Implementing A Letter Of Credit Style Business Process For Small-Scale Contracting Using Smart Contracts

Mathew B. Fukuzawa[1], Michael G. Kay[2], Brandon M. McConnell[2,3], Kristin Thoney-Barletta[4], and Donald P. Warsing[2,5]

[1]Operations Research Graduate Program, NC State University, Raleigh, NC
[2]Department of Industrial & Systems Engineering, NC State University, Raleigh, NC
[3]Center for Additive Manufacturing and Logistics, NC State University, Raleigh, NC
[4]Department of Textile and Apparel, Technology & Management, NC State University, Raleigh, NC
[5]Department of Business Management, NC State University, Raleigh, NC

Abstract

Purpose - Demonstrate proof-of-concept for negotiating and executing a small-scale contracting job based on letter of credit within a blockchain network using smart contracts.

Design/Methodology/Approach - Using Ethereum smart contracts, we model a small-scale general contracting scenario under perfect conditions. Execution is demonstrated with the Remix Integrated Development Environment (IDE).

Findings - We show the feasibility of conducting a small-scale contracting job using smart contracts. The entire process, including job details, payment, and verification, can be conducted digitally.

Originality/value - This research continues the efforts of previous research on letters of credit on blockchain but applies it to the general contracting scenario at small scale.

Research limitations/implications - Further work is required to investigate variations in the assumptions, such as dishonesty and incompetence. Also, full-scale decentralized application is not explored here.

Practical implications - This process expands the scope of current practices and tools, such as Angi, in a decentralized manner with blockchain.

Social Implications - Full-scale adoption at the small scale is likely difficult due to disbelief in technology, cost, and resistance to change.

Keywords: blockchain, smart contract, letter of credit, general contracting, contractor, contract.

Paper type: Research paper.

1 Introduction

Many homeowners encounter a situation that requires hiring a contractor to perform a job. These scenarios are typically one-time interactions between contractor (CTR) and client (CLT), involving smaller dollar values to process an exchange for a service. In many ways, this process resembles the international trade environment, where a seller and buyer interact to exchange goods. In fact, some of the same issues persist on a smaller scale. Given no previous history of interaction, there is an inherent lack of trust between CTR and CLT. A CTR has no guarantee of payment, while the CLT has no guarantee of quality. Internal enforcement options are limited; effective means are costly and time-consuming through a third party. The fallout from a failed relationship must still be dealt with by either party. Finally, there is considerable leeway with CTR credentialing requirements. In the state of North Carolina (NC), CTR licensure is only required if the total cost of the project exceeds \$30,000 (NCLBGC, 2023*a*). While licensure is not a safeguard against job dissatisfaction, it can provide peace of mind for the client in the form of CTR competence and knowledge of state building codes (City of Cary, NC, 2023). Likewise, there is no requirement for NC contractors to be insured (NCLBGC, 2023*b*), nor are written contracts mandatory.

A CTR has several means to address issues and protect interests, from proactive measures like addressing payment terms in a contract to reactive measures like filing a lien on a CLT's property. While contract terms establish agreed-upon guidelines, they are not a guarantee of payment, and payment schedules are highly inconsistent across the contracting community. Liens can also be effective, but they do not guarantee payment either.

The CLT's proactive methods mainly include research. Public review sites like Google and Yelp can help provide insight into a CTR's history. However, these ratings are often highly subjective, emotionally charged, and inconsistent. Furthermore, ratings and reviews are not always accessible for a CTR. Third-party (3P) services such as Angi, HomeAdvisor, and Frontdoor can provide access to a list of vetted CTRs. Some of these can even manage the payment process. However, these services are largely just connection routes akin to the digital white pages. Additionally, certain features and benefits are only available to those customers who are willing to purchase a membership. Most notably, these 3P services do not facilitate contract negotiation or execution. In terms of reactive measures, CLTs have traditional and non-traditional means to address breakdowns in the process. The legal system is always an option, but parties may find this route to be costly and time-consuming. Within NC, a CLT may seek administrative actions against a CTR via a state contracting board, but the board has no power to force work, reimburse funds, or recoup property (NCLBGC, 2023*b*). Non-traditional means include intimidation and threats, along with crowdsourced pleas for assistance. Neither of these methods has a noteworthy history of success.

Blockchain is a potential solution to address some of these issues with the contracting process. General blockchain design relieves pressure from trusting the other transacting party. All transactions are recorded, and the ledger is immutable, so a complete history is available at all times. Most currently accepted payment schemes are easily translated into smart contracts. Privacy can be adjusted as needed with the appropriate network type, depending on the requirements. Finally, blockchain provides the potential for a self-contained process through inclusion in a decentralized application (dApp). With these programs, all elements of the contracting process, including CTR research, credential verification, contract specification, payment, execution, and termination, could be encompassed within one system. Additionally, this application highlights the exclusion of a 3P, although there is room to expand this concept with future research.

We borrow the smart contracting letter of credit (LoC) framework from Chang, Chen and Wu (2019) to address the various functions of the small-scale contracting process. The LoC is a tool to facilitate international goods trading, touted as a secure, guaranteed form of payment (US Department of Commerce, 2023). Furthermore, this research builds on the work of Toorajipour et al. (2022) by exploring a potential use of blockchain technology to facilitate independent business transactions at a smaller scale. There is a growing call for this sort of expansion of blockchain research as it is currently focused narrowly on cryptocurrency and related financial transactions (Centobelli et al., 2021, Chan et al., 2023, Kimani et al., 2020, Pereira et al., 2019, Toorajipour et al., 2022). Thus, our paper proposes and models a new application area for blockchain smart contracts to address this critique, with emphasis on how the framework could be used to overcome the inherent lack of trust within contracting and how it provides alternative enforcement options to traditional legal avenues.

Following this introduction, a literature review is provided to help understand the related research involving contracts, including enforcement and incentive alignment. Afterward, a conceptual framework is presented for managing the small-scale contracting process based on the LoC smart contract design. Demonstration of the process is shown through examples from the Remix Integrated Development Environment (IDE). Finally, suggestions for future study are presented.

2 Literature Review

The literature review covers a broad range of topics. Some work is necessary to highlight elements of contracts, particularly with regard to negotiation. Other research provides background on business-related elements of contracts. Finally, we explore recent research on large-scale construction and trade.

2.1 Contracts & Negotiation

2.1.1 Contracts

Contracts are commonly used to facilitate negotiation, so a short understanding on their characteristics is necessary. Bix (2012) identifies three elements required for contract existence: offer, acceptance, and consideration. An offer must be reasonably definite and not vague, and it does terminate. Accepted offers imply that parties have agreed to terms, which should appear on the same form (a different form, e.g., paper, may constitute a counteroffer and thus not prove acceptance). Even without a signed contract indicating acceptance, the following acts are generally sufficient to prove acceptance: sending or accepting payment, delivering goods, or performing service. While it is not required that an agreement be in writing, there should be evidence indicating an agreement was initiated. Of course, consent cannot be gained when under duress or subject to undue influence. Consideration marks the difference between contracts and gifts; with a contract, there is a promise to return something of value.

2.1.2 Enforcement

Williamson (1996) argues that economic transactions are subject to opportunism, and hence contract provisions must safeguard against this threat; regardless of contract complexity, however, contracts are always incomplete as they fail to describe every possible scenario. North (1990, p. 54) states, "the inability of societies to develop effective, low-cost enforcement of contracts is the most important source of both historical stagnation and contemporary underdevelopment in the Third World." North (1990) continues about the problems with contract enforcement:

- when enforcement is costly or not present, cheating and dishonesty are more appealing than cooperation;
- the cost of providing enforcement may exceed the gains that parties can benefit from it;
- standard game theory assumptions of perfect information and indefinite play are inappropriate for the real world in justifying self-enforcing solutions;
- a sound legal framework is required to handle enforcement violations, coupled with a method for identifying punishment and incentives for carrying it out; and
- enforcement is costly, but a third-party coercive body may use their power to reshape markets in a way that creates perceptions of unfair advantages to some groups.

Cannon et al. (2000) argue that while contracts are important for assigning obligations, their inability to adapt to environmental uncertainties allows parties to behave selfishly. Poppo and Zenger (2002) argue that formal contracts are complements of *relational governance*, a set of social exchange rules that control exchanges; with a higher level of relational governance, managers tend to employ more complex contracts. Others assess the efficacy of the legal system (Ho, 2001, Johnson et al., 2002, Shou et al., 2016), which is often associated with contract enforcement.

Some authors examine trust in the presence of contractual control measures, where increased restrictions can lead to decreased trust (Malhotra and Lumineau, 2011, Zhou and Poppo, 2010). Bai et al. (2016) suggest that a contract based on behavior only engenders buyer-supplier conflict because it is likely to arouse defensive attitudes; output-based contracts attenuate conflict because clear links exist between output and rewards.

2.1.3 Escrow

Escrow accounts are another form of contract, typically between one or more negotiating parties and a neutral third party. Because smart contracts are used in this paper to hold funds, their functionality can be described in a similar manner to an escrow account. The term *escrow* refers to any account where funds are held and disbursed by a third party, with the most common usage occurring in mortgage lending (Mills, 1994).

The use of smart contracts as escrow accounts is fairly standard practice, but research focuses on design differences. In particular, the use of a third party is of high interest due to the fact that blockchain's original intent was to eliminate the need for outside verification. Zimbeck (2014) proposes one of the first strictly two-party escrow designs for cryptocurrency trading, called BitHalo. Within BitHalo, both sender and receiver are required to verify receipt of product off-chain; if parties do not agree, then all funds are burned (lost) (Zimbeck, 2014). Meng et al. (2019) propose Themis, another two-party escrow design, but parties seek arbitration from a group of peers on the network in the case of a dispute. Asgaonkar and Krishnamachari (2019) propose an improvement to BitHalo, whereby verification is performed on-chain, assuming the asset in question is a digital commodity that can be assigned a unique cryptographic hash. Schwartzbach (2021) proposes a two-party escrow design for any physical good or service that is based on game theory, but disputes are still resolved through an arbiter. In contrast, Goharshady (2021) argues that game-theoretic assumptions of simultaneous play in a two-player non-cooperative game are inappropriate for smart contracts; transactions are visible to other users, who can ruin the process with an extortion attack.

2.1.4 Negotiation

Negotiation is as much art as it is science, and thus there is much research on the topic in other disciplines such as psychology, economics, and management. Early work focuses on game theory with standard assumptions of rational behavior and self-interest (Messick and McClintock, 1968, Nash, 1951, Von Neumann and Morgenstern, 1947). Brett and Thompson (2016) draw on the earlier work of Walton and McKersie (1965) in describing four sub-processes of negotiation. Several authors refute game-theoretic assumptions because of the presence of trust, which promotes information exchange and affects risk-taking (Butler Jr, 1999, Kong et al., 2014, Mayer et al., 1995). Brett (2000) defines two types of negotiation—transactional and conflict resolution, arguing that there is a need for negotiators to understand the cultural differences between sides since these may impact the negotiation process.

Emotions can also have profound effects on negotiation outcomes, such as anger, which generally causes retaliation (Friedman et al., 2004, Wang et al., 2012). Barnes (1981, p. 110) addresses a common manager assumption that the world is unsafe so a person should "adopt a position of pervasive mistrust to survive"; instead, we should search for *paradoxical actions* which counter these assumptions.

The small-scale contracting environment presents scenarios that may be characterized by a power imbalance, suggesting that either the customer or the contractor has a greater level of power in negotiation or in enforcement. Fisher et al. (2011) argue that any agreement should be measured against a best alternative to a negotiated agreement (BATNA); negotiating power is determined by the attractiveness of a BATNA. It is also suggested to consider the opponent's BATNA, although this is likely unknown. Some researchers examine the level of power in negotiation, with different behaviors along the power spectrum (De Dreu, 1995, Van Kleef et al., 2006, Wong, 2014).

Negotiation involves cooperation, which in turn involves incentive alignment. Supply chains are classic examples of where failure to properly align incentives among key nodes leads to issues with performance (Biswas, 2011, Norrman and Naslund, 2019, Simatupang and Sridharan, 2005). Others explore incentive alignment of policy or structure adoption at the strategic level (Belfo, 2013, Lv et al., 2022). Other recent work uses game theory modeling of incentive alignment to develop solutions for two-party interaction (Haagensen and Debois, 2022, Heindel and Weber, 2020, Noorian et al., 2014, Wu and Wang, 2023).

Additionally, blockchain adoption is another area of active research for incentive alignment. In particular, blockchain offers a more realistic environment in which to model situations that do not require such restrictive conditions as game theory. Within a supply chain, some research examines how participants benefit from blockchain usage (Niu, Dong and Liu, 2021, Niu, Shen and Xie, 2021, Niu et al., 2022), while other work focuses on the incentives for adoption (Li et al., 2022, Rejeb et al., 2021).

Clohessy and Acton (2019) study general factors for blockchain adoption within an organization, citing the following variables as pertinent:

- Developed countries: these countries are likely to employ greater initiatives in promoting blockchain and incentivizing its use;
- Remove the stigma: an association with the term *cryptocurrency* can carry a negative connotation, and organizations need to separate the concept from the improper use examples;
- Management support: positive influence from the top-level leadership is a requirement, supporting blockchain research and initiatives; and
- Large versus small: organizational size affects willingness to incorporate blockchain, with larger companies finding greater benefit; smaller companies struggle to find a need for blockchain and complain about complexity; acceptance is more likely if organizations are allowed to experiment with cost-effective solutions prior to adoption.

2.2 Human Behavior

Several theories on human behavior help to frame our understanding of motivation and purpose in achieving desirable outcomes. Isaac et al. (2001) discuss *expectancy* theory, the idea that individuals act out of self-interest and their effort is tied to a belief in the level of performance attainable. Eisenhardt (1989) explains *agency* theory, where a metaphorical contract is used to govern the relationship between a principal and their agent; it also assumes that humans act out of self-interest, so most research aims to study the governance mechanisms that defend against agent opportunism. Alderfer (1969) proposes an adaptation of Maslow's hierarchy of needs, which he calls existence, relatedness, and growth, or *E.R.G.* theory; humans' core needs include existence—the obtaining of material and physiological desires; relatedness—the seeking of significant relationships with others; and growth—the search for opportunities to develop and reach full potential. Kaplan and Norton (1996) capitalize on the use of rewards to motivate behavior; they propose the *balanced scorecard*, a tool to link organization strategy to individual performance objectives typically using financial rewards.

2.3 Business Relationships

The management of business relationships is widely studied, with numerous connections to entrepreneurship, economics, game theory, and much more. For business relationships that involve multiple interactions, there are several noteworthy studies, namely, in the realm of contracts and power. Baker et al. (2002) define the term *relational contract* to help explain the value of future interactions; "relational contracts cannot be enforced by a third party and so must be self-enforcing: the value of the future relationship must be sufficiently large that neither party wishes to renege." Qian et al. (2020) explain that within established relationships, contractual execution and relational norms contribute positively to collaborative performance; legal enforcement has a positive impact on the relationship between contractual execution and collaborative performance.

Xie et al. (2019, p. 1265) explain "where there is an interdependence relationship, there is a power conflict; where there is a benefit, there is opportunism." Furthermore, when a buyer (in a buyer-supplier relationship) has a higher level of confidence in the legal framework, he/she is less inclined to spot opportunities for personal gain (Xie et al., 2019). Antia and Frazier (2001) reveal that contract enforcement is related to the level of interdependence among business channel members, e.g., a firm may benefit from a power advantage when its level of dependence is low because it is less concerned with consequences; a firm with a power advantage may have low interest in maintaining quality relationships. Kashyap and Murtha (2017) find that with a greater amount of enforcement completeness, there is a negative effect on franchisee compliance; in essence, this scenario yields a *consummate compliance*.

2.4 Large-Scale Projects

Construction and international trade are two areas that have benefited from recent research on smart contracts.

2.4.1 Construction

Construction is a lively research area because of its numerous processes that warrant the study of optimization or efficiency. Much research is focused at the large scale, international level because of construction's global reach. Some researchers explore the causes of payment disputes (Abdul-Rahman et al., 2014, Liu et al., 2019). More recent research sheds light on applications of blockchain technology to the construction process (Li et al., 2019). Ahmadisheykhsarmast and Sonmez (2020) demonstrate smart contract usage within a dApp that collects contract terms, estimates progress points and payments, blocks payments for 30 days, and releases payment to all personnel levels within a cascading payment system.

2.4.2 International Trade

International trade (and trade in general) is an area subject to trust issues. In particular, the finance associated with this trade is a source of enduring trust matters between banks and trading partners. Kowalski et al. (2021) find that blockchain technology can improve the security of transactions and improve communication, but more research is needed to uncover how trust can be improved; blockchain alone is not enough to develop trust between parties. In the accompanying survey, one interviewee states, "There is still a way that someone sends lead instead of gold. Blockchain can't eliminate that, because one can always find oneself in a situation where someone sends something that resembles what has been stated in the contract but with a lower quality" (Kowalski et al., 2021, p. 5).

Others focus more specifically on the LoC process and blockchain technology. Chang, Chen and Wu (2019) address the friction points with the LoC process and propose a redesigned framework using three smart contracts to manage documents, cash, and logistics. Toorajipour et al. (2022) also propose the conceptual design of a blockchain-based LoC process, but modeled as a pure peer-to-peer (P2P) transaction (no third party bank). Furthermore, Toorajipour et al. (2022) identify a need for future blockchain research to explore other areas besides those relating to cryptocurrency, suggesting an examination of business transactions.Chang, Luo and Chen (2019) conduct a review and note the benefits observed from using blockchain: transparency, ease of information transmission, traceability, dis-intermediation (removal of intermediaries), cost reduction, and the ability to incorporate the internet-of-things (IOT).

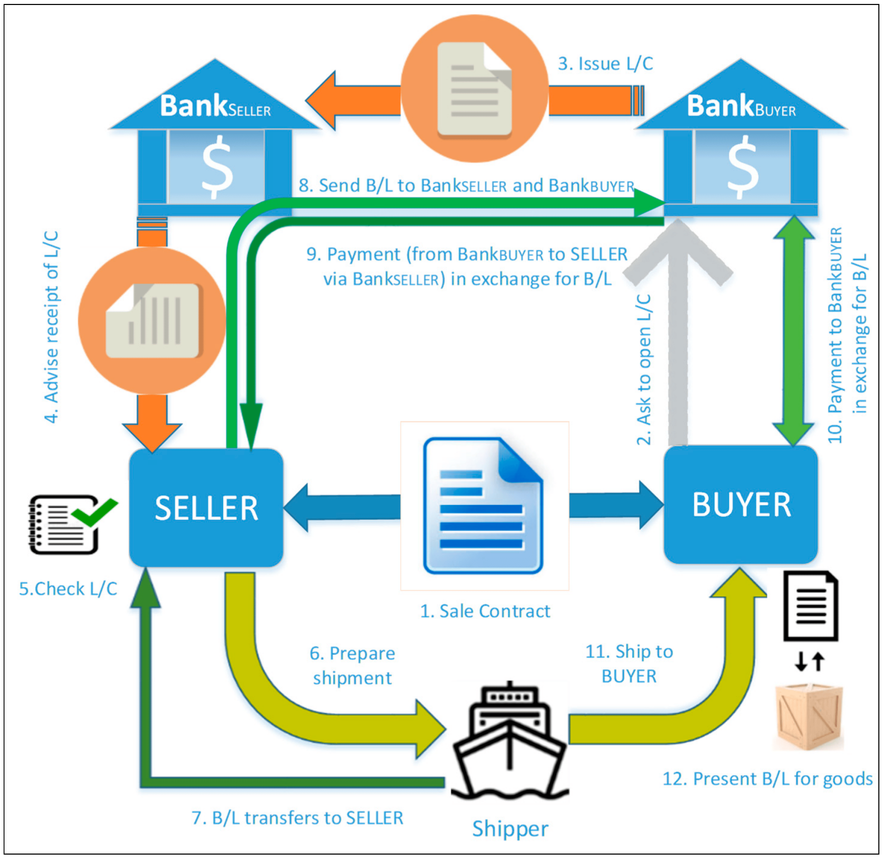
3 Modeling

3.1 Framework

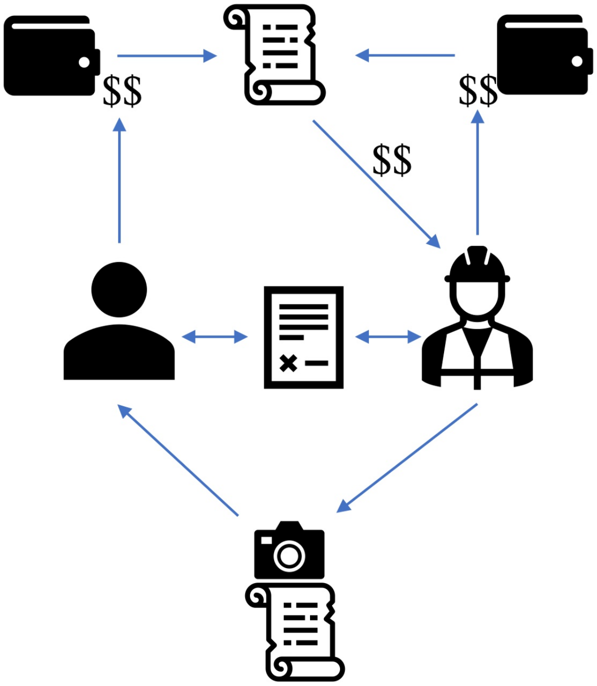
As a trade finance tool, the LoC involves large amounts of money with new trading partners employed on a large-scale, international level. The LoC is often used in higher-risk situations when credit is poor or unknown; thus, the barrier to entry is also high due to the accompanying financial regulations. The LoC is deemed secure because it provides a payment guarantee, i.e. the process forces proof of trust when the shipping bill of lading transfers from the seller to the buyer's bank. The bill of lading transfer is a requirement for the buyer's bank to initiate payment (US Department of Commerce, 2023).

Figure 1a shows the traditional LoC process; it begins after a contract is established between a seller and buyer (1). The buyer then applies for a LoC with their bank (2), which contains pertinent details of the shipment and payment terms. The buyer's bank then sends the LoC to the seller's bank (3), who then reviews it with the seller for accuracy (4/5). The goods are then prepared for shipment (6), and the bill of lading (B/L) is prepared. The B/L is a legal receipt document describing the shipping contents and mode of transportation and delivery (U.S. Customs and Border Protection, 2023). The shipper provides the B/L to the seller (7); it is then transferred between seller and buyer banks (8). In exchange for the B/L, the buyer's bank sends payment to the seller's bank (9). The buyer must then pay their line of credit in order to receive the B/L (10), which is then used to claim the goods upon arrival (11/12).

With this research, we consider the same LoC philosophy but implement the concept at the smallest scale between CLT and CTR. While typical domestic laws still apply, the barrier to entry with this process is low despite the same issues with a lack of trust. Figure 1b depicts the parallel idea with small-scale contracting. An agreement is made between CLT and CTR, the terms of which are typically enclosed in a contract. Here, we move those terms to a smart contract. Both CLT and CTR have funds tied to a bank; the parallel idea in the blockchain setting is that both parties are tied to a digital wallet with funds. We consider proof of job progress in the form of digital image submission as a requirement to transfer payment to the contractor. The image verification process mimics the B/L transfer in the LoC structure.



(a) Typical LoC process (adapted from Chang, Luo and Chen (2019)).



(b) Small-scale contracting framework.

Figure 1: LoC philosophy at large and small scales.

3.2 Methods

We use Ethereum to design the smart contracts within a hypothetical blockchain network, but the image submission process requires some off-chain techniques. By design, blockchain is not intended to be a centralized storage solution. Thus, for various memory and cost reasons, it is impractical to store large files such as images on the blockchain. An alternative option is decentralized storage, whereby a file is indexed by its content; a unique hash is assigned to the file (IPFS, 2023*c*, Sen, 2023), which we can store in a smart contract variable (IPFS, 2023*b*).

The off-chain protocol employed here is the InterPlanetary File System (IPFS), although there are other decentralized protocols. For the purpose of this research, the CLT and CTR are both participants on IPFS. The CTR's responsibilities include uploading digital images of job progress to IPFS, and storing the associated hash(es)[1] in a smart contract. The CLT retrieves the hash(es) from the smart contract, queries the IPFS network for the hash, and retrieves the image. In this implementation, IPFS uses a base58 encoding scheme, producing a 46-byte hash (IPFS, 2023*a*).

3.3 Process Overview

Figure 2 provides an overview of the proposed process. The entire transaction between CLT and CTR is conducted via three primary smart contracts. The contract descriptions below, described in general, refer to the numbering system within Figure 2. The circled letters in Figure 2 refer to a sequencing of steps; all other numbers are merely for reference. All three contracts will be explored in greater detail within the implementation section.

3.3.1 Scope of Work Contract (SoWC)

The SoWC captures the job requirements and other basic information; it replaces the function of a traditional written contract. In our implementation, a rudimentary set of contract details are included to show functionality, but this can be expanded to provide greater depth. Once deployed (1a), the CLT submits the job requirements to the contract with another set of basic details to initiate negotiation (1b). The CTR responds to the initial proposal with their capabilities (1c). The CLT can either accept or reject those terms after review (1d/1e); if rejected, the CLT may propose a new set of terms or terminate the process (1f). If the terms are accepted, the CLT finalizes the contract (1h), which simultaneously deploys an instance of the payment contract (2a). In parallel, the CTR submits credentialing info (1g), if applicable. For this project, credentials include a CTR's license number and insurance verification. These are hypothetically verified via oracles (1j) against a state database (see assumptions). For example, a NC CTR license is a seven-character designation, e.g. L.12345.

3.3.2 Payment Contract (PC)

The PC is the means by which funds are transferred between CLT and CTR. The CTR submits a stake to the PC (2c/2d) while the CLT submits payment for the job to the PC (2e, denoted as *deposit*). Note, *deposit* and *payment* are used interchangeably hereafter when referring to the CLT. The purpose of the stake is to keep the CTR honest, with the exact amount determined by the results of the credentialing check (2b) in the SoWC (1g/1j). If the CTR is credentialed (i.e. license and insurance = true), then the stake amount is low; else, the stake is higher. Both the stake and the deposit are blocked in the contract until certain conditions are met. The CLT deposit (2e) simultaneously deploys an instance of the job progress contract (3a). When the job is completed, the CLT deposit is released to the CTR (2f), after which the CTR may recoup the stake (2g).

3.3.3 Job Progress Contract (JPC)

The purpose of the JPC is to manage the job verification efforts between CTR and CLT. This is analogous to the transfer of the bill of lading in the LoC process; payment is not transferred until the job is verified as complete. The JPC is deployed from the PC with only the CLT's Ethereum account address; it is minimally

---

[1] Note: The term *hash* is lazily used here; the IPFS returned value is called a content identifier (CID).

structured because most work is performed off-chain. In Figure 2, the Image Contract (IC) is shown separate from the JPC, but this is merely for demonstration purposes; they are considered to be part of the same contract.

The CTR uploads a digital image of the job progress to IPFS (5a), receives a hash (5b), and submits it to the IC (4a). The CLT can then retrieve the hash from the smart contract (4b). Note while it's possible to manually retrieve the hash with a simple view function within the contract itself, this implementation automates the process with an off-chain script file (5d). The script file interacts with the chain by retrieving the hash and subsequently querying the IPFS network for the image (5c). If the job is complete, the CLT changes the `jobDone` variable to true (3c); otherwise, the CLT continues to recheck IPFS for updated images. When the job is verified complete, the CTR may access the CLT deposit in the PC (2f). Note, the payment is only released if `jobDone` is true.

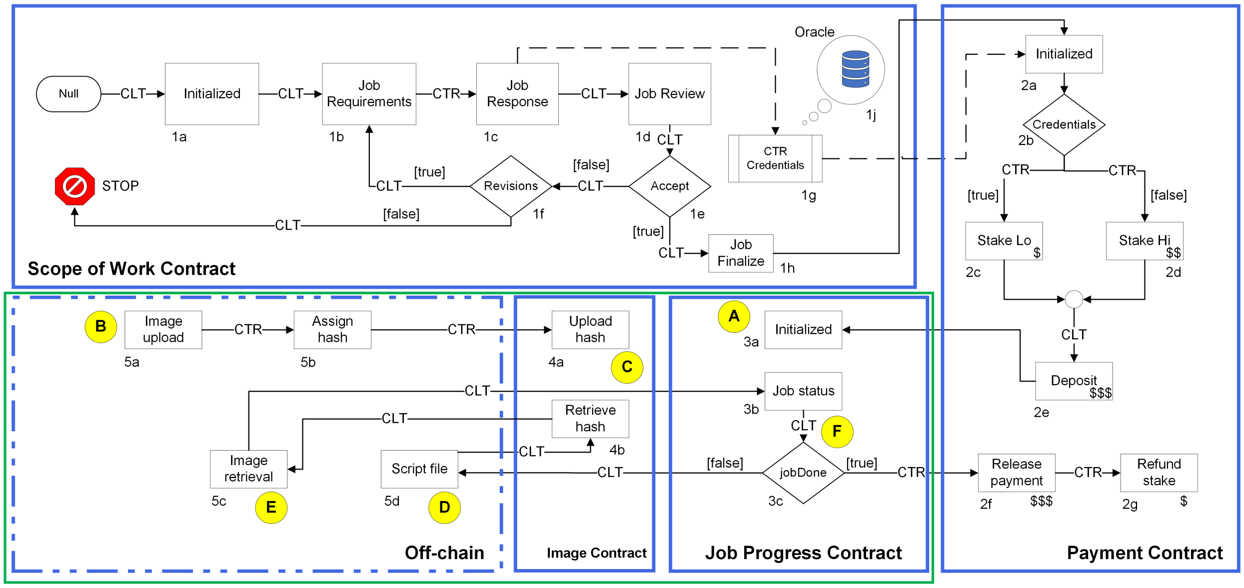


Figure 2: Overview of the LoC process for the contracting job with smart contracts.

3.4 Assumptions

In order to model the single-use contracting job with a blockchain structure, we make the following assumptions.

3.4.1 Necessary

Scale: This project is intended for small-scale use in all applicable elements, namely cost, construction/job scope, and personnel involved. We acknowledge that cost is a real concern at the small scale, including blockchain transaction costs. However, we ignore the blockchain-related costs for this implementation.

Oracles: The process of CTR credential verification is done via oracles which are used by the smart contract to select the appropriate stake amount. Oracles are not physically employed in this project but are assumed to be present.

Image tampering: We are not concerned with digital image alteration; we assume that photos are not edited for nefarious means, and furthermore, the capability to detect such alteration exists.

Metadata: The required image metadata in this implementation are GPS location and date-time stamp; see the previous assumption.

Cost amounts: Since ETH values fluctuate, the stake and deposit amounts are hard-coded with conversion rates as of May 2023. While APIs and conversion tools exist with smart contract integration, they are not employed here for simplicity.

Node Usage: Both client and contractor are assumed to be participating nodes on IPFS.

Banking: Each party is assumed to have funds tied to a bank or digital wallet; this connection ensures funds availability and is analogous to the link between shipper/buyer and bank in the LoC process.

Security: While there are certain purposeful design elements injected to prevent some actions, this research does not design for security against attack. Proactive smart contract security measures are assumed to be employed here, but further work is needed to eliminate this assumption.

3.4.2 Simplification

Power: There is a perceived power imbalance at the small scale in favor of the CTR, who has more accessible enforcement mechanisms (e.g. placing a lien on the CLT's property, requiring prepayment for services, imposing interest on unpaid funds, etc.). Furthermore, cost favors the CTR in the sense that the CLT's cost of enforcement is higher than the cost of performing the service (primarily due to the cost of legal action).

CTR Stake: The purpose of the stake is to deter CTR malfeasance. The amount set in this implementation is arbitrary at 1 ETH and 2 ETH for the low and high stakes, respectively.[2] We assume that the amount is affordable for the CTR but high enough to be painful if the contract terms are not followed. Additionally, no negative actions dealing with the stake are explored in this base case, e.g. with CTR dishonesty, we may choose not to refund the stake.

Blockchain Network Type: The setup lends itself to either a private or consortium-style network, with emphasis on CLT and CTR history remaining hidden. Access is permission-based so that no external party can interfere with the chain without approval. Furthermore, although this implementation is shown in Ethereum, i.e. native currency is ETH, the project scope should generalize to any blockchain network with a corresponding conversion to paper currency.

CLT Payment: In this simple case, the client deposit is equal to the cost of the job; other payment amounts and options are not explored here.

Participant Motives: Neither the CTR nor the CLT intends to deceive or cheat the other party. Thus, no options to deal with deceit are shown, e.g. withholding payment, diverting stake, etc.

Trust: To align a LoC system at smaller scale, we assume that this CLT-CTR relationship is new and not a long-standing, positive relationship. Assume that there is a lack of trust between both sides.

Pricing order: In this instance, the CLT proposes a budget before the CTR. Other cases may include CTR-initiated pricing, but that feature is not modeled here.

Language: Although this implementation uses a combination of Solidity, Javascript, and HTML to demonstrate off-chain concepts, there are other specialized languages that can achieve the same end state; nothing here is prescriptive.

---

[2] As of 31 May 2023, 1 ETH $\approx$ \$1,856 USD; 2 ETH $\approx$ \$3,712 USD.

Contract Terms: The `jobReturn()` and `jobCapability()` functions are meant to capture basic information about the agreement between CLT and CTR. By no means is this intake process intended to completely replicate the design and body of a traditional written contract, but it is assumed that code can be adjusted to do so if required.

4 Implementation

The following subsections demonstrate the use of smart contracts in this application. All smart contract testing was performed in the Remix IDE.

4.1 Scope of Work Contract

The SoWC is depicted with greater detail in Figure 3; once again, alphanumeric references refer to the notations within this figure.

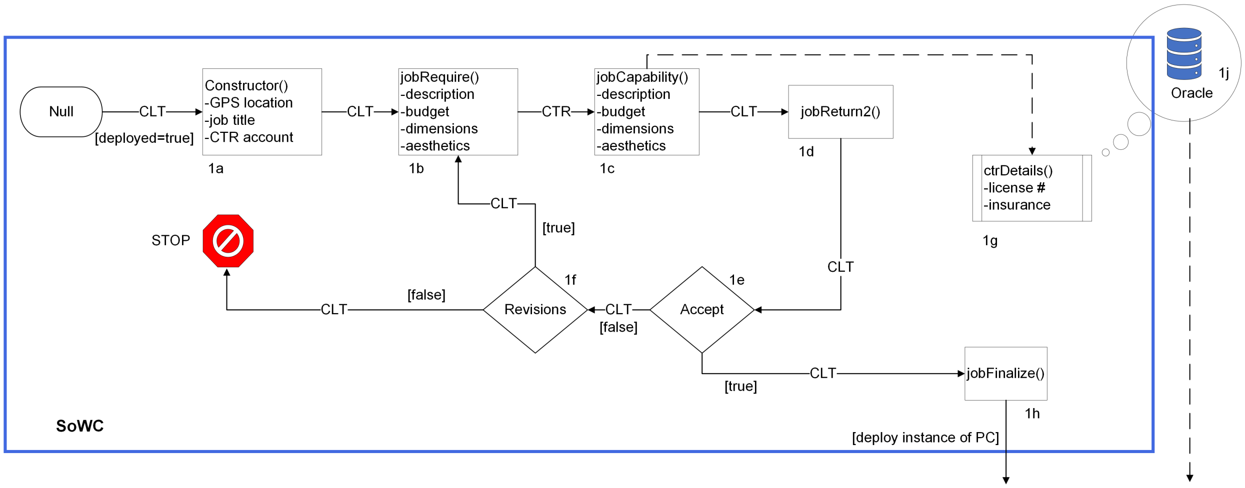


Figure 3: Scope of Work contract.

The constructor (1a) requires four inputs: the latitude and longitude of the job, a job title, and the CTR's Ethereum account address. For simplicity, only the numeric portion of the decimal degrees is required; see Figure 4 for how this is performed in Remix. For example, this figure represents the GPS location (35.7703829, -78.6751011).

LAT: 357703829

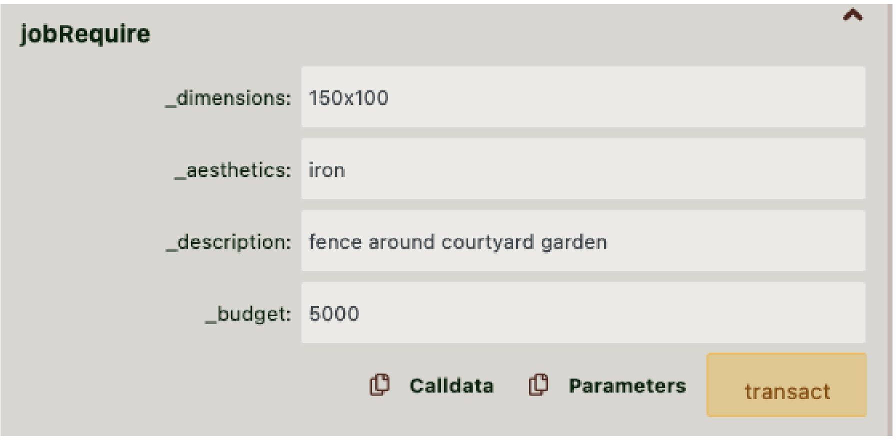
LONG: 786751011

JOB: install fence

CTRADDRESS: 0xAb8483F64d9C6d1EcF9b849Ae677dD3315835cb2

Figure 4: SoWC constructor.

Following deployment, the CLT submits a set of job parameters to the `jobRequire` function (1b). The first three inputs are accepted as strings; the last is a positive integer describing the CLT cost budget for the project. We assume that the units for budget are USD in this example, but that is not necessary if conversion tools are used; USD is used for ease of understanding.



_dimensions: 150x100

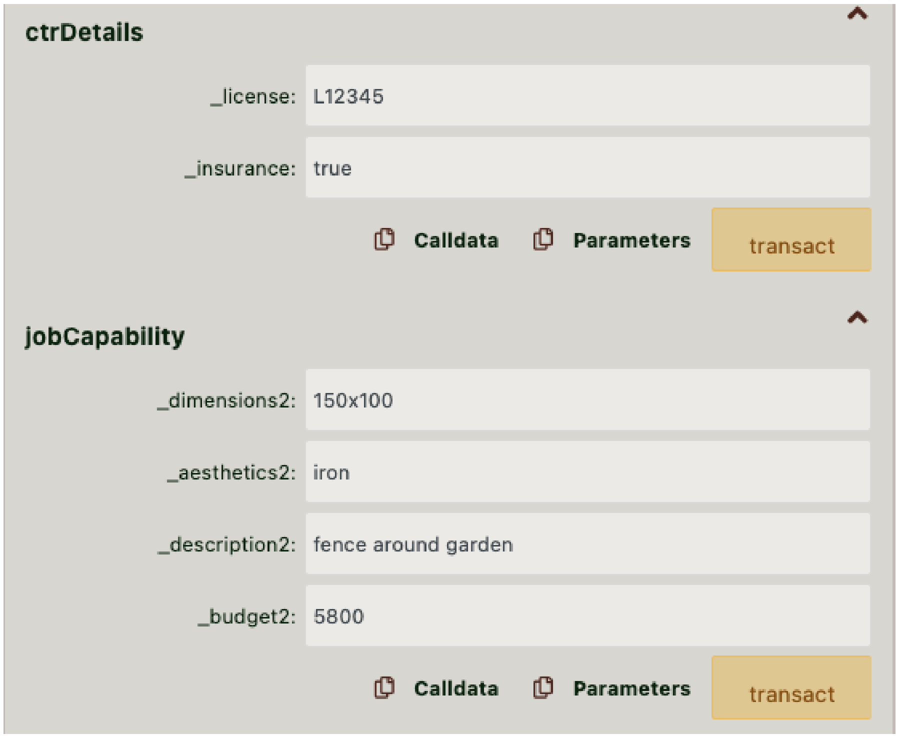
_aesthetics: iron

_description: fence around courtyard garden

_budget: 5000

Figure 5: CLT submits job requirements.

The CTR is then provided an opportunity to respond to the CLT demands (1c). To simplify this, the CTR inputs are the same as the CLT. Also, recall that CTR credentials are submitted in parallel to this process (1g). Both of these functions are displayed in Figure 6.



ctrDetails

_license: L12345

_insurance: true

jobCapability

_dimensions2: 150x100

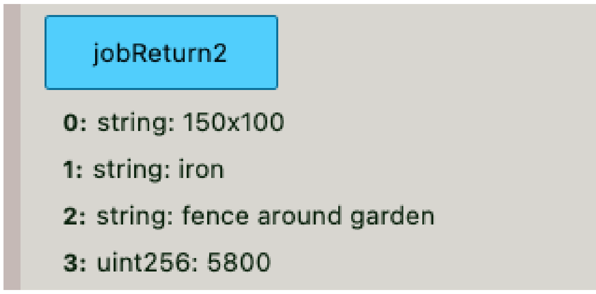
_aesthetics2: iron

_description2: fence around garden

_budget2: 5800

Figure 6: CTR response.

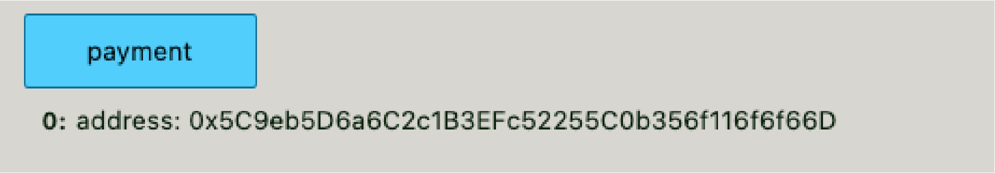
Now the CLT must make a decision about the CTR's terms, which are accessed with the `jobReturn2` function (1d). As we see in Figure 7, the CTR's estimated cost is higher than the CLT proposed budget. The CLT is thus at a decision point (1e): to accept or reject the CTR terms. If rejected (1f), the CLT may choose to re-negotiate or terminate the relationship. For this example, we assume that the CLT accepts the CTR terms on the first iteration.



```
0: string: 150x100
1: string: iron
2: string: fence around garden
3: uint256: 5800
```

Figure 7: CLT view of CTR terms.

The CLT finalizes the terms of the contract with the `jobFinalize` function (1h); this function does not require any inputs, nor does it produce outputs. It finalizes variables and deploys an instance of the PC. Proof of deployment is shown in Figure 8; the variable `payment` is the name given to the instance of the PC deployed. The address shown in Figure 8 is the address of the payment smart contract, indicating that the PC was deployed in Remix. The PC is deployed with five elements: job cost, CTR license and insurance verification, and the Ethereum account addresses for CLT and CTR.



0: address: 0x5C9eb5D6a6C2c1B3EFc52255C0b356f116f6f66D

Figure 8: PC deployed upon SoWC finalization.

4.2 Payment Contract

The PC is depicted with greater detail in Figure 9; once again, alphanumeric references refer to the notations within this figure.

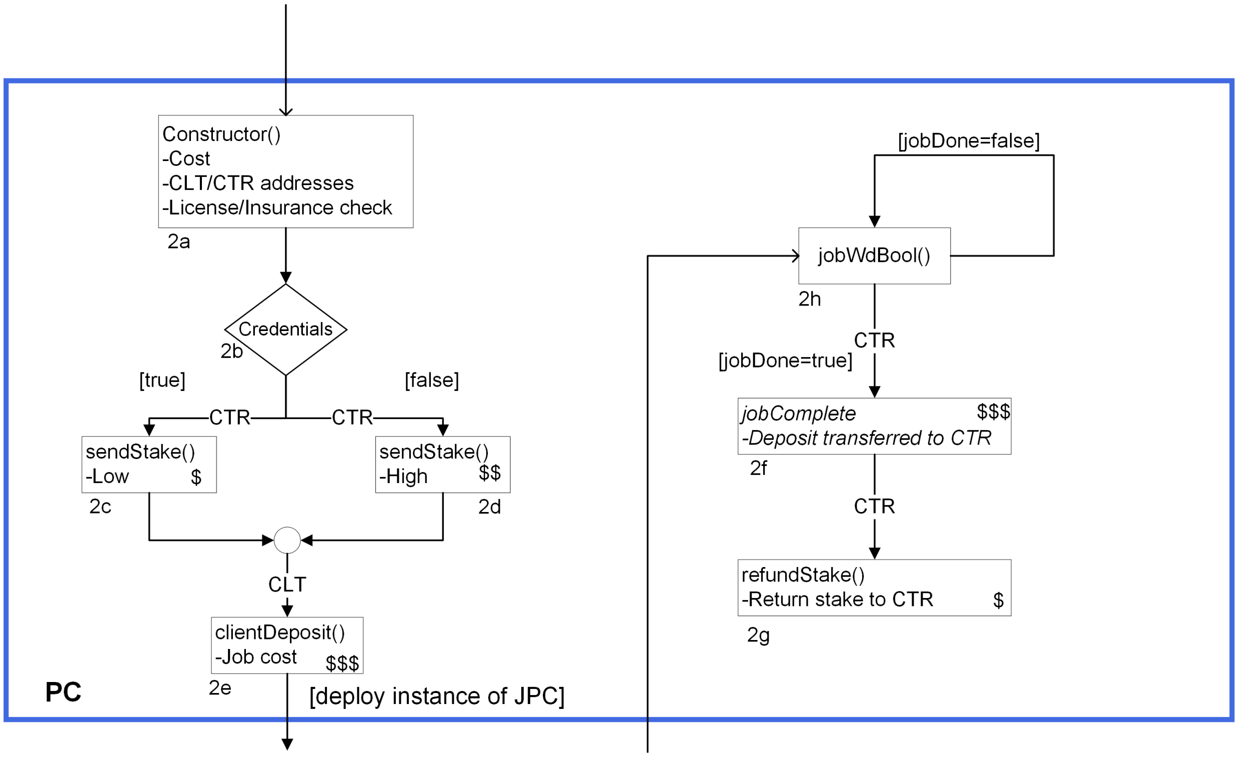
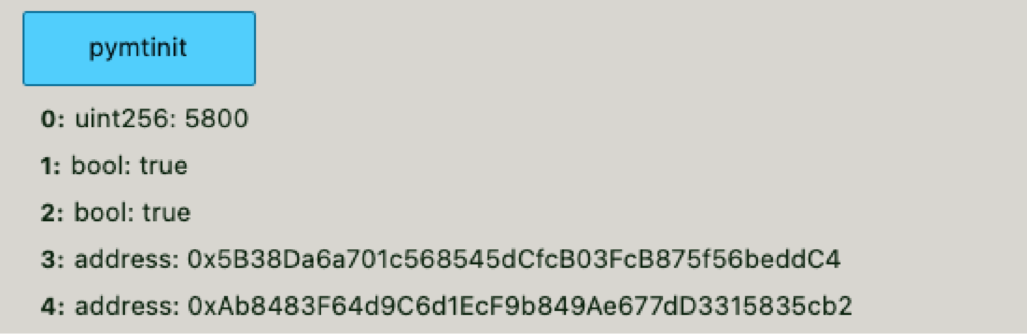


Figure 9: Payment contract.

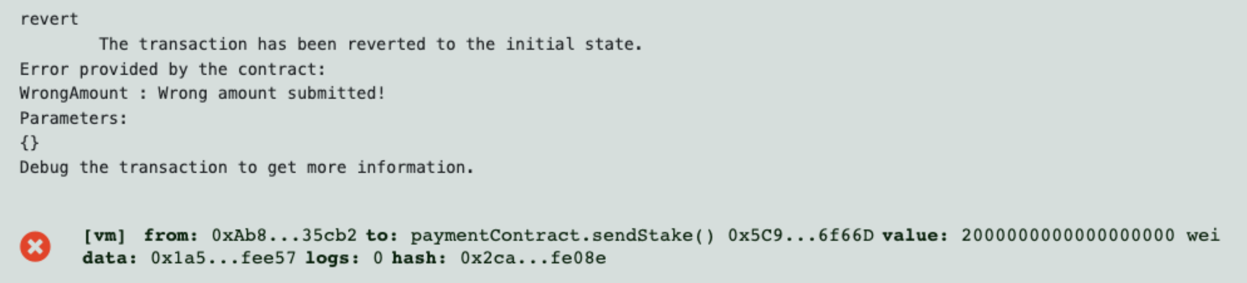
The process continues with the instance of the PC deployed with the finalization of the SoWC. Using the same example, the `pymtinit` function returns the elements of the constructor (2a). Referring to Figure 10, assume that CTR credentials are verified (corresponding to the "true" values); the 42-character addresses represent the Ethereum accounts of the CLT and CTR, respectively. The CTR stake amount depends on the results of the credentials check (2b).



```
0: uint256: 5800
1: bool: true
2: bool: true
3: address: 0x5B38Da6a701c568545dCfcB03FcB875f56beddC4
4: address: 0xAb8483F64d9C6d1EcF9b849Ae677dD3315835cb2
```

Figure 10: Elements deployed with PC.

The payments continue in a sequential manner, requiring the CTR stake first (2c/2d). In this example, the stake is lower because the CTR is licensed and insured. Recall that the stake amounts were arbitrarily set at 1 and 2 ETH for the low and high stakes, respectively. The PC has some built-in error checking; if the incorrect stake amount is submitted, an error appears (see Figure 11).

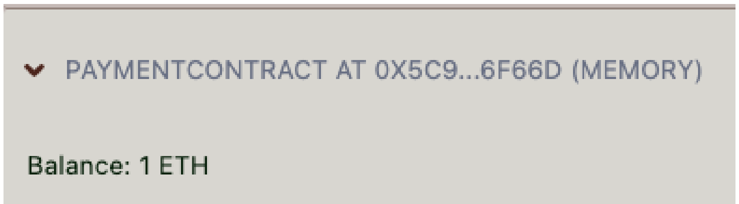


```
revert
        The transaction has been reverted to the initial state.
Error provided by the contract:
WrongAmount : Wrong amount submitted!
Parameters:
{}
Debug the transaction to get more information.


[vm] from: 0xAb8...35cb2 to: paymentContract.sendStake() 0x5C9...6f66D value: 2000000000000000000 wei
data: 0x1a5...fee57 logs: 0 hash: 0x2ca...fe08e
```

Figure 11: Incorrect stake amount submitted.

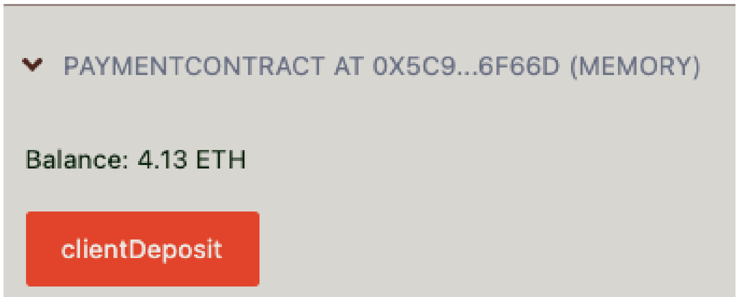
Once the correct stake amount is submitted, the CLT submits payment for the cost of the job (2e). Recall that both the CTR stake and CLT deposit amounts reside in the PC until job completion.



PAYMENTCONTRACT AT 0X5C9...6F66D (MEMORY)

Balance: 1 ETH

(a) CTR stake submitted (low amount); PC has a balance.



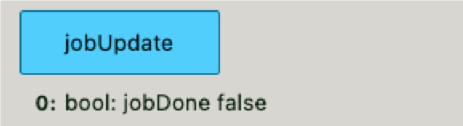
PAYMENTCONTRACT AT 0X5C9...6F66D (MEMORY)

Balance: 4.13 ETH

(b) CLT payment submitted; PC balance increases (\$5800 USD ≈ 3.13 ETH as of May 2023 (Google, 2023)).

Figure 12: PC balance after CTR and CLT payments received.

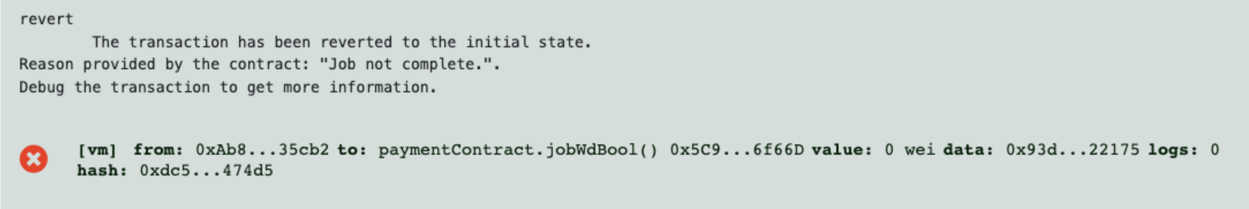
Once the CLT deposit is submitted, an instance of the JPC is also deployed with only the CLT address ((3a), see Figure 17). Refer back to Figure 9; following CLT deposit, the transfer of funds relies on a conditions-based set of criteria (2h). The `jobWdBool` function (2h) operates independently of the initial payment functions; even though it is meant for interaction post-deposit, its structure only relies on the Boolean variable `jobDone` (3c) in the JPC. Since the job is not complete, `jobDone` is initialized to false, and the PC will not transfer funds until this variable is set to true.



0: bool: jobDone false

Figure 13: Withdrawal function checks job status in the JPC.

Another example of built-in error checking is shown in Figure 14. If the CTR attempts to withdraw funds prior to job completion, an error is produced; the error results when `jobDone` is still set to false.

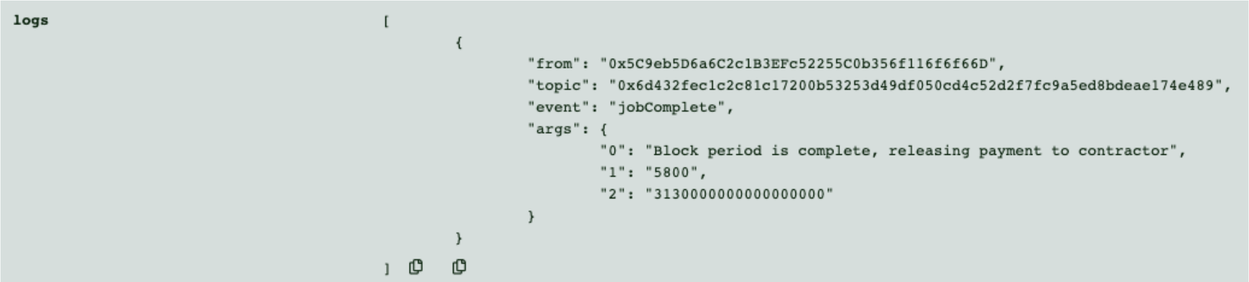


```
revert
        The transaction has been reverted to the initial state.
Reason provided by the contract: "Job not complete.".
Debug the transaction to get more information.

[vm] from: 0xAb8...35cb2 to: paymentContract.jobWdBool() 0x5C9...6f66D value: 0 wei data: 0x93d...22175 logs: 0
hash: 0xdc5...474d5
```

Figure 14: Remix error when CTR attempts to withdraw funds prior to job completion.

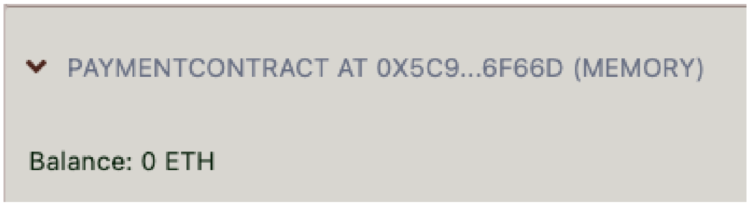
Once the job is complete and the CLT activates the `setJobDone` function within the JPC (see Subsection 4.3), the payment transfer is allowed. The CTR accesses the `jobWdBool` function, which transfers the CLT deposit from the PC to the CTR's account (2f); Figure 15 shows the Remix log outcome from this transfer.



```
logs                                          [
                                                      {
                                                              "from": "0x5C9eb5D6a6C2c1B3EFc52255C0b356f116f6f66D",
                                                              "topic": "0x6d432fec1c2c81c17200b53253d49df050cd4c52d2f7fc9a5ed8bdeae174e489",
                                                              "event": "jobComplete",
                                                              "args": {
                                                                      "0": "Block period is complete, releasing payment to contractor",
                                                                      "1": "5800",
                                                                      "2": "3130000000000000000"
                                                              }
                                                      }
                                              ]
```

Figure 15: Remix event log showing successful CTR withdrawal.

After the deposit is transferred, the CTR is then allowed to recover their stake with the `refundStake` function (2g). As shown in Figure 16, once the stake is refunded, the PC balance shows 0 ETH.



PAYMENTCONTRACT AT 0X5C9...6F66D (MEMORY)

Balance: 0 ETH

Figure 16: Result of CTR stake refund.

4.3 Job Progress Contract

The JPC is depicted as part of a larger process in Figure 17 (the layout is intentionally different here to better depict the inter-contract dependence). While the PC relies on changes to the JPC, the incorporation of digital images as proof of job progress requires some off-chain work beforehand. Additionally, a separate, dedicated IPFS contract is used to store the image hash on chain. This demonstration is adapted from the QuickNode example by Sen (2023); thus, the tools and techniques employed below mimic this guide and are not meant to be prescriptive. The circled letters in Figure 17 are provided to help understand the sequence of major events.

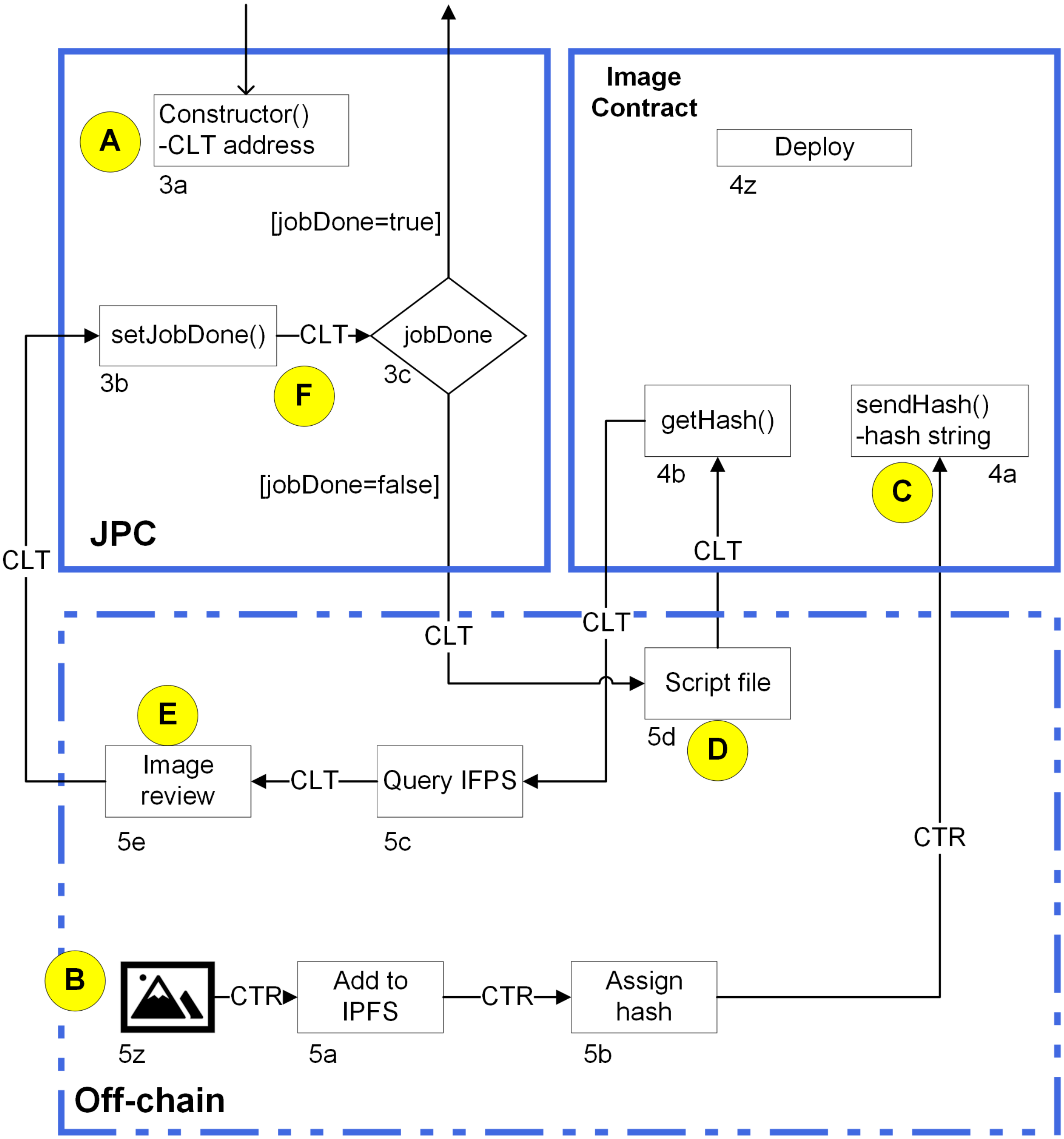
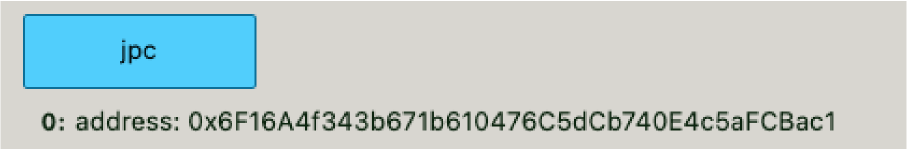


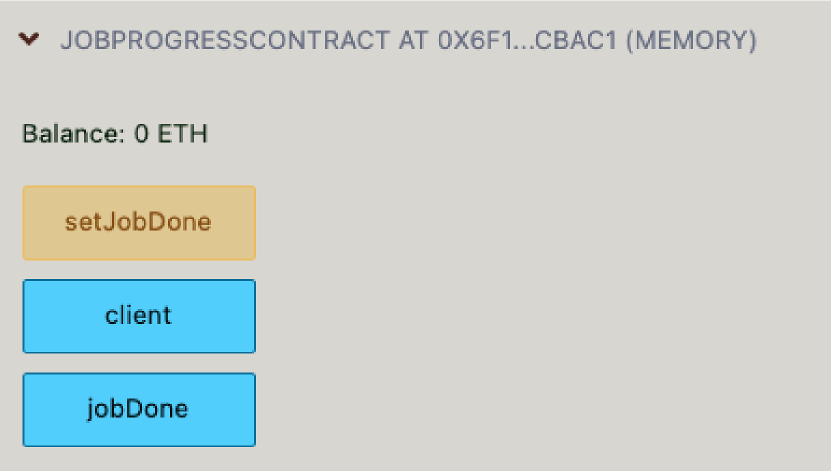
Figure 17: Job Progress contract.

As Figure 17 shows, there is minimal structure to the JPC contract; its primary purpose is to update the `jobDone` variable. The CLT performs this update with the `setJobDone` function (3b), but there are several actions that must be performed first. The CTR begins the process by capturing a digital image of the job progress (e.g. cell phone camera) and then uploading the image to IPFS (5z/5a).



0: address: 0x6F16A4f343b671b610476C5dCb740E4c5aFCBac1

(a) Instance of the JPC deployed to the blockchain.



JOBPROGRESSCONTRACT AT 0X6F1...CBAC1 (MEMORY)

Balance: 0 ETH

(b) JPC has minimal functionality.

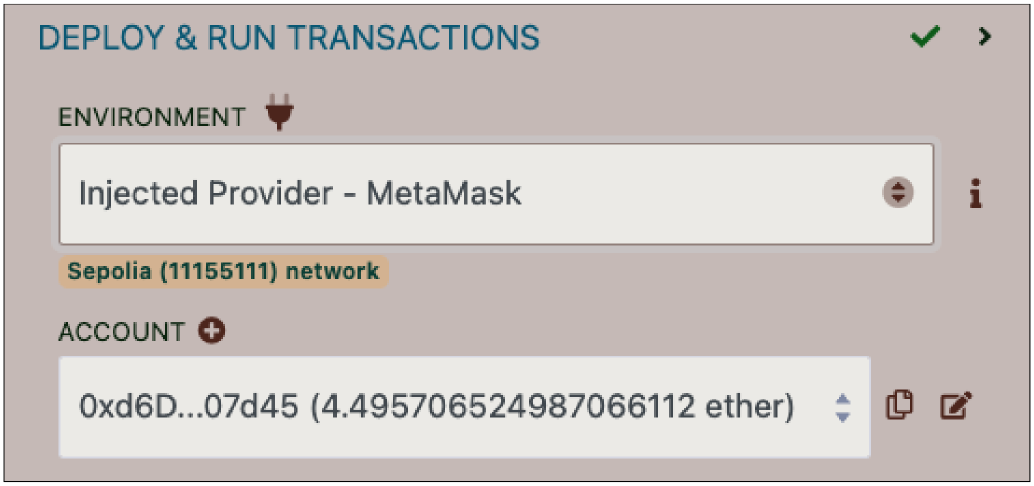
Figure 18: Initial deployment of the JPC.

The interaction with IPFS is performed through the command line interface (CLI). Once the image is secured and the IPFS repository is initialized, the IPFS daemon is started to allow online access to the network. Figure 19 demonstrates successful image upload as well as the content-based addressing feature of decentralized storage; the IPFS CID (i.e. hash) is assigned to the file (5b).

```
added QmY38jhkYsUBbFX4FyQ2gTk61DPmh1AovGKpvZ8HWbcoXh blackfence.jpg
 2.16 MiB / 2.16 MiB [=========================================] 100.00%
```

Figure 19: Adding the image to IPFS.

The CTR then compiles and deploys the (IPFS) image contract (4z), but in this example the deployment occurs on the Sepolia live testnet;[3] Remix is still used as the compiler. Note in Figure 20 that Remix is connected to the CTR's Metamask digital wallet (Metamask holds real and test ETH). The testnet is used because of the manner in which the blockchain is queried from an off-chain script file; connection to a node provider is required.


DEPLOY & RUN TRANSACTIONS

ENVIRONMENT

Injected Provider - MetaMask

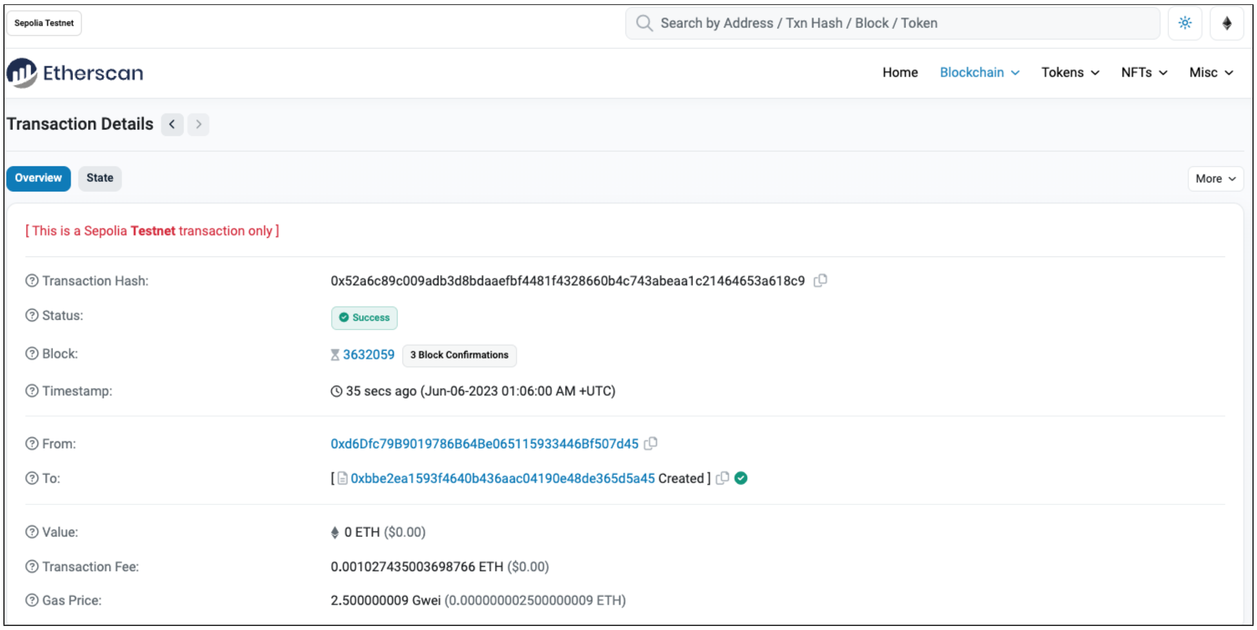
Sepolia (11155111) network

ACCOUNT

0xd6D...07d45 (4.495706524987066112 ether)

Figure 20: Deploying IPFS contract to Sepolia testnet.

Metamask requests confirmation of the transaction, and once it is validated, it is visible on Etherscan.



Transaction Details

[This is a Sepolia **Testnet** transaction only]

Transaction Hash: 0x52a6c89c009adb3d8bdaaefbf4481f4328660b4c743abeaa1c21464653a618c9

Status: Success

Block: 3632059 3 Block Confirmations

Timestamp: 35 secs ago (Jun-06-2023 01:06:00 AM +UTC)

From: 0xd6Dfc79B9019786B64Be065115933446Bf507d45

To: [0xbbe2ea1593f4640b436aac04190e48de365d5a45 Created]

Value: 0 ETH ($0.00)

Transaction Fee: 0.001027435003698766 ETH ($0.00)

Gas Price: 2.500000009 Gwei (0.000000002500000009 ETH)

Figure 21: Confirmation of IPFS contract deployment to Sepolia testnet.

Now that the contract is deployed, the CTR copies the IPFS assigned hash of the image, and uploads it to the `sendHash` function of the IPFS contract (4a). Note that the hash in Figure 22 matches the hash from Figure 19. Once the hash is stored and the state is updated, the CLT begins work to retrieve the image.

---

[3] Deployment on a live testnet implies a gas cost, whereas before we operated in a development sandbox.

Deployed Contracts

IPFS AT 0X1CA...25A82 (BLOCKCHAIN)

Balance: 0 ETH

sendHash: QmY38jhkYsUBbFX4FyQ2gTk61DPmh1AovGKpvZ8HWbcoXh

```
0: string: QmY38jhkYsUBbFX4FyQ2gTk61DPmh1AovGKpvZ8HWbcoXh
```

Figure 22: Uploading image hash to IPFS contract.

In this example, Javascript is used to query the network for the hash and view the image, but this could be performed in alternate ways. Additionally, a node client is required for interaction with the Ethereum blockchain; the client software ensures that we connect to the latest version of the chain. Some of the major clients as of 2023 include Geth, Nethermind, and Besu (Ethereum, 2023*a*); this example uses an endpoint from QuickNode. In general, the script file (5d) does the following:

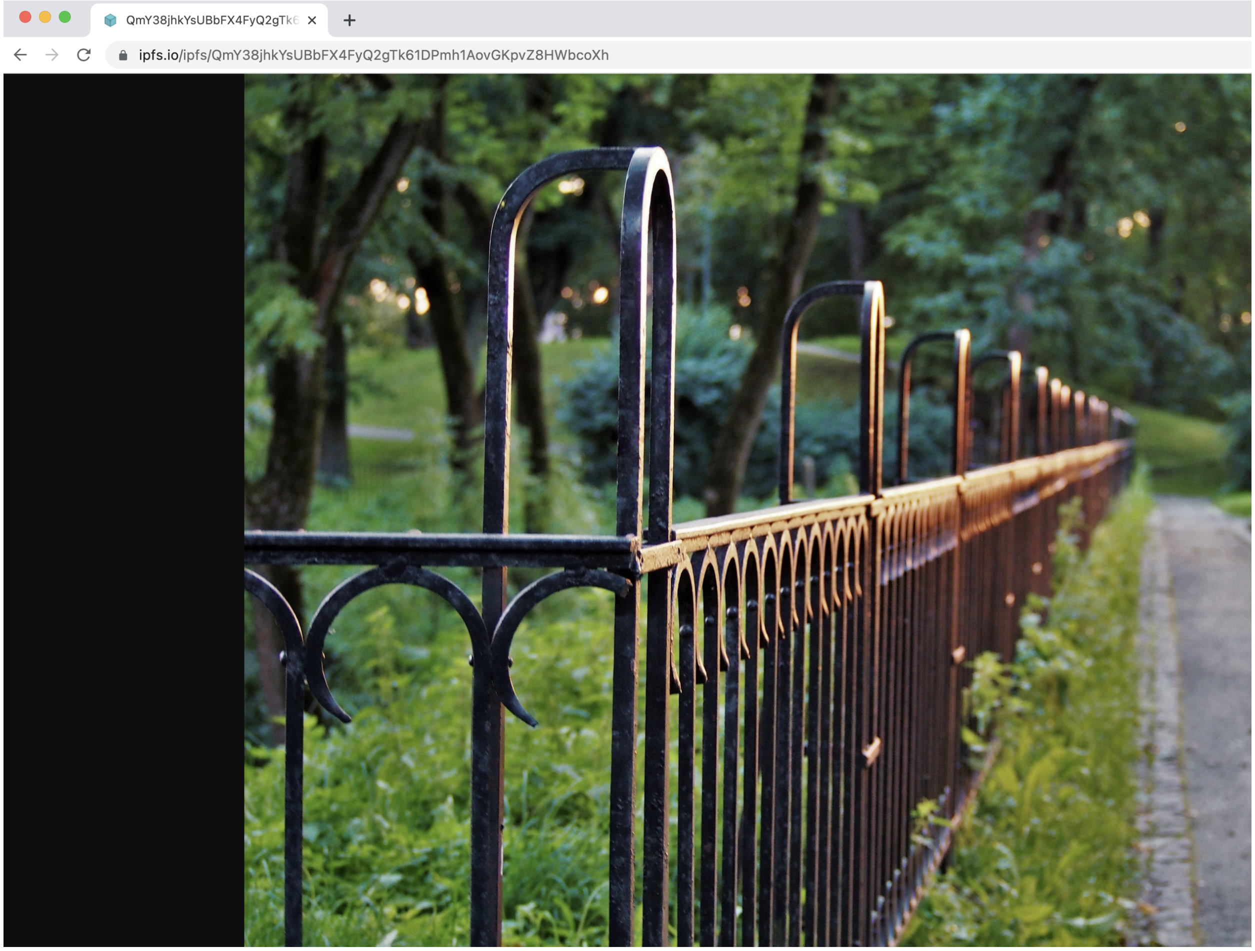
- Connects our node to the chain;
- Finds the address of the deployed IPFS contract;
- Creates an instance of the IPFS contract;
- Accesses the `getHash` function within this instance to retrieve the image hash (4b);
- Outputs a secondary script file and HTML file, the latter of which includes a link to the image in IPFS (5c/5e).

The HTML file includes a basic design that is viewed in browser. When clicked, the link *View Image* sends a browser request to IPFS for the image hash. If found, the image is returned in the browser. These descriptions are portrayed in Figure 23.

Recover Image

View Image

(a) HTML file with link.



(b) Image returned in a browser from IPFS.

Figure 23: Image retrieval with IPFS (used as per Tedder (2017)).

5 Discussion

The previous example demonstrates the use of smart contracts to automate the negotiation, verification, and payment process of a small-scale contracting job. We borrow structure from research on the incorporation of blockchain technology into the LoC process because of similar challenges between participants. The primary aspect addressed here is a lack of trust; with a one-time interaction where participants are unfamiliar with the other party, uncertainty exists about payment, contractor performance history, and job satisfaction. With the transfer of traditional contract terms to a blockchain network, the job conditions are immutable and auditable for future review. Additionally, participants must both be tied to a financial account to prove the availability of funds; these funds are held within the smart contract until certain conditions are met. Furthermore, the contractor is required to show proof of job progress, namely in the case of a remote job without client physical interaction. These aspects, in combination, help to overcome the problem of trust by removing some of the common pain points of the hiring process.

The other aspect dealt with here is enforcement. Both parties have the potential to seek closure in this process. For the contractor, he/she would like timely payment in accordance with the terms of the contract. In the current system, a contractor likely has more accessible means to enforce payment. A contractor can involve legal action through the courts (and maybe more financially stable to do so), but they can also submit for a lien on the client's property until paid. With this research, the contractor is assured that the client has funds because payment is submitted to the smart contract before work begins. On the other hand, the client's options are limited. Legal action against a contractor is costly, and depending on the length of the case, these fees may exceed the cost of the actual job. As we see in the case of NC, not even the state licensing board is much help for a client seeking action against a contractor. This research requires several things from a contractor: a stake to commit the contractor to the job, making it more likely that the contractor will finish the job (e.g. similar to an ante in poker, or a stake by a validator in a PoS consensus mechanism); proof of job progress in the form of digital evidence. Both of these elements work together to give greater control of the hiring process to the client.

Finally, we explain the rationale for design specifics within each of the smart contracts. The three contract design was inspired by the work of Chang, Chen and Wu (2019).

5.1 SoWC Justifications

Recall that the purpose of the SoWC is to capture the terms of the job, akin to the terms found in a traditional written contract. The contract is deployed including the GPS location of the proposed job, in the case that the CLT is remote and cannot be present onsite with the CTR. This aspect can also provide some cost savings in terms of time. With the dedicated functions for CLT and CTR input, these are analogous to the traditional negotiation between parties when determining the scope of a job. Instead of trying to recall a verbal discussion or having to take handwritten notes, all inputs are recorded to the chain when the transactions are submitted. Even if the negotiation process requires multiple iterations, all of this history is recorded, which may be useful in future discussion. The CLT's finalization of the contract terms replaces the traditional signed, written contract, and the agreement is visible to all who have access to the chain.

Finally, the intent of the CTR credential verification is to add extra peace of mind for the CLT. If a CTR is licensed and insured, then the CLT has some reason to believe that the CTR will perform honest work. Granted, verified credentials are not a guarantee of complete job satisfaction; with this verification process built in to the blockchain architecture, the CLT saves time and energy investigating CTR backgrounds.

5.2 PC Justifications

The PC is the most detailed of the three contracts. Once again, the purpose of the stake is to deter the CTR from misbehavior. This idea was borrowed from the PoS consensus mechanism, which requires blockchain validators to submit a stake of at least 32 ETH ($\approx \$58,962$ USD as of June 2023, (Ethereum, 2023*b*, Google, 2023)) just to be eligible for block control. The same idea is present, that this amount of money is rather detrimental if lost. Thus, the CTR submits a stake before all other payments; the arbitrary amounts of 1 and 2 ETH are not prescriptive, but they are intended to align with the small-scale nature of this contracting job. Using NC as an example, a small job under $30,000 would not require a CTR license; hence, the stake should not be that high either. However, the stake is potentially in the same neighborhood as the actual job cost, which could serve as a total job loss for the CTR in the case of dishonesty.

In this base case, we force the CLT deposit to be equivalent to the total job cost. Other payment options are not explored here, but this can be adjusted as needed. Obviously, the purpose of CLT payment is to ensure that the CTR receives a return for their service, but we hold the funds in the contract account to prevent early access. This method of blocking funds was adapted from Ahmadisheykhsarmast and Sonmez (2020). The payment transfer process is presented in two avenues: conditions-based and time-based. For a small job like the one presented here, it is reasonable to assume that the project may last several hours or perhaps up to 48 hours. Some participants may feel more inclined to use a time-based withdrawal, i.e., funds are transferred to the CTR after a certain time period. This is meant to guard against a CLT stalling, as the conditions-based option is under CLT control. These two options can also work in concert with one another, where the time-based option can serve as a backup.

Payment release sequencing is also intentional. The CTR receives job funds before recovering their stake, serving as one last check that the job was completed. Additionally, the CTR owns the responsibility of the stake recovery, preventing the CLT from accessing those funds. CTR dishonesty is not explored in this base case; therefore, the stake is always returned. However, with future cases that do involve cheating, the stake is not refunded and is instead burned.

5.3 JPC Justifications

While the JPC is the lightest of all three contracts, it is also the most intertwined. Its main purpose is to capture the CTR proof-of-progress, targeted at the scenario where the CLT is remote. Also, the JPC variable `jobDone` is automatically checked by the PC, and this is intentional to lessen the steps required by the participants. The alternative option here is to treat each contract in a vacuum, but this is error-prone. Thus, the only action in the JPC is for the CLT to mark the job as complete. Again, since we assume complete honesty in this case, the job always reaches completion.

With the off-chain work, the design is adapted from the tutorial provided by the QuickNode team (Sen, 2023). Thus, the use of Javascript to query the network is modeled after that design. However, any object-oriented language with the appropriate library to connect to Ethereum may be used, e.g., Python, C++, and Java. As mentioned previously, the cost to store large files such as images on the blockchain is prohibitive, so a less expensive storage solution is implemented. The cost instead to store a string containing the image hash is comparatively much cheaper. Also, the choice of IPFS for decentralized storage is not prescriptive; there are other distributed file system protocols available, e.g., Swarm, Arweave, Storj, etc.

6 Conclusions and Future Work

Blockchain research is needed in other application areas to explore the potential for distributed ledger technology usage. Small-scale contracting is one area where trust and payment security can become problematic for all parties involved, but blockchain technology may assist in overcoming these challenges. This research draws upon existing frameworks to create a structure and process for the CLT and CTR interaction, albeit with restrictive assumptions.

A major limitation of this research includes the examination of the environment without complications, i.e., all parties are honest, there is no attempt to deceive, and there are no external interruptions to the process, such as weather, time, or supply issues. Blockchain could accommodate some of these nuances, but this requires extensions to our model and can be addressed in future research. Additionally, this process requires participation from both parties in on-chain and off-chain methods; buy-in to this usage at the small scale is likely to be more difficult to achieve. Coupled with this difficulty is a consideration of usage costs, which are significant factors for small-scale users.

We also identify a few areas in which this concept can be expanded.

- Choice/Competition: Just like a borrower has a choice of lender, a client has a choice of contractor. Part of a client's research would typically include other user reviews, proof of the contractor's previous work, and a cost comparison. However, costs are not always available, as some contractors have agendas for why they keep pricing secret. With a public or consortium blockchain network, this system could be used by a client to view contractor history, including the jobs performed and the costs charged for service, so that the client can make an informed decision. With the consortium chain, a trusted group of participants controls the chain and can manage its rules, similar to how a delivery service like DoorDash manages what information it pulls from a restaurant and what information it allows customers to see. Furthermore, a managed chain could include a classification of contractors—those who are upfront and transparent about their history and those who are closed. Similarly, connections from the chain to reputable review services like Google, Yelp, Better Business Bureau (BBB), etc., could contribute to this knowledge base.

- Digital Provenance/Data Provenance: This includes the lineage, ownership, and truthfulness of a digital file (National Library of Medicine, 2023). In essence, to have data provenance is to provide a set of digital breadcrumbs. With regard to the digital image used in this research to verify job progress, more work is needed to verify the accuracy of the image but also to ensure that the metadata has not been altered. Some companies are already working in this research space, developing solutions to address the issue of digital authenticity. For example, Truevision has a *Vision* app that users take photos with on their phone; their algorithms can seal a set of metadata that is required by the app and also detect image manipulation. Truevision also advertises integration with existing systems through an Application Programming Interface (API), so the potential exists to align this with the smart contract process.
- Secondary Verification: The previous topic lends itself to a host of secondary verification tools that could be employed to examine job progress. At higher echelons, satellites and drones are already used for this purpose, but autonomous vehicles equipped with cameras are also another option, as well as using RFID tags to verify location.
- Language Structure: As stated previously, nothing here is prescriptive in terms of modeling. In particular, there is room for expansion on the job requirements phase, perhaps in a more standard form like JSON or XML. Also, various other scripting languages may be used to interact with the blockchain, such as Python.
- Application: The full capability of this technology is realized through integration in a dApp. This smart contract application is just a small part of the complete user experience, which could incorporate mobile phones connected to payment accounts.

References

Abdul-Rahman, H., Kho, M. and Wang, C. (2014), 'Late Payment and Nonpayment Encountered by Contracting Firms in a Fast-Developing Economy', *Journal of Professional Issues in Engineering Education and Practice* **140**(2). `https://doi.org/10.1061/(ASCE)EI.1943-5541.0000189`.

Ahmadisheykhsarmast, S. and Sonmez, R. (2020), 'A smart contract system for security of payment of construction contracts', *Automation in construction* **120**. `https://doi.org/10.1016/j.autcon.2020.103401`.

Alderfer, C. P. (1969), 'An empirical test of a new theory of human needs', *Organizational Behavior and Human Performance* **4**(2), 142–175. `https://www.sciencedirect.com/science/article/abs/pii/003050736990004X`.

Antia, K. D. and Frazier, G. L. (2001), 'The Severity of Contract Enforcement in Interfirm Channel Relationships', *Journal of Marketing* **65**(4), 67–81. `https://doi.org/10.1509/jmkg.65.4.67.18385`.

Asgaonkar, A. and Krishnamachari, B. (2019), Solving the Buyer and Seller's Dilemma: A Dual-Deposit Escrow Smart Contract for Provably Cheat-Proof Delivery and Payment for a Digital Good without a Trusted Mediator, *in* '2019 IEEE International Conference on Blockchain and Cryptocurrency (ICBC)', IEEE, pp. 262–267. `10.1109/BLOC.2019.8751482`.

Bai, X., Sheng, S. and Li, J. J. (2016), 'Contract governance and buyer-supplier conflict: The moderating role of institutions', *Journal of Operations Management* **41**, 12–24. https://doi.org/10.1016/j.jom.2015.10.003.

Baker, G., Gibbons, R. and Murphy, K. J. (2002), 'Relational Contracts and the Theory of the Firm', *The Quarterly Journal of Economics* **117**(1), 39–84. https://doi.org/10.1162/003355302753399445.

Barnes, L. B. (1981), 'Managing the Paradox of Organizational Trust', Harvard Business Review. https://hbr.org/1981/03/managing-the-paradox-of-organizational-trust, Accessed 15 June 2023.

Belfo, F. (2013), 'A Framework to Enhance Business and Information Technology Alignment Through Incentive Policy', *International Journal of Information Systems in the Service Sector* **5**(2), 1–16. https://www.igi-global.com/article/content/78933.

Biswas, P. K. (2011), 'Networks of small enterprises, architecture of governance and incentive alignment: Some cases from India', *AI & Society* **26**, 383–391. https://link.springer.com/article/10.1007/s00146-011-0318-6.

Bix, B. (2012), *Contract Law: Rules, Theory, and Context*, Cambridge University Press, New York.

Brett, J. M. (2000), 'Culture and Nnegotiation', *International Journal of Psychology* **35**(2), 97–104. https://doi.org/10.1080/002075900399385.

Brett, J. and Thompson, L. (2016), 'Negotiation', *Organizational Behavior and Human Decision Processes* **136**, 68–79. https://doi.org/10.1016/j.obhdp.2016.06.003.

Butler Jr, J. K. (1999), 'Trust Expectations, Information Sharing, Climate of Trust, and Negotiation Effectiveness and Efficiency', *Group & Organization Management* **24**(2), 217–238. https://doi.org/10.1177/1059601199242005.

Cannon, J. P., Achrol, R. S. and Gundlach, G. T. (2000), 'Contracts, Norms, and Plural Form Governance', *Journal of the Academy of Marketing Science* **28**(2), 180–194. https://doi.org/10.1177/0092070300282001.

Centobelli, P., Cerchione, R., Esposito, E. and Oropallo, E. (2021), 'Surfing blockchain wave, or drowning? Shaping the future of distributed ledgers and decentralized technologies', *Technological Forecasting and Social Change* **165**, 120463. https://doi.org/10.1016/j.techfore.2020.120463.

Chan, H.-L., Choi, T.-M. and De la Torre, D. M. (2023), 'The "SMARTER" framework and real application cases of blockchain', *Technological Forecasting and Social Change* **196**, 122798. https://doi.org/10.1016/j.techfore.2023.122798.

Chang, S. E., Chen, Y.-C. and Wu, T.-C. (2019), 'Exploring blockchain technology in international trade: Business process re-engineering for letter of credit', *Industrial Management & Data Systems* **119**(8), 1712–1733. https://doi.org/10.1108/IMDS-12-2018-0568.

Chang, S. E., Luo, H. L. and Chen, Y. (2019), 'Blockchain-enabled trade finance innovation: A potential paradigm shift on using letter of credit', *Sustainability* **12**(1), 188. https://doi.org/10.3390/su12010188.

City of Cary, NC (2023), 'Faq'. https://www.carync.gov/services-publications/residential-permits-inspections/home-construction-projects, Accessed 25 May 2023.

Clohessy, T. and Acton, T. (2019), 'Investigating the influence of organizational factors on blockchain adoption: An innovation theory perspective', *Industrial Management & Data Systems* **119**(7), 1457–1491. https://doi.org/10.1108/IMDS-08-2018-0365.

De Dreu, C. K. (1995), 'Coercive Power and Concession Making in Bilateral Negotiation', *Journal of Conflict Resolution* **39**(4), 646–670. https://doi.org/10.1177/0022002795039004003.

Eisenhardt, K. M. (1989), ‘Agency Theory: An Assessment and Review’, *Academy of Management Review* **14**(1), 57–74. https://journals.aom.org/doi/abs/10.5465/amr.1989.4279003.

Ethereum (2023*a*), ‘Nodes and Clients’. https://ethereum.org/en/developers/docs/nodes-and-clients/#execution-clients, Accessed 5 June 2023.

Ethereum (2023*b*), ‘Proof-of-Stake (POS)’. https://ethereum.org/en/developers/docs/consensus-mechanisms/pos/, Accessed 6 January 2023.

Fisher, R., Ury, W. and Patton, B. (2011), *Getting to Yyes: Negotiating Agreement Without Giving In*, Penguin, New York.

Friedman, R., Anderson, C., Brett, J., Olekalns, M., Goates, N. and Lisco, C. C. (2004), ‘The Positive and Negative Effects of Anger on Dispute Resolution: Evidence From Electronically Mediated Disputes’, *Journal of Applied Psychology* **89**(2), 369. https://psycnet.apa.org/record/2004-12430-014.

Goharshady, A. K. (2021), Irrationality, Extortion, or Trusted Third-Parties: Why it is Impossible to Buy and Sell Physical Goods Securely on the Blockchain, *in* ‘2021 IEEE International Conference on Blockchain (Blockchain)’, IEEE, pp. 73–81. 10.1109/Blockchain53845.2021.00021.

Google (2023), ‘Google Finance’. https://www.google.com/finance, Accessed 31 May 2023.

Haagensen, F. and Debois, S. (2022), Incentive Alignment Through Secure Computations, *in* ‘International Conference on Business Process Management’, Springer, pp. 343–360. https://link.springer.com/chapter/10.1007/978-3-031-16103-2_23.

Heindel, T. and Weber, I. (2020), Incentive Alignment of Business Processes, *in* ‘Business Process Management: 18th International Conference, BPM 2020, Seville, Spain, September 13–18, 2020, Proceedings 18’, Springer, pp. 93–110. https://link.springer.com/chapter/10.1007/978-3-030-58666-9_6.

Ho, S.-c. (2001), ‘Executive insights: Growing consumer power in china: Some lessons for managers’, *Journal of International Marketing* **9**(1), 64–83. https://doi.org/10.1509/jimk.9.1.64.19828.

IPFS (2023*a*), ‘Content Identifiers (CIDs)’. https://docs.ipfs.tech/concepts/content-addressing/#cid-versions, Accessed 10 May 2023.

IPFS (2023*b*), ‘IPFS and the problems it solves’. https://docs.ipfs.tech/concepts/ipfs-solves/#data-sovereignty, Accessed 10 May 2023.

IPFS (2023*c*), ‘What is IPFS’. https://docs.ipfs.tech/concepts/what-is-ipfs/#defining-ipfs, Accessed 10 May 2023.

Isaac, R. G., Zerbe, W. J. and Pitt, D. C. (2001), ‘Leadership and Motivation: The Effective Application of Expectancy Theory’, *Journal of Managerial Issues* pp. 212–226. https://www.jstor.org/stable/40604345.

Johnson, S., McMillan, J. and Woodruff, C. (2002), ‘Courts and Relational Contracts’, *Journal of Law, Economics, and Organization* **18**(1), 221–277. https://doi.org/10.1093/jleo/18.1.221.

Kaplan, R. S. and Norton, D. P. (1996), ‘Using the Balanced Scorecard as a Strategic Management System’, *Harvard Business Review* . https://hbr.org/2007/07/using-the-balanced-scorecard-as-a-strategic-management-system.

Kashyap, V. and Murtha, B. R. (2017), ‘The joint effects of ex ante contractual completeness and ex post governance on compliance in franchised marketing channels’, *Journal of Marketing* **81**(3), 130–153. https://doi.org/10.1509/jm.14.0089.

Kimani, D., Adams, K., Attah-Boakye, R., Ullah, S., Frecknall-Hughes, J. and Kim, J. (2020), ‘Blockchain, business and the fourth industrial revolution: Whence, whither, wherefore and how?’, *Technological Forecasting and Social Change* **161**, 120254. https://doi.org/10.1016/j.techfore.2020.120254.

Kong, D. T., Dirks, K. T. and Ferrin, D. L. (2014), 'Interpersonal Trust within Negotiations: Meta-analytic Evidence, Critical Contingencies, and Directions for Future Research', *Academy of Management Journal* **57**(5), 1235–1255. `https://journals.aom.org/doi/abs/10.5465/amj.2012.0461`.

Kowalski, M., Lee, Z. W. and Chan, T. K. (2021), 'Blockchain technology and trust relationships in trade finance', *Technological Forecasting and Social Change* **166**. `https://doi.org/10.1016/j.techfore.2021.120641`.

Li, J., Greenwood, D. and Kassem, M. (2019), 'Blockchain in the built environment and construction industry: A systematic review, conceptual models and practical use cases', *Automation in construction* **102**, 288–307. `https://doi.org/10.1016/j.autcon.2019.02.005`.

Li, Q., Ma, M., Shi, T. and Zhu, C. (2022), 'Green investment in a sustainable supply chain: The role of blockchain and fairness', *Transportation Research Part E: Logistics and Transportation Review* **167**, 102908. `https://doi.org/10.1016/j.tre.2022.102908`.

Liu, J., Li, H., Skitmore, M. and Zhang, Y. (2019), 'Experience mining based on case-based reasoning for dispute settlement of international construction projects', *Automation in Construction* **97**, 181–191. `https://doi.org/10.1016/j.autcon.2018.11.006`.

Lv, Y., Ma, G., Lv, L., Ding, J. et al. (2022), 'The Design of Incentive Mechanism for Policy-Oriented Guarantee Institutions' Digital Transformation in China', *Discrete Dynamics in Nature and Society* **2022**. `https://doi.org/10.1155/2022/6334476`.

Malhotra, D. and Lumineau, F. (2011), 'Trust and Collaboration in the Aftermath of Conflict: The Effects of Contract Structure', *Academy of Management Journal* **54**(5), 981–998. `https://doi.org/10.5465/amj.2009.0683`.

Mayer, R. C., Davis, J. H. and Schoorman, F. D. (1995), 'An Integrative Model of Organizational Trust', *Academy of Management Review* **20**(3), 709–734. `https://doi.org/10.5465/amr.1995.9508080335`.

Meng, H., Bian, E. and Tang, C. (2019), Themis: Towards Decentralized Escrow of Cryptocurrencies without Trusted Third Parties, *in* '2019 Sixth International Conference on Software Defined Systems (SDS)', IEEE, pp. 266–271. `10.1109/SDS.2019.8768704`.

Messick, D. M. and McClintock, C. G. (1968), 'Motivational bases of choice in experimental games', *Journal of Experimental Social Psychology* **4**(1), 1–25. `https://doi.org/10.1016/0022-1031(68)90046-2`.

Mills, E. S. (1994), 'The functioning and regulation of escrow accounts', *Housing Policy Debate* **5**(2), 203–218. `https://doi.org/10.1080/10511482.1994.9521160`.

Nash, J. (1951), 'Non-cooperative games', *Annals of Mathematics* pp. 286–295. `https://doi.org/10.2307/1969529`.

National Library of Medicine (2023), 'Data Provenance'. `https://www.nnlm.gov/guides/data-glossary/data-provenance`, Accessed 23 May 2023.

NCLBGC (2023*a*), 'FAQ for Consumers', North Carolina Licensing Board for General Contractors. `https://nclbgc.org/faq-consumers/`, Accessed 28 February 2023.

NCLBGC (2023*b*), 'FAQ for contractors', North Carolina Licensing Board for General Contractors. `https://nclbgc.org/faq-contractors/`, Accessed 28 February 2023.

Niu, B., Dong, J. and Liu, Y. (2021), 'Incentive alignment for blockchain adoption in medicine supply chains', *Transportation Research Part E: Logistics and Transportation Review* **152**, 102276. `https://doi.org/10.1016/j.tre.2021.102276`.

Niu, B., Shen, Z. and Xie, F. (2021), 'The value of blockchain and agricultural supply chain parties' participation confronting random bacteria pollution', *Journal of Cleaner Production* **319**, 128579. `https://doi.org/10.1016/j.jclepro.2021.128579`.

Niu, B., Xu, H. and Dai, Z. (2022), ‘Check Only Once? Health Information Exchange Between Competing Private Hospitals’, *Omega* **107**, 102556. https://doi.org/10.1016/j.omega.2021.102556.

Noorian, Z., Zhang, J., Liu, Y., Marsh, S. and Fleming, M. (2014), ‘Trust-oriented buyer strategies for seller reporting and selection in competitive electronic marketplaces’, *Autonomous Agents and Multi-Agent Systems* **28**, 896–933. https://link.springer.com/article/10.1007/s10458-013-9243-z.

Norrman, A. and Naslund, D. (2019), ‘Supply Chain Incentive Alignment: The Gap Between Perceived Importance and Actual Practice’, *Operations and Supply Chain Management* **12**(3), 129–142. http://doi.org/10.31387/oscm0380237.

North, D. (1990), *Institutions, Institutional Change, and Economic Performance*, Norton.

Pereira, J., Tavalaei, M. M. and Ozalp, H. (2019), ‘Blockchain-based platforms: Decentralized infrastructures and its boundary conditions’, *Technological Forecasting and Social Change* **146**, 94–102. https://doi.org/10.1016/j.techfore.2019.04.030.

Poppo, L. and Zenger, T. (2002), ‘Do formal contracts and relational governance function as substitutes or complements?’, *Strategic Management Journal* **23**(8), 707–725. https://doi.org/10.1002/smj.249.

Qian, L., Wang, Y. and Yang, P. (2020), ‘Do control mechanisms always promote collaborative performance: the role of formal institutions and business ties’, *Journal of Business & Industrial Marketing* **35**(11), 1871–1886. https://doi.org/10.1108/JBIM-07-2019-0342.

Rejeb, A., Keogh, J. G., Simske, S. J., Stafford, T. and Treiblmaier, H. (2021), ‘Potentials of blockchain technologies for supply chain collaboration: a conceptual framework’, *The International Journal of Logistics Management* **32**(3), 973–994. https://doi.org/10.1108/IJLM-02-2020-0098.

Schwartzbach, N. I. (2021), An Incentive-Compatible Smart Contract for Decentralized Commerce, *in* ‘2021 IEEE International Conference on Blockchain and Cryptocurrency (ICBC)’, IEEE, pp. 1–3. 10.1109/ICBC51069.2021.9461077.

Sen, S. (2023), ‘How to integrate IPFS with Ethereum’, QuickNode. https://www.quicknode.com/guides/ethereum-development/dapps/how-to-integrate-ipfs-with-ethereum/, Accessed 10 May 2023.

Shou, Z., Zheng, X. V. and Zhu, W. (2016), ‘Contract ineffectiveness in emerging markets: An institutional theory perspective’, *Journal of Operations Management* **46**, 38–54. https://doi.org/10.1016/j.jom.2016.07.004.

Simatupang, T. M. and Sridharan, R. (2005), ‘An integrative framework for supply chain collaboration’, *The International Journal of Logistics Management* **16**(2), 257–274. https://www.emerald.com/insight/content/doi/10.1108/09574090510634548/full/html.

Tedder, M. K. (2017), ‘Black Metal Fence’. Used in reliance on fair use; use of this image complies with Pexels license rules, available at https://www.pexels.com/license/. https://www.pexels.com/photo/black-metal-fence-635405/.

Toorajipour, R., Oghazi, P., Sohrabpour, V., Patel, P. C. and Mostaghel, R. (2022), ‘Block by block: A blockchain-based peer-to-peer business transaction for international trade’, *Technological Forecasting and Social Change* **180**. https://doi.org/10.1016/j.techfore.2022.121714.

U.S. Customs and Border Protection (2023), ‘Bill of Lading document’. https://help.cbp.gov/s/article/Article-758?language=en_US, Accessed 26 September 2023.

US Department of Commerce (2023), ‘Letter of Credit’. https://www.trade.gov/letter-credit, Accessed 26 March 2023.

Van Kleef, G. A., De Dreu, C. K., Pietroni, D. and Manstead, A. S. (2006), ‘Power and emotion in negotiation: power moderates the interpersonal effects of anger and happiness on concession making’, *European Journal of Social Psychology* **36**(4), 557–581. https://doi.org/10.1002/ejsp.320.

Von Neumann, J. and Morgenstern, O. (1947), *Theory of games and economic behavior, 2nd rev.*, Princeton University Press.

Walton, R. and McKersie, R. (1965), *A Behavioral Theory of Labor Relations*, McGraw-Hill, New York.

Wang, L., Northcraft, G. B. and Van Kleef, G. A. (2012), 'Beyond negotiated outcomes: The hidden costs of anger expression in dyadic negotiation', *Organizational Behavior and Human Decision Processes* **119**(1), 54–63. https://doi.org/10.1016/j.obhdp.2012.05.002.

Williamson, O. E. (1996), *The Mechanisms of Governance*, Oxford University press.

Wong, R. S. (2014), 'Same Power But Different Goals: How Does Knowledge of Opponents' Power Affect Negotiators' Aspiration in Power-Asymmetric Negotiations?', *Global Journal of Business Research* **8**(3), 77–89. https://papers.ssrn.com/sol3/papers.cfm?abstract_id=2451204.

Wu, J. and Wang, X. (2023), 'Platform-leading blockchain adoption for traceability under upstream competition', *Annals of Operations Research* pp. 1–37. https://link.springer.com/article/10.1007/s10479-022-05147-8.

Xie, E., Yang, Z. and Reddy, K. (2019), 'Institutional environment and inter-firm collaborative relationships: an introductory perspective and ideas for future research', *Journal of Business & Industrial Marketing* **34**(6), 1264–1269. https://doi-org.prox.lib.ncsu.edu/10.1108/JBIM-07-2019-405.

Zhou, K. Z. and Poppo, L. (2010), 'Exchange hazards, relational reliability, and contracts in china: The contingent role of legal enforceability', *Journal of International Business Studies* **41**, 861–881. https://link.springer.com/article/10.1057/jibs.2010.7.

Zimbeck, D. (2014), 'Two Party double deposit trustless escrow in cryptographic networks and bitcoin', BitHalo. https://cryptochainuni.com/wp-content/uploads/BitHalo-whitepaper-two-party-double-deposit-trustless-escrow-in-cryptographic-networks-bitcoin.pdf, Accessed 21 July 2023.