise, progressive experiments in distributing tokens as sweat or network equity increase the ability to tailor rewards for different and highly-specific forms of value that might otherwise be hard to approximate in a traditional organization, where judges would be more likely to insert preconceptions from corporate law to unfamiliar access and other discretionary rights.

For example, is an NFT owner who can chat with other NFT owners part of an unincorporated association if they co-produce media (often just memes, really)—does holding a digital key to a chatroom or onchain Contract function equate hierarchy and management authority? It ideally should not matter, as they are concerns of Contract interpretation and therefore do not require stringent formalities other than what can be effectively enforced by the Contract code.

This resting topic of enforceability is the limiting factor for Contract advancement. Understandably, property rights can only be as complex as their ability to be actually enforced. Further, the question of whether blockchains will work at least as well as traditional providers will reside on their ability to allow consumers to decide on the amount of custody they are willing to surrender in exchange for quasi-judicial remedies like onchain arbitration (*i.e.*, choose-your-own-immutability).

It is not the position of many blockchain advocates that life savings and critical business operations should be entirely at the mercy of code, but instead, that there now simply exists new kinds of settlement security guarantees.¹⁹ This is a substantial point of interest and confusion for many observers of this technology—that Contracts and encoding aspects of association into blockchain data will doom them to the mercy of opsec. Instead, the well-understood role of private arbitration to support agreements can provide safer Contract use with onchain Auth that are native to their unique format.

¹⁸ “EIP-170: Contract code size limit,” Vitalik Buterin, *Ethereum Improvement Proposals*, <https://eips.ethereum.org/EIPS/eip-170>.

¹⁹ See, e.g., the emergent doctrine of “qualified code deference,” <https://github.com/lex-node/SCoDA-Simple-Code-Deference-Agreement>.

Chapter 5: *Lex*

When all of these components come together, *Contracts*, *Accounts*, *Owners*, and *Auth*, the makings of a novel jurisdiction for both choice-of-law and resolution in agreements becomes more apparent.

And it makes sense. Finality is the service of The Law, and the emergence of blockchains represents a social, more so than technical, upgrade on coordination for this purpose. Historically, settling disputes was intrinsically linked to those with social authority to use force to seize property—The State. Let's therefore consider this context more closely for *Contracts* and how they can be governed in ways that accommodate the full range of finality preferences.

Evident from actual market activity, the ability to trust blockchains as *truly final* allows users to build increasingly complex financial tools. For example, by being able to rely on hardcoded values in a token *Contract*, like DAI, entirely code-based organizations and multi-billion-dollar protocols can build on top without fear of their foundations breaking. As discussed, but worth repeating: interference with *Contracts* should be met with a high-degree of skepticism to avoid unexpected externalities and cascading damages.

Classical contracts under this regime should therefore be better suited for matters not managed by computers or that necessarily benefit from mutability. More often than not, they can be expected to more simply provide additional *Contract* context and elaborate on exceptions to code, such as Readme files.

A hybrid finality service—*Lex*—blending the fast justiciable medium of blockchain execution with public legal determinations (judgments), is desirable to expand contracting online, but also fill the critical justice gap for *Contracts* that lack a formal resolution mechanism over their *Auth* or state.

Aside from panels of arbitrators assigned to *Contracts*, there are novel mechanism designs rooted in crypto-economics to allow crowds to coordinate as juries to provide judgments. As these are tested and compared to panels or others with similar legal training, the market will continue to grow for onchain resolution, and competition will allow for increasingly efficient services.

We can look ahead to these early resolution tools, and speculate on how their norms will harden to provide case precedents and accomplish fairness. It would seem desirable to have such tools at the market's disposal as soon as possible, with the safety valve of human-operated services pushed more and more to the edges. Until that day, judgment machines will be effectively an assemblage of tools (or, in the LLM parlance, a "panel of experts") combined into conveniences like any other software application—recordation, execution, and resolution.

Coda

We hopefully at this point have achieved a basic intuition on where everything is headed with Contracts—where *Code*, for all intents and purposes at least for matters of private agreements, works *As Law*. You should also be able to build from the excerpted examples and work to reduce paperwork and bloat in your dealings.

What should hopefully most resonate at this point in your adventure is the realization that encoded commitments appear fundamentally fair and resist the uncertainties of our times by making key aspects indisputable. They also make it far simpler and cheaper to manage property and specify ownership.

The natural tradition resonates deeply here. *Prediction*: A body of contract law will continue to advance in support of purely digital assets, by reconstructing remedies usually provided for agreement defaults in traditional legal systems, with internet-native enforcement mechanisms.

This is not anything like anarchy, and works to complement and de-obstruct aspects of commercial and social governance that are at odds right now with the proliferation of the internet and its values. Classical contracts allow us to easily understand that not much will change here, after all, and that the advancement of legal systems with their cryptographic forms, like the coordination among most communities, will be driven by incentives and needs.

Right now, it should be clear at least that, universally, time is valued among all market participants, and wherever technology serves the legal sector to save time while not changing the quality of its results, it should be fully embraced. Lawyers and coders alike will get to be on the ground floor of this transition, and that should be something you lean into (and most importantly, forget to ignore!).