DLC Markets: A non-custodial Bitcoin Derivatives Trading System

Théo Pantamis, DLC Markets

January 10, 2024

Abstract

DLC Markets[1] introduces a groundbreaking approach that enhances security and efficiency in the Bitcoin Over-The-Counter (OTC) derivatives trading process. By leveraging Discreet Log Contracts (DLCs) on Bitcoin, this model eliminates the need for trust in counterparties and ensures transparent margin calculations and collateral management. Participants can securely lock their margin funds in a DLC, mitigating counterparty risk and reducing the reliance on centralized intermediaries. This trustless margin model streamlines the OTC derivatives trading process by automating margin calls, collateral transfers, and settlement, leading to increased efficiency and reduced operational complexities. It empowers participants with full control over their assets and fosters a more transparent and resilient OTC derivatives market.

1 The state of Bitcoin and DLC

In 2017, Thaddeus Dryja, the co-creator of the Lightning Network, published the seminal paper [1] introducing Discreet Log Contracts (DLCs). DLCs are native Bitcoin smart contracts that allow to create conditional payments tied to the revelation of the discreet log contained within the Schnorr signature of an oracle. The most impactful application of DLCs is the possibility to create trustless derivative products on Bitcoin where counterparty risk is reduced to the correct certification of the Bitcoin price by the oracle. With Taproot soft-fork activation, production ready and battle-tested implementations of the Schnorr signature algorithm are finally available to be used for such application.

Since the publication of the DLCs paper, the protocol has been improved to better support oracles' price attestation, reduce on-chain footprint and computation of fraud proofs in case of oracle breach [2]. DLCs have their own specifications [3] and there are several well-maintained implementations available that make it easier to set up derivatives [4, 5]. Several products using DLCs have been released in alpha or beta [6, 7] and focus mostly on serving retail users

[1] More info: `dlcmarkets.com`

and providing a trust minimized stable-sat, a fluctuating bitcoin balance (where the smallest unit is the satoshi or “sat”, hence the name), ensuring a fixed dollar value.

Yet, the impact of DLC extends beyond the retail sphere to the realm of institutional finance. Indeed, there is a growing demand for efficient and secure trading solutions in the Over-The-Counter (OTC) derivatives markets.

Within the traditional financial markets, there are two distinct methods for trading OTC derivatives: either through a Central Clearing Counterparty (CCP) or directly with the counterparty. Each approach carries its own benefits and drawbacks. In order to establish a basic level of security and streamline the process, counterparties who wish to trade directly with one another are legally bound by an agreement, such as the ISDA agreement, for instance. For Bitcoin-based derivatives contracts, there is no equivalent to the ISDA agreement. As a result, counterparties are left with limited options and are often compelled to rely on a CCP at some point. While it is possible for them to engage in direct trading, the associated counterparty risks are substantial, making this solution unscalable. Replicating the concept of the ISDA agreement for Bitcoin derivatives is a natural step. However, our approach differs in that we seek to achieve this not through legal means, but through the technical prowess of the Bitcoin protocol.

With DLC Markets, our objective is to provide the technical infrastructure and coordination required to bring counterparty risk-free trading and hedging in institutional finance and to make Bitcoin the settlement layer of the financial world. To fulfill this ambitious goal, DLC Markets builds upon the DLC specifications introducing several innovations leveraging a central coordinator. This approach facilitates streamlined DLC setup and clearing, paving the way for a netting layer grounded in Bitcoin. This paper serves as a technical introduction to the contributions that DLC Markets brings to institutional finance.

2 Facilitate DLC setup with a coordinator

2.1 Usual setup of DLC

Thaddeus Dryja, who introduced the concept of Discreet Log Contracts (DLCs), also co-conceived the Lightning Network (LN). It is not a mere coincidence.

DLCs look like a one-time-use payment channel, where among many pre-signed off-chain transactions, only one will be confirmed on-chain. The setup of a DLC is very similar to the opening of a payment channel in the LN. After having prepared and signed off-chain transactions (called ”Contract Execution Transactions” or CETs), funds are sent to a 2-of-2 multisig. However, there are several key differences with the typical scenario of channel funding in the LN.

In the LN, the funding transaction is often one-sided, all the funds coming from one party called the initiator. This pattern contrasts sharply with DLC setups, where both parties must contribute collateral in the same funding transaction to the multisig address. Channel dual-funding [8] already presents consid-

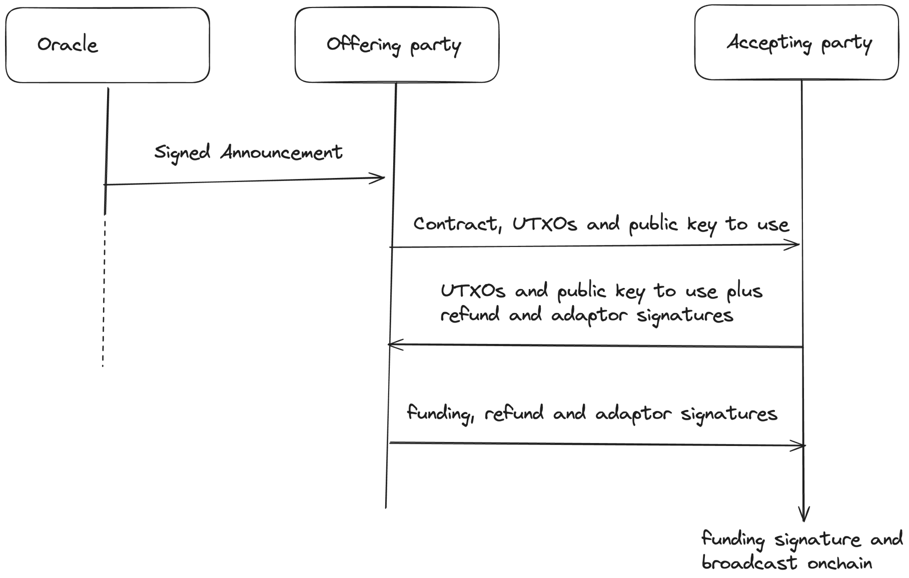
Oracle
Offering party
Accepting party
Signed Announcement
Contract, UTXOs and public key to use
UTXOs and public key to use plus
refund and adaptor signatures
funding, refund and adaptor signatures
funding signature and
broadcast onchain

Figure 1: DLC setup according to specifications. Parties exchange UTXOs and public key before signing first the CET then funding transaction

erable challenges, since it requires to maintain a transaction building and signing session that could potentially expose information about an honest party's Unspent Transaction Outputs (UTXOs, the coins in Bitcoin's ledger) and even restrict liquidity. However, in the case of DLC, dual-funding is somewhat simplified. Indeed, the amount of collateral each party must lock is predetermined and known to both parties in advance, as dictated by the contract. Thus the specified protocol to set up a DLC according to the current specifications, as depicted in figure 1, is a bit simpler.

2.2 DLC Markets solves the free option issue

Despite the advantages of DLC dual funding, several challenges persist, introducing complexities that may prove cumbersome to address. These challenges stem from the fact that parties must maintain a long build transaction and signing session with several interactions where one party may remain unresponsive for an extended duration while the other divulges information.

The most critical issue arises at the very last step of the DLC setup. The offering party provided its funding signatures, allowing the accepting party to initiate the contract on-chain. While, at preceding stages, a party could safely cancel the whole setup if it detected the other party's unresponsiveness for malicious reasons, this is much more complicated for the offering party once the accepting party can create and retain the fully signed funding transaction for itself. In such a scenario, the offering party must resort to double-spending the UTXOs it used to fund the contract in order to cancel it. The accepting party may then react by broadcasting the funding transaction and use its change output to increase the fee through Child-Pay-For-Parent, to force the offering party to stay in the contract. Although seemingly non-malicious, since no funds are lost or stolen, it is actually giving the accepting party the choice to enter in the contract much later than what the offering party initially anticipated. During the period at which the offering party did not try to double spend its own UTXOs, the accepting party actually benefited from a free-option that the offering party cannot easily price in advance [9, 10].

2.3 Coordinator-based setup of a DLC

DLC Markets addresses the intricacies associated with the free-option dilemma through a centralized coordinator. Every exchange of messages between the offering and accepting parties is channeled through this coordinator, which retains the funding signatures provided by the offering party. After validating the funding signatures of the offering party, the coordinator only sends the offering refund and adaptor signatures to the accepting party to let it verify them and securely sign the funding transaction. Once the coordinator has both funding signatures, it adds them to the funding transaction itself and broadcasts the funding transaction. In instances where the accepting party is unresponsive, the offering party retains the ability to double spend its UTXOs, prompting the coordinator to withhold finalizing the funding transaction. This affords the offering party the option to safely cancel the contract without being compelled to re-enter it should the accepting party attempt to ”exercise” its option by sending the funding signature to the coordinator.

Given that DLC Markets relies on a central coordinator to solve this issue, this infrastructure can also be leveraged to enhance the overall reliability and setup of a DLC. The coordination setup is illustrated in figure 2, wherein the coordinator stores the state of the contract of the involved parties. This enables the parties to operate stateless software that verifies exchanged data without necessitating simultaneous online presence, although timely responses to contract steps remain imperative.

One notable inefficiency arises in the initial message sent by the offering party due to the absence of knowledge regarding the UTXOs and public key that the accepting party may use to participate in the DLC. To mitigate this, we anticipate the coordinator’s continuous online presence in contrast to the accepting party’s intermittent availability. The trading party can register their wallet with the coordinator, providing the necessary public key to sign the CET. Consequently, the coordinator can directly provide the offering party with a valid selection of UTXOs of the accepting party along with a public key it can sign for in a DLC setup. This streamlined process empowers the offering party to compute and sign the refund transaction and CET right away, transmitting them to the accepting party through the coordinator. Once the accepting party comes online, it can sign the CET and refund transaction securely, evaluate the off-chain transaction signatures of the offering party, and, if in agreement,

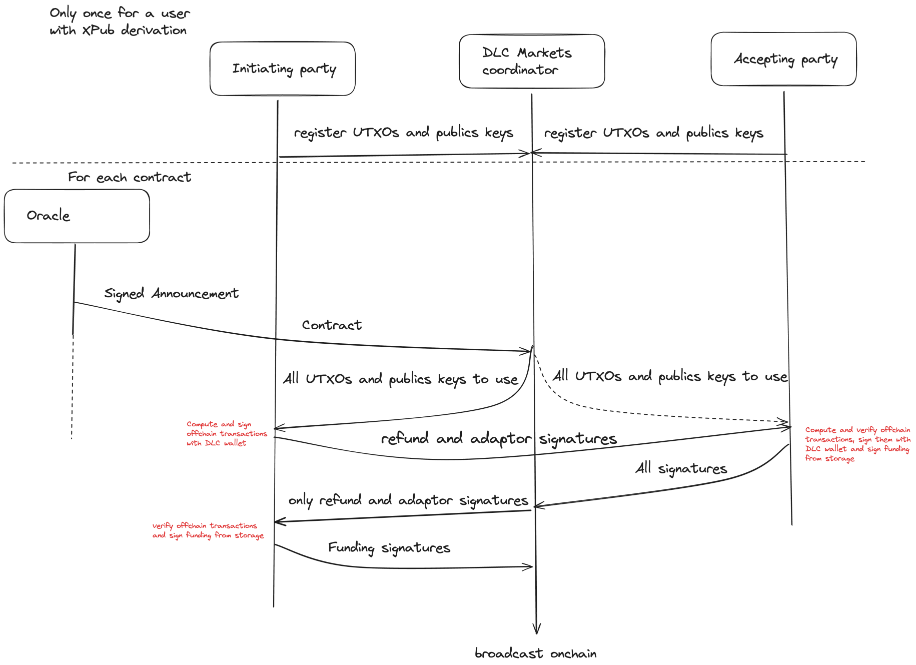
Only once for a user
with XPub derivation
Initiating party
DLC Markets
coordinator
Accepting party
register UTXOs and publics keys
register UTXOs and publics keys
For each contract
Oracle
Signed Announcement
Contract
All UTXOs and publics keys to use
All UTXOs and publics keys to use
Compute and sign
offchain transactions
with DLC wallet
refund and adaptor signatures
Compute and verify offchain
transactions, sign them with
DLC wallet and sign funding
from storage
All signatures
only refund and adaptor signatures
verify offchain transactions
and sign funding from storage
Funding signatures
broadcast onchain

Figure 2: DLC setup in DLC Markets. The number of times a party must be present is given by the number of plain arrows pointing to them. The initiating party must still be connected twice, once to get the oracle announcement, initiate the contract with the coordinator and sign off-chain transactions and another time to sign the funding transaction after validating off-chain transactions signatures. However, the accepting party only needs to be present once to accept the contract, by providing all signatures after verifying off-chain transaction signatures that the initiating party was already able to compute, thanks to the coordinator being a proxy of it at the beginning.

safely sign the funding transaction. The offering party does not itself benefit from any free option against the accepting party because the coordinator always keep the funding signatures to broadcast the DLC funding transaction on-chain for himself and should never send funding signatures to any party. The offering party, upon reconnecting, verifies the off-chain transaction signature, signs the funding transaction and sends the funding signature to the coordinator. The coordinator then aggregates the funding signatures and initiates the broadcast of the funding transaction, safely starting the DLC.

3 Extending the role of the Coordinator beyond DLC setup

3.1 Simplified On-chain Renewal of DLC

In the life cycle of a derivative, particularly in cases of high underlying volatility, the initial margin might not be adequate to offset the Profit and Loss (PL) of the contract. One potential approach to mitigate this risk is to over-collateralize the contract to account for all potential outcomes. But this is not capital-efficient. Alternatively, a solution could be to add a margin call. Whenever the PL of the contract gets closer to the margin (indicating that the margin might not be sufficient to cover the PL), the counterparty with the unfavorable position (negative PL) is requested to contribute additional margin. For a contract governed by a DLC, this modification implies the ability to alter it. While one might consider adding a new DLC on top of the existing one, this approach involves double collateralization of the same contract, which is evidently not capital-efficient.

These issues can be solved by renewing the DLC before it reaches maturity (marked by the time where the oracle attest the price and that parties may unilaterally claim their funds). Parties may reproduce a margin call procedure as illustrated in figure 3. They agree on the frequency of margin calls and the amount of collateral each party must lock into the DLC, ensuring that the counterparty has a sufficiently high probability of gaining what it is expected by unilaterally claiming the funds at maturity, chosen slightly after the next planned margin call. The collateral amount can be computed using a Value at Risk computation based on an estimation of current volatility, for example. At the planned time of margin call, one party initiates a new contract setup with the coordinator using the UTXO from the currently running DLC. The coordinator can also reuse previously registered wallet informations by each party to provide the UTXOs and address of each party to the other if additional collateral is required or cashes in a change output from the new funding transaction without any extra step.

Upon signing and confirming the funding of the new contract, the old DLC is entirely replaced. If a party has cashed in some old collateral, it becomes available for unrestricted use.

3.2 Non-custodial liquidation with disjoint union DLCs and automatic settlement

While the renewal of DLCs enables the creation of Bitcoin derivatives that are closer to traditional derivatives, a critical and necessary feature is still absent: there is no liquidation mechanism because the settlement is only possible at the oracle maturity (when it signs the Bitcoin price with the nonce in the announcement). To address this issue, we can use disjoint union DLCs [11].

In a disjoint union DLC, a single contract doesn't solely represent the settlement of the derivative; instead, it encompasses a set of contracts, includ-

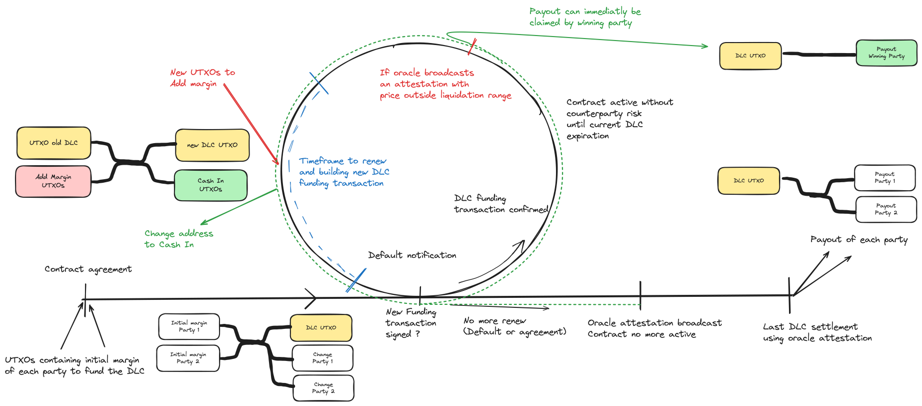

Figure 3: Time flow chart of margin call steps and hedged period for DLC with most expected transactions format. It's important to note that the contract may continue running and face liquidation even if a party declines to meet the upcoming margin call. The contract reaches its conclusion only when the oracle produces a price attestation exceeding the liquidation threshold upon sampling or at the conclusion of the last DLC. This methodology expands the flexibility of DLCs, enabling more dynamic and adaptive contract management.

ing various intermediary contracts we will refer to as ”liquidation contracts”. Similar to the settlement contract, each liquidation contract is associated with announcements, potentially from the same oracle but at different, often earlier, maturities. This is illustrated in figure 4.

A distinctive feature of a liquidation contract is that its payout curve doesn't encompass all potential outcomes that the oracle may attest. Specifically, in the case of liquidation, parties generate adaptor signatures that can only be decrypted if the oracle's attested price falls outside the liquidation threshold range. This enables each party to claim all funds in the DLC UTXO as soon as the oracle attests a price outside the range covered by the collateral of the counterparty at the maturity time of an intermediate contract as illustrated by the green settlement path in figure 3. This rapid access to liquidity allows the winning party to initiate new trades promptly if necessary and encourages the unfortunate party to initiate a margin call itself (to prevent liquidation due to a volatility spike). During this margin call, the winning party can capitalize on the profit and avoid locking its funds until settlement.

Upon liquidation or expiration of a DLC, any party can use the oracle's price attestation to generate a signed transaction, facilitating the claiming of funds based on the predetermined payout rule established during the DLC setup. Contract execution can be undertaken by the involved party or, alternatively, by the coordinator. The coordinator, having access to the adaptor signature of the Contract Execution Transaction (CET) exchanged during the DLC setup, can independently construct a fully signed transaction and broadcast it, eliminating

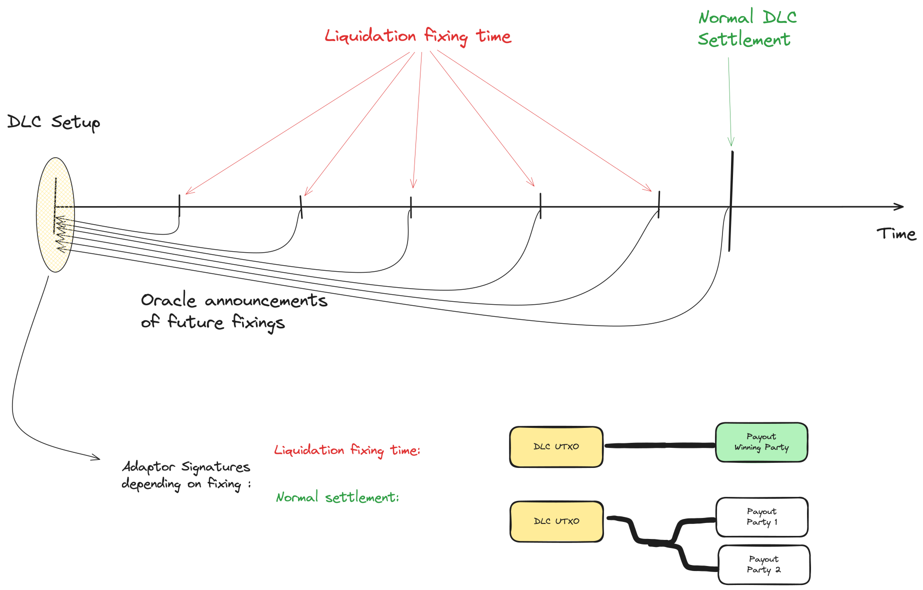

Figure 4: In a future contract with liquidations, parties decide on the fixing duration for periodic checks on potential liquidations, facilitated by an intermediary oracle attestation. As announcements are communicated beforehand, trading parties can generate adaptor signatures during setup. These signatures serve as a means to claim funds based on the liquidation contract if the attested price falls outside the covered range determined by the collateral.

the need for active involvement from any party.

Essentially, the default settlement state requires no action from any party unless they distrust the coordinator's capability to handle it. In such cases, parties can simply store the exchanged historical data, and access to the DLC private key is not even necessary if they record their own adaptor signature during setup.

3.3 Xpub, Coordinator and mining fees management

The coordinator collects a fee for its service with each funding transaction. This fee is an output of the funding transaction. It provides an incentive to the coordinator to not give a free option to one party by revealing funding signatures early since it would prevent the funding transaction to be mined if this option is not exercised. This mechanism not only compensates the coordinator for its role but also equips it with the means to enhance transaction fees using its own funds through Child-Pay-For-Parent as a service. This capability becomes particularly valuable in scenarios where the mining fees of the funding transaction might fall short of ensuring a swift confirmation. To also provide this service for settlement transactions of DLCs, which fee is fixed much sooner than confirmation during the DLC setup, the coordinator supports adding an anchor output it controls to all CETs and the refund transaction where the amount is a supplementary fee paid for the service at settlement.

The coordinator offers another crucial service: the provision of Unspent Transaction Outputs (UTXOs) and public key information to the initiating party. This facilitation streamlines the DLC setup process and bolsters the coordinator's ability to contribute to the overall efficiency of the transaction network. To do that, the coordinator requires at least that each party registers two BIP32 extended public key (a.k.a. Xpub), one for the derivation of public keys used in DLC multisig UTXOs and one to find UTXOs that can be used by a party to fund a DLC and addresses for it to receive payouts and changes. Such requirement allow to preserve privacy efforts of each party compared with always reusing the same public key and addresses. However, this reveals informations to the coordinator who is also responsible for the choice of the distribution of funds from payout and change, in the user wallet. Users who want a finer grained management of their change and payout can register more Xpubs and register which one to use for each trade or by default.

4 Conclusion

4.1 What we have already built

Building on all of the above, we can propose a secure Bitcoin derivatives trading platform powered by DLCs: DLC Markets[1]. The key features of DLC Markets are:

Direct Bilateral Trading Only trade with the counterparties you select.

Customizable Market Parameters Tailor market parameters and margin rules to fit your unique trading needs in this nascent market.

Trustless Margin Management No more counterparty risk.

As part of our ongoing development efforts, we have successfully implemented several key components:

Pythia A new oracle implementation in Rust forked from sibyls [12] using postgres DB tailored to our specific requirements. This implementation will soon be released as an open-source project, contributing to the broader development community.

dlc-core A fork of rust-dlc [4], incorporating a dedicated crate for our symmetrical coordinator-based setup. This setup not only enables on-chain liquidation and renewal of Discreet Log Contracts (DLC) but also introduces adjustments to the funding, CET and refund transactions to accommodate coordinator fees and anchors outputs.

[1] More info: `dlcmarkets.com`

wasm-dlc An open-source WebAssembly (WASM) implementation built upon the aforementioned fork. This allows DLC Markets to operate directly within web browsers, offering users an optimal and secure experience.

DLC-Markets A web interface which automates contract definition and setup, collateral management and provides real-time valuation updates, robust risk management features and a high level of customization in both the trade and the margin management to support every step of the OTC life cycle.

4.2 The future of Bitcoin finance

Our symmetric setup opens the door to incorporating more parties into a DLC. The renewal process can be transferred to other parties in cooperative scenarios and the coordinator may help to simplify the transfer of a DLC [13], paving the way for the development of a netting layer that addresses counterparty risk.

Looking ahead, we envision the integration of Taproot multisig in DLC and enhanced liquidation mechanisms with BLS attestations [14] along with easier stateless oracle management.

References

[1] Thaddeus Dryja. Discreet log contracts. `https://adiabat.github.io/dlc.pdf`, 2017. Accessed: 2023-12-01.

[2] Thibaut Le Guilly, Nadav Kohen, and Ichiro Kuwahara. Bitcoin oracle contracts: Discreet log contracts in practice. In *2022 IEEE International Conference on Blockchain and Cryptocurrency (ICBC)*, pages 1–8. IEEE, 2022.

[3] Thibaut Le Guilly, Nadav Kohen, Ichiro Kuwahara, and other contributors. dlcspecs. `https://github.com/discreetlogcontracts/dlcspecs/`. Accessed: 2023-12-01.

[4] Thibaut Le Guilly and other contributors. rust-dlc. `https://github.com/p2pderivatives/rust-dlc`. Accessed: 2023-12-01.

[5] Suredbits and the bitcoin-s developers. Bitcoin-s, feature-rich toolkit for making bitcoin and lightning applications on the jvm. `https://github.com/bitcoin-s/bitcoin-s`. Accessed: 2023-12-01.

[6] Mutiny. Mutiny wallet. `https://www.mutinywallet.com`. Accessed: 2023-12-01.

[7] 10101. 10101 finance. `https://10101.finance/`. Accessed: 2023-12-01.

[8] niftynei. interactive tx protocol. Lightning BOLTs specification repository: `https://github.com/lightning/bolts/pull/851`. Accessed: 2023-12-01.

[9] Ichiro Kuwahara. Free option problem with dlc. Scaling DLC Part2 https://medium.com/crypto-garage/scaling-dlc-part2-free-option-problem-with-dlc-ff939311954c. Accessed: 2023-12-01.

[10] Sachin Meier. Solving the free option problem for onchain dlcs inbox. https://mailmanlists.org/pipermail/dlc-dev/2023-March/000172.html. Message to the DLC devs mailing list.

[11] Kohen Nadav. Dlc v0 milestone. https://github.com/discreetlogcontracts/dlcspecs/blob/master/v0Milestone.md#disjoint-union-dlcs-done. DLC specification repository, accessed: 2023-12-01.

[12] Lava. sibyls, a numeric (and extensible) oracle implementation for bitcoin. https://github.com/lava-xyz/sibyls. Accessed: 2023-12-01.

[13] Kohen Nadav. Transferring discreet log contracts. https://suredbits.com/transferring-discreet-log-contracts/. Accessed: 2023-12-01.

[14] Varun Madathil, Sri AravindaKrishnan Thyagarajan, Dimitrios Vasilopoulos, Lloyd Fournier, Giulio Malavolta, and Pedro Moreno-Sanchez. Cryptographic oracle-based conditional payments. *Cryptology ePrint Archive*, 2022.