SADO WHITE PAPER

Self-Authenticating Decentralized Ordinals orderbook

ALPHA
v0.0.3

Birthday Research

Last Updated: 8th May, 2023

Table of Contents

Abstract

Ordinals as a mathematical concept for notation are not new. However, using them to identify and track individual satoshis for the transportation of digital artifacts on the Bitcoin blockchain is a relatively new craze. At present, the majority of trade is tracked in a Google spreadsheet. Despite the trustless nature of blockchain technology, the seamless trading of Ordinals between two unknown entities without the involvement of third-party arbitration and/or services has proven to be elusive.

By utilizing a content-addressable decentralized data storage method, it is possible to broadcast and fulfill self-authenticating orders to buy or sell Ordinals at specific prices. This can be done without needing to involve any third-parties for anything other than relaying signed transactions.

Introductions

Ordinal Theory has been around since 1883. It is a form of mathematical notation used to describe infinite sets of numbers. However, it was only in 2023 that it was applied to Bitcoin as a way of tracking individual satoshis, the lowest denomination of Bitcoin. Ordinal Theory gives satoshis individual identities and allows them to be tracked by linking them to data that is permanently stored on the blockchain within segregated-witness scripts.

Although the ORD server is relatively infant and only recently available to the public, Casey Rodarmor who started the project has been working behind the scenes on Ordinal Theory and its application to Bitcoin for some time. His terminology and explanations regarding these theories and how they create a sense of rarity for each individual satoshi is quite extensive whilst also drawing parallels to archeology and even astrology when discussing the “cycles” that occurs every 24 years when the same block is used for both halving and difficult adjustments. The first satoshi found within one of those blocks is considered one of only five legendary Ordinals.

Ordinals on their own have no resemblance to NFTs. Although clearly inspired by them in some way, Rodarmor has expressed that he is not a fan of their lack of immutability and heavy reliance on InterPlanetary File System (IPFS), which provides no guarantee of future availability in the same way that Bitcoin inscriptions do.

When creating a new Ordinal, the full inscription data is stored within segregated-witness scripts that become permanently available. Thus, instead of NFTs, Ordinals are viewed as digital artifacts. In the context of cryptocurrencies, digital artifacts refer to digital records and data inscribed to an individual satoshi on the Bitcoin blockchain as a transaction on the network.

Inscriptions on Bitcoin are true digital artifacts because they are immutable and tracked on the oldest, most decentralized, and most secure blockchain. In contrast to NFTs on other blockchains (such as Ethereum or Solana), inscriptions cannot be modified post-creation and are not plagued by off-chain data storage or smart contract audits.

When transferring a digital artifact, you are in fact only transferring an Ordinal - however, if you trace the lineage of that Ordinal, you will eventually find the original inscription containing the data or media.

In the project roadmap, Rodarmor outlines this problem of provenance as one of two issues preventing this technology from going mainstream.

The other problem he states that needs to be fixed is the ability to perform trustless trades with others on the same network. It is this problem that the SADO protocol is attempting to address, and it does so through the use of IPFS.

The IPFS is a mesh network of addressable content that is by default transient and extremely useful in cases such as shared orderbooks for trading.

Specifications

The objective of these specifications is to provide a flexible protocol that can enable multiple use-cases as efficiently as possible whilst removing the need for centralized third parties, and also providing enough core functionality to allow for future improvements.

The initial protocol will focus on four generic use cases:

- Making Sell Orders
- Making Buy Orders
- Taking Sell Orders
- Taking Buy Orders

Making Sell Orders

Required JavaScript Object Notation (JSON) for making sell orders:

- type = sell
- ts = timestamp to act as nonce
- location = the location of Ordinal being sold (txid:vout format)
- cardinals = the integer number of lowest denomination required to purchase the Ordinal
- maker = the address of the maker correlating to key used in signature

Optional JSON parameters for making sell orders:

- expiry = the block height at which the offer should no longer be valid
- satoshi = can be used to replace cardinals to indicate specific Ordinal location required
- meta = JSON string containing additional meta pertaining to Ordinal

Messages must then be signed and signature added to JSON as follows:

- signature = the signature of signing the JSON string with the sellers private key
- desc = additional field required for bech32 signature standard

Messages are then added to IPFS in order to generate a CID, which is then added within the OP_RETURN of a transaction with outputs to the Ordinal owner and any collation addresses used to publicly publish or moderate orderbooks.

The OP_RETURN must follow the following format:

sado=order:<CID_FROM_IPFS_ORDER>

Making Buy Orders

Required JSON for making buy orders:

- type = buy
- ts = timestamp to act as nonce
- location = the location of Ordinal being sold (txid:vout format)
- cardinals = the integer number of lowest denomination required to purchase the Ordinal
- maker = the address of the maker correlating to key used in signature

Optional JSON parameters for making sell orders:

- expiry = the block height at which the offer should no longer be valid
- satoshi = can be used to replace cardinals to indicate specific Ordinal location required
- meta = JSON string containing additional meta pertaining to seller

Messages must then be signed and signature added to JSON as follows:

- signature = the signature of signing the JSON string with the buyers private key
- desc = additional field required for bech32 signature standard

Messages are then added to IPFS in order to generate a CID, which is then added within the OP_RETURN of a transaction with outputs to the Ordinal owner and any collation addresses used to publicly publish or moderate orderbooks.

The OP_RETURN must follow the following format:

sado=order:<CID_FROM_IPFS_ORDER>

Taking Sell Orders

In order to buy an Ordinal, the taker must:

- Construct a partially signed Bitcoin transaction (PSBT) as specified by order
- Sign the PSBT and construct an offer object
- Add the offer object to IPFS in order to obtain an offer CID
- Relay the offer CID to the order maker

The offer should be constructed as follows:

- ts = timestamp to act as nonce
- origin = CID of original order
- offer = signed PSBT
- taker = the address of the taker correlating to key used in signature

Messages must then be signed and signature added to JSON as follows:

- signature = the signature of signing the JSON string with the takers private key
- desc = additional field required for bech32 signature standard

Offers are then added to IPFS in order to generate a CID, which is then added within the OP_RETURN of a transaction with outputs to the maker and any collation addresses used to publicly publish or moderate Ordinalbooks.

The OP_RETURN must follow the following format:

sado=offer:<CID_FROM_IPFS_ORDER>

In order to accept the BUY offer, the maker must:

- Decrypt, authenticate, sign and relay PSBT

Taking Buy Orders

In order to sell an Ordinal, the taker must:

- Construct a partially signed PSBT as specified by order
- Sign the PSBT and construct an offer object
- Add the offer object to IPFS in order to obtain an offer CID
- Relay the offer CID to the order maker

The offer should be constructed as follows:

- ts = timestamp to act as nonce
- origin = CID of original order
- offer = signed PSBT
- taker = the address of the taker correlating to key used in signature

Messages must then be signed and signature added to JSON as follows:

- signature = the signature of signing the JSON string with the takers private key
- desc = additional field required for bech32 signature standard

Offers are then added to IPFS in order to generate a CID, which is then added within the OP_RETURN of a transaction with outputs to the maker and any collation addresses used to publicly publish or moderate Ordinalbooks.

The OP_RETURN must follow the following format:

sado=offer:<CID_FROM_IPFS_ORDER>

In order to accept the sell offer, the maker must:

- Decrypt, authenticate, sign, and relay PSBT

Extensions

While the meta field within sell orders (which are verified by owners) can be used for a variety of different use cases, defining standards is beyond the scope of the current white paper.

Bidding

The application utilizing SADO can enable simple bidding by choosing to display miscalculated orders that can be processed or re-relayed should the bid/ bidding be approved by the maker.

Discovery

Passively discovering offers to buy Ordinals or to find the Ordinals being offered by specific and/or known addresses can be easily managed using SADO's broadcasting of content identifiers.

However, discovering and/or sharing unknown Ordinals requires additional third-party outputs and active scanning to known collation addresses that are defined or used by the community.

The orderbook at **sado.space** uses the following collation addresses:

- Bitcoin Testnet = mmCHfGDbKCFTpV7tH5ki9uFx8KwWJQEGMv

Roadmap

This initial specification was created in order to facilitate an initial prototype to prove a theory.

The original intent of this document was for internal use only.

Future publicized versions of this document and specification should include:

- Actual examples of real orders and offers
- Information regarding the use of the SADO JavaScript SDK
- Information regarding the use of the ORDIT web wallet