What is a Subchain?

While much has been developed in the way of digital asset creation and decentralized exchange mechanisms through meta-protocols, no solution has been applied to Bitcoin-style blockchains directly.

In this paper is presented a novel solution by which one may create multiple unspent transaction output (UTXO) data sets within a single blockchain through the application of a consensus based smart contract, allowing for the fixed transfer of value from one UTXO to another. This functionality is hereafter referred to as a "subchain".

How It Works

Modifying the Blockchain

From the Bitcoin standard, the blockchain requires two modifications: the implementation of Intelligent Transactions and the addition of an unsigned 256-bit integer (*nUTXOId*) to the CTxOut class. *While theoretically unnecessary, the inclusion of* nUTXOId *dramatically reduces the computational overhead of multiple UTXO data sets*.

Defining the New UTXO Data Set

A new UTXO data set is defined through an additional OP code, OP_UTXO. An OP_UTXO transaction is created which contains a JSON payload which defines the asset and an additional parameter which determines the action to be taken: create or modify. Only certain parameters may be changed after creation, and invalid data is ignored by the consensus mechanism.

The JSON data payload is structured as:

Version: The asset payload version number.
Name: The human readable name of the asset.
Symbol: Three to five character alphanumeric UTF-8 trading symbol.
Asset_Id: An unsigned 256-bit integer which uniquely identifies the asset to be created. Best practice is to increment on the last asset ID accepted by the network, as consensus will ignore any attempt to duplicate an existing asset ID.
Exchange_Rate: The fixed rate of exchange between the base asset (TAO) and the asset.
P2SH_Version: A byte value signifying the first character of Base58 check encoded addresses for the asset.
P2AH_Version: A byte value signifying the first character of Base58 check encoded asset exodus address.

The uniqueness of a UTXO data set is defined by the Asset ID and the Address Byte.

A Case Study

Note: to prevent potential exploits, subchain assets cannot be directly exchanged and must be converted on-chain to the base asset for the purposes of exchange. While certainly possible to automate a trustless process through an additional protocol layer executed by an

oracle node, it is not a matter of the core consensus mechanism to accommodate such functionality.

Alice desires to create an asset on the Tao blockchain for crowdfunding and operation her new DApp, and so executes a valid OP_UTXO transaction which, when fully confirmed, has a transaction hash value of:

51eaf04f9dbbc1417dc97e789edd0c37ecda88bac490434e367ea81b71b7b015.

The Intelligent Transaction JSON payload would be:

```
{
        version: 0x10001,
        name: "Alice DApp",
        symbol: "XAD",
        asset_id: 0x01,
        exchange_rate: 100.00,
        p2sh_version: 0x17,
        p2ah_version: 0x53
}
```

A bare public key address (BPKA) is a deterministic address from which tokens sent cannot be spent, however the Tao network employs a "multi-sided BPKA" for the transfer of assets between subchains. To fund the asset from TAO tokens, a valid BPKA is derived in the following manner:

Step 1 - Perform a SHA-256 hash on the transaction hash
e94643368a7aab1b18777731491671147 97e616d79752e9a28dcb0b827893a1e

Step 2 - Perform a RIPEMD160 hash on the result
fc2d99a7088a02117abb767dc2bbc24ede32eea2

Step 3 - Add the version byte for asset creation to the front, in this case 0x7F ("t")
7Ffc2d99a7088a02117abb767dc2bbc24ede32eea2

Step 4 - Perform a SHA-256 hash on the result
56e491b8aa1998ce2148f7fc9c3de378a26f3422cc3136b105d464f9f14947f1

Step 5 - Perform a SHA-256 hash on the result
a1224bd8d138402c8bd6c648c3aa8c91a6bc17cc4b6665969f49fc41352fc9f4

Step 6 - The first four bytes of the result become the checksum value
a1224bd8

Step 7 - Append the checksum to the extended RIPEMD160 hash
7Ffc2d99a7088a02117abb767dc2bbc24ede32eea2a1224bd8

Step 8 - Perform Base58 check encoding on the result to create an Exodus Address

```
tVuzxVQCiaxgcgKCx3QPkooMBTWqrpwPAw
```

The address result from the steps above (tVuzxVQCiaxgcgKCx3QPkooMBTWqrpwPAw) is in the format of a Pay-To-Asset-Hash (P2AH), or an "Exodus Address". The P2AH address is the means by which the consensus protocol signifies that the funds being received are for the creation of the asset contained within the intelligent transaction payload which can generate the hash value of the P2AH address.

Note: Funds sent to invalid P2AH addresses generated from invalid OP_UTXO transactions are permanently destroyed.

Note: As the blockchain is downloaded, P2AH addresses are cached in a local database with the matching asset definition parameters and all funding transactions for ease of use.

Let's assume Alice sends funds to the Exodus Address created above in a transaction which has the following *vin scriptSig*:

```
48304502205be1eeb9c2bbbec97958fa52109dfb4845e15882772f28433c42089aca24
8322022100af23b8dc1d2b1d9bae4bf1a31027554a7c0c2dc75ead0297617e6e4a9939
31650121022e2531ee7a16dbaf83e3554334bb6adea93af0fb2f221cb17b9ffbb587e6b
b7f
```

From this may be extracted the public key which Alice used to signed the transaction:

```
022e2531ee7a16dbaf83e3554334bb6adea93af0fb2f221cb17b9ffbb587e6bb7f
```

From this public key may be generated a unique address for the subchain for which Alice is already in possession of the private key. As the rate of exchange of the base asset to Alice's asset is already known, the tokens for the new asset are considered received to the asset-specific address associated with the public key of the funding transaction.

Therefore, Alice's funds for her new asset will be received by address AMbPZiLVTLVY82GjxHzjgR21TALPvacGjv, which corresponds to the public key from the signature of the funding transaction. This is known as an Exodus Transaction.

If Alice sent an amount of 0.1591 TAO to fund the asset, network consensus is then aware of the exchange rate and is able to credit Alice with 15.91 XAD.

Spending the Asset

As newly funded asset is transferred from one address to another, the asset ID is specified in the *nUTXOId* field of the *vout*. A new transaction will contain *vin*'s which either originate from the Exodus Address used to fund the asset or *vout*'s which bears the Asset ID of the asset received to that address, and either case is verified by network consensus to guarantee the funds indicated are spendable.

Exchanging the Asset

As each public key may serve as both a Tao address and an asset address, so may a transaction hash serve as an Exodus Address from the asset back into Tao. In this instance, Alice would send funds to an address generated using the P2AH_Version and hash from the transaction in which the asset was created. In this example, the XAD -> TAO Exodus Address would be ao3TdiHXUea8dcBPsakNRHpjVGBLYjvNZ2. As the funding transaction would bear *vout*'s with a matching *nUTXOId*, the network consensus would know this is an Exodus Transaction and properly fund the matching TAO address.