(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
31 August 2017 (31.08.2017)

WIPO | PCT

(10) International Publication Number
WO 2017/145018 Al

(51) International Patent Classification:
G06Q 20/38 (2012.01)

(21) International Application Number:
PCT/IB20 17/050861

(22) International Filing Date:
16 February 2017 (16.02.2017)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:

1603 123.9	23 February 2016 (23.02.2016)	GB
1603 125.4	23 February 2016 (23.02.2016)	GB
1607063.3	22 April 2016 (22.04.2016)	GB

(71) Applicant: **NCHAIN HOLDINGS LIMITED** [AG/AG]; Fitzgerald House, 44 Church Street, St. John's (AG).

(72) Inventors: **WRIGHT, Craig Steven;** c/o Urquhart-Dykes & Lord LLP, 7th Floor Churchill House, Churchill Way, Cardiff CF10 2HH (GB). **SAVANAH, Stephane;** c/o Urquhart-Dykes & Lord LLP, 7th Floor Churchill House, Churchill Way, Cardiff CF10 2HH (GB).

(81) Designated States *(unless otherwise indicated, for every kind of national protection available):* AE, AG, AL, AM, AO, AT, AU, AZ, BA, BB, BG, BH, BN, BR, BW, BY, BZ, CA, CH, CL, CN, CO, CR, CU, CZ, DE, DJ, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IR, IS, JP, KE, KG, KH, KN, KP, KR, KW, KZ, LA, LC, LK, LR, LS, LU, LY, MA, MD, ME, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PA, PE, PG, PH, PL, PT, QA, RO, RS, RU, RW, SA, SC, SD, SE, SG, SK, SL, SM, ST, SV, SY, TH, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, ZA, ZM, ZW.

(84) Designated States *(unless otherwise indicated, for every kind of regional protection available):* ARIPO (BW, GH, GM, KE, LR, LS, MW, MZ, NA, RW, SD, SL, ST, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, RU,

[Continued on nextpage]

(54) Title: A METHOD AND SYSTEM FOR THE SECURE TRANSFER OF ENTITIES ON A BLOCKCHAIN

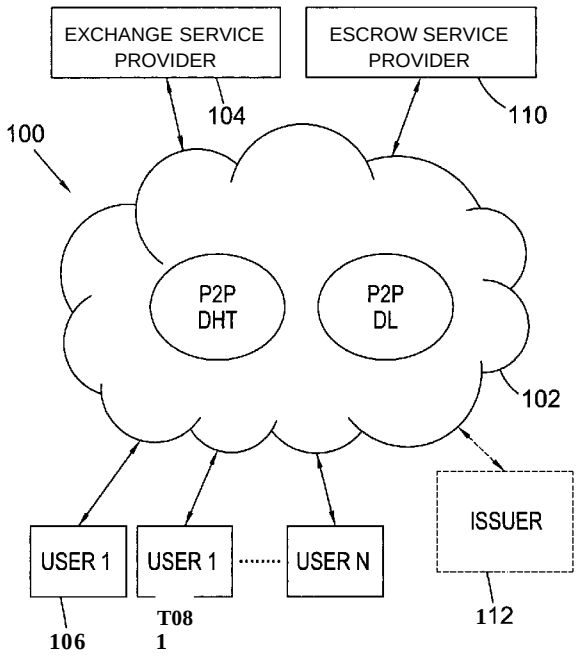

Fig. 1

(57) Abstract: The invention provides a secure method for exchanging entities via a blockchain. The invention incorporates tokenisation techniques, and also techniques for embedding metadata in a redeem script of a blockchain transaction. Embodiment(s) provide a method of: generating a first script, the first script comprising: a first set of metadata associated with a first invitation for the exchange of a first entity by a first user, the first set of metadata comprising an indication of the first entity to be offered for exchange and a first location condition for the exchange, a first user public key (PI A) associated with the first user, wherein the first user public key (PI A) is part of an asymmetric cryptographic pair comprising the first user public key (PI A) and a first user private key (VIA). The script may further comprise a first third-party public key (PIT) associated with a first third-party, wherein the first third-party public key (PIT) is part of an asymmetric cryptographic pair comprising the first third-party public key (PIT) and a first third-party private key (V1T) The method further comprises the steps of hashing the first script to generate a first script hash and publishing the first script and the first script hash on a distributed hash table (DHT).

WO 2017/145018 A1

TJ, TM), European (AL, AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HR, HU, IE, IS, IT, LT, LU, LV, MC, MK, MT, NL, NO, PL, PT, RO, RS, SE, SI, SK, SM, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, KM, ML, MR, NE, SN, TD, TG).

Published:

— *with international search report (Art. 21(3))*

"A method and system for the Secure transfer of entities on a blockchain"

Technical Field

[0001] The present invention relates to distributed, peer-to-peer ledgers and, in particular, blockchain technologies. The invention also relates in part to tokenisation and security techniques, and secure mechanisms for transferring entities and/or ownership of entities via a blockchain. It may comprise a method of performing secure transactions between different parties over a blockchain.

Background & Introduction To The Technical Area

[0002] A blockchain is a peer-to-peer, electronic ledger which is implemented as a computer-based decentralised, distributed system made up of blocks which in turn are made up of transactions. Each transaction is a data structure that encodes the transfer of control of a digital asset between participants in the blockchain system, and includes at least one input and at least one output. Each block contains a hash of the previous block to that blocks become chained together to create a permanent, unalterable record of all transactions which have been written to the blockchain since its inception. Transactions contain small programs known as scripts embedded into their inputs and outputs, which specify how and by whom the outputs of the transactions can be accessed. On the Bitcoin platform, these scripts are written using a stack-based scripting language.

[0003] In order for a transaction to be written to the blockchain, it must be "validated". Network nodes (miners) perform work to ensure that each transaction is valid, with invalid transactions rejected from the network. Software clients installed on the nodes perform this validation work on an unspent transaction (UTXO) by executing its locking and unlocking scripts. If execution of the locking and unlocking scripts evaluate to TRUE, the transaction is valid and the transaction is written to the blockchain. Thus, in order for a transaction to be written to the blockchain, it must be

i) validated by the first node that receives the transaction - if the transaction is validated, the node relays it to the other nodes in the network; and ii) added to a new block built by a miner; and iii) mined, i.e. added to the public ledger of past transactions.

[0004] Although blockchain technology is most widely known for the use of cryptocurrency implementation, digital entrepreneurs have begun exploring the use of both the cryptographic security system Bitcoin is based on and the data that can be stored on the Blockchain to implement new systems. It would be highly advantageous if the blockchain could be used for automated tasks and processes which are not limited to the realm of cryptocurrency. Such solutions would be able to harness the benefits of the blockchain (e.g. a permanent, tamper proof records of events, distributed processing etc) while being more versatile in their applications.

[0005] One area of current research is the use of the blockchain for the implementation of "smart contracts". These are computer programs designed to automate the execution of the terms of a machine-readable contract or agreement. Unlike a traditional contract which would be written in natural language, a smart contract is a machine executable program which comprises rules that can process inputs in order to produce results, which can then cause actions to be performed dependent upon those results.

[0006] Another area of blockchain-related interest is the use of 'tokens' (or 'coloured coins') to represent and transfer real-world entities via the blockchain. A potentially sensitive or secret item can be represented by the token which has no discernable meaning or value. The token thus serves as an identifier that allows the real-world item to be referenced from the blockchain. Tokenisation techniques can be used in respect of many different types of contexts where security, anonymity and cross-platform compliance are important. One such application area is financial applications, although the present invention is not limited to use in relation to financial transactions.

[0007] In this document we use the term 'blockchain' to include all forms of electronic, computer-based, distributed ledgers. These include, but are not limited to consensus-based blockchain and transaction-chain technologies, permissioned and un-permissioned ledgers, shared ledgers and variations thereof. The most widely known application of blockchain technology is the Bitcoin ledger, although other blockchain implementations have been proposed and developed. While Bitcoin may be referred to herein for the purpose of convenience and illustration, it should be noted that the invention is not limited to use with the Bitcoin blockchain and alternative blockchain implementations and protocols fall within the scope of the present invention.

Summary

[0008] The invention is defined in the appended claims.

[0009] The present invention may provide solutions for the secure control and/or transfer or exchange of an asset via a blockchain. Herein, the term "entity" may be used interchangeably with "asset". Additionally or alternatively, the invention may enable control and/or transfer of ownership of the asset. This may be a digital or virtual asset such as a smart contract, or a real-world/physical asset. The asset may be a right such as a licence or right to use or some sort of right relating to some type of property. The invention may use tokenisation techniques to facilitate this control or transfer. The invention may enable the transfer/exchange to be performed in a secure manner, incorporating the use of cryptographic keys, while not requiring any alteration of the underlying blockchain protocol. The invention may use techniques for embedding metadata in a script associated with a blockchain transaction (Tx).

[0010] The invention provides, not least: enhanced optimisation of memory usage for electronic transfers, improved security and data integrity through the use of hashing techniques, improved security through the removal of a need for a trusted third party, and enhanced anonymity of data. It may also provide improved communication

mechanisms to enable disparate or distinct parties to identify each other and/or exchange data via the novel methods and/or architectures provided by the invention. This list of advantages is not limiting or exhaustive.

[001 1] The invention may require the interaction and inter-communication of various distinct and separate computer-based resources, such as one or more user devices and a distributed computer system (blockchain) which includes computing nodes arranged to execute blockchain-related software and protocols.

[0012] The invention may provide a computer implemented method of offering an exchange of an entity. The method may comprise:

generating a first script, the first script comprising:

a first set of metadata associated with a first invitation for the exchange of a first entity by a first user, the first set of metadata comprising an indication of the first entity to be offered for exchange and a first location condition for the exchange,

a first user public key (P1A) associated with the first user, wherein the first user public key (P1A) is part of an asymmetric cryptographic pair comprising the first user public key (P1A) and a first user private key (VIA),

hashing the first script to generate a first script hash; and

publishing the first script and the first script hash on a distributed hash table (DHT).

The first script may comprise a first third-party public key (PIT) associated with a first third-party, wherein the first third-party public key (PIT) is part of

an asymmetric cryptographic pair comprising the first third-party public key (PIT) and a first third-party private key (V1T).

[0013] The first script may be a script associated with a blockchain transaction (Tx). It may be a redeem script.

[0014] The method may further comprise generating a first invitation transaction (Tx) for inclusion on a peer-to-peer (P2P) distributed ledger (i.e. blockchain), the first invitation transaction comprising an indication of a second entity to be transferred in exchange for the first entity, and the first script hash.

[0015] The P2P distributed ledger may be the Bitcoin block chain. However, it will be appreciated that any other suitable P2P distributed ledger is envisaged.

[0016] Thus, the hash of the (redeem) script may be provided within, or associated with, a blockchain transaction (Tx). This may be a P2SH transaction in accordance with the Bitcoin protocol, or another functionally equivalent transaction type in another blockchain protocol. The hash of the script may serve as a look-up key to a hash table or other storage resource. This storage resource may be a public domain repository of invitations. The storage resource may comprise the look-up key (ie the hash) and all the fields from the metadata which, in combination, define the invitation. The look-up key may be a hash of the rest of the record i.e. a hash of the concatenated metadata values. In a preferred embodiment, the metadata may comprise a pointer or other reference to the location of a contract associated with a token. The contract may be stored in a separate storage resource. The invitation (as defined by the metadata in the storage resource) may be linked to the blockchain transaction via the hash.

[0017] A number of advantages are provided by the invention, some of which are now described. Firstly, because information regarding the exchange is included in metadata securely embedded in a distributed ledger, the exchange can be carried out

securely on a peer-to-peer basis, thereby making a trusted third party unnecessary. This in turn avoids the need for a large amount of sensitive information about both parties to the exchange to be held by any third party such as a service provider, which in turn avoids the risks associated with that third party's security being compromised. This advantage is provided while also preserving anonymity of the transactions. Since the first script is hashed, it would be impracticably difficult to change the values of the metadata without causing a change in the corresponding hash values of the script. This also enables the terms of the transaction to be verified by the parties, since they are locked on a publicly available distributed ledger, which enables the integrity of the transactions to be trusted. The advantage is also provided that the first metadata can be embedded in one or more of the places available for public keys in the first script, thereby enabling nodes not suitable for processing the metadata to simply transmit the script to a further node, as opposed to blocking its progress. This in turn improves computational efficiency of the related transactions. A further advantage is provided that control data can be incorporated into the metadata, for example an access code for a barrier in the case of a token representing a ticket for a venue or a travel ticket or voucher. The metadata may also contain a pointed to an off-block repository of the details of the exchange, thereby enabling less memory and/or processing resources to be used in processing the related transactions. A yet further advantage is provided that the token can be divided, enabling two or more transaction outputs, each of which can relate to tokenised or untokenised electronically transferable digital asset.

[0018] The first entity that is offered for exchange and/or the second entity to be transferred in exchange for the first entity may comprise one or more of the following:

a) a cryptocurrency;

b) a contract;

c) goods;

d) services.

[0019] The contract may be related, at least in part, to one or more of the following:

a) a fiat currency;

b) title deeds;

c) goods;

d) services.

[0020] The first location condition may comprise one or more of the following:

a) location information indicative of a geographical location or area for the exchange of the first entity;

b) location format information indicative of a format of the location information;

c) range information indicative of a range limit on the geographical location or area relating to the exchange; and

d) range format information indicative of a format of the range information.

[0021] The method may comprise a step of:

matching the first invitation of the first script to a second script that is published on the DHT by comparing at least the first entity specified in the first script with a second entity specified in the second script,

wherein the second script comprises a second set of metadata associated with a second invitation for an exchange of a second entity by a second user, the second set of metadata comprising an indication of the second entity to be offered in exchange for the first entity,

a second user public key (P2A) associated with the second user, wherein the second user public key (P2A) is part of an asymmetric cryptographic pair comprising the second user public key (P2A) and a second user private key (V2A), and

a second third-party public key (P2T) associated with a second third-party, wherein the second third-party public key (P2T) is part of an asymmetric cryptographic pair comprising the second third-party public key (P2T) and a second third-party private key (V2T).

[0022] The method may comprise a step of determining whether the location condition is fulfilled by identifying a location of the first user and/or the second user.

[0023] The second third party may be the same as the first third party.

[0024] The method may further comprise a step of publishing the second script on the DHT, comprising:

generating the second script:

hashing the second script to generate a second script hash; and

publishing the second script and the second script hash on the DHT.

[0025] The method may comprise a step of generating a second transaction invitation for inclusion on the P2P distributed ledger, the second invitation transaction comprising an indication of the first entity to be transferred in exchange for the second entity; and the second script hash.

[0026] In one example, the method may further comprise:

generating a first exchange transaction for inclusion on the P2P distributed ledger, the first exchange transaction comprising:

the first script,

the first user public key (PI A),

the first third-party public key (PIT),

a first input provided from an output of the first invitation transaction, and

a first output indicating a first quantity of a first entity to be transferred to the second user; and

recording the first exchange transaction on the P2P distributed ledger.

[0027] Further, the method may comprise:

generating a second exchange transaction for inclusion on the P2P distributed ledger, the second exchange transaction comprising:

the second script,

the second user public key (P2A),

the second third-party public key (P2T),

a second input provided from an output of the second invitation transaction, and

a second output indicating a second quantity of a second entity to be transferred to the first user; and

recording the second exchange transaction on the P2P distributed ledger.

[0028] Alternatively, the method may comprise:

generating a first exchange transaction for inclusion on the P2P distributed ledger, the first exchange transaction comprising:

the first script,

the first user public key (PI A),

the first third-party public key (PIT),

the second script,

the second user public key (P2A),

the second third-party public key (P2T)

a first input provided from an output of the first invitation transaction,

a second input provided from an output of the second invitation transaction,

a first output indicating a first quantity of a first entity to be transferred to the second user, and

a second output indicating a second quantity of a second entity to be transferred to the first user; and

recording the first exchange transaction on the P2P distributed ledger.

[0029] The step of generating the first exchange transaction may comprise facilitating signing of the first script with the first user private key (VIA); and facilitating signing of the first script with the first third-party private key (PIT).

[0030] Similarly, the step of generating the second exchange transaction may comprise facilitating signing of the second script with the second user private key (VIA); and facilitating signing of the second script the second third-party private key (P2T).

[0031] The method may further comprise facilitating one or more of the first and second users to accept the exchange before generating or recording the first and/or second exchange transactions.

[0032] One or more of the first exchange transaction, and the second transaction may be a pay-to-script-hash (P2SH) transaction.

[0033] In accordance with embodiments of the present disclosure, there is provided a computer system for offering an exchange of an entity, the computer system comprising:

a first processing device configured to:

generate a first script comprising:

a first set of metadata associated with a first invitation for the exchange of a first entity by a first user, the first set of metadata comprising an indication of the first entity to be offered for exchange and a first location condition for the exchange,

a first user public key (PI A) associated with the first user, wherein the first user public key (P1A) is part of an asymmetric cryptographic pair comprising the first user public key (P1A) and a first user private key (VIA),

hash the first script to generate a first script hash; and

publish the first script and the first script hash on a distributed hash table (DHT).

The script may also comprise a first third-party public key (PIT) associated with a first third-party, wherein the first third-party public key (PIT) is part of an asymmetric cryptographic pair comprising the first third-party public key (PIT) and a first third-party private key (V1T);

[0034] Embodiments of the present disclosure further relate to a processor or group of processors operable to perform the above described method.

[0035] Embodiments of the present disclosure also relate to a computer readable medium having stored thereon instructions which, when executed, are operable to perform the above described method.

[0036] Embodiment(s) of the invention may comprise a technique for embedding metadata in a (blockchain) transaction, comprising the steps of:

generating a blockchain transaction (Tx) having an output (TxO) related to an asset (Bl) and a hash of a redeem script which comprises:

metadata comprising a token which is a representation of, or a reference to, a tokenised entity;

and

at least one (preferably two or more) public cryptographic keys.

The digital asset (B 1) may be a quantity of cryptocurrency eg Bitcoin. The redeem script may be provided within a locking script of the transaction output TxO. The metadata may be provided in the redeem script at a location which is designated in a blockchain protocol as a location for a cryptographic key. This provides the advantage that the metadata can be transferred without the need for any alteration to the

underlying blockchain protocol. Nodes operating the protocol will be agnostic to the use of the metadata in place of the cryptographic key.

The method may further include the step of submitting the transaction Tx to a blockchain. In effect, the cryptocurrency (Bl) may thus be locked on the blockchain in association with the token. The quantity of cryptocurrency (Bl) can only be spent (redeemed) upon provision of an unlocking script which meets the requirements of the locking script for the output TxO. In particular, a redeem script must be presented which, when hashed, matches the hash provided in the locking script of TxO. As the locking script for output TxO comprises the hash of the redeem script which in turn includes the token (in the metadata), the cryptocurrency (Bl) is associated with the token. Upon presentation of the correct unlocking (redeem) script, ownership of the cryptocurrency (Bl) may be transferred to the redeeming party or user i.e. it is spent.

[0037] Any feature(s) mentioned above in relation to one aspect or embodiment may be used in relation to any other embodiment or aspect. Any feature mentioned in relation to a method of the invention may apply equally to a corresponding, implanting system and vice versa.

Brief Description of Drawings

[0038] Embodiments of the present disclosure will now be described, by non-limiting example only, with reference to the accompanying drawings, in which:

[0039] Figure 1 is a schematic diagram of a system according to an embodiment of the present disclosure;

[0040] Figure 2 is a flow diagram of a process performed by a user of the system of Figure 1;

[0041] Figure 3 is a flow diagram illustrating a process performed by an exchange service provider;

[0042] Figure 4 is a table illustrating a metadata format for an invitation generated by an exchange service provider;

[0043] Figure 5 is a table illustrating a metadata format for an invitation generated by an exchange service provider;

[0044] Figure 6 is a flow diagram illustrating a process of matching invitations from two or more users of the system of Figure 1;

[0045] Figure 7 is a transaction table for a plurality of transactions between a plurality of parties to the system of Figure 1;

[0046] Figure 8 is a transaction diagram illustrating transactions between parties to the system of Figure 1;

[0047] Figure 9 is a transaction table for a plurality of transactions between a plurality of parties to the system of Figure 1;

[0048] Figure 10 is a transaction table for a plurality of transactions between a plurality of parties to the system of Figure 1;

[0049] Figure 11 is a flow diagram illustrating a process performed by an exchange service provider;

[0050] Figures 12 to 15 show tables illustrating metadata formats for an invitation generated by an exchange service provider;

[0051] Figure 16A is a transaction table for a transaction between two parties to the system of Figure 1;

[0052] Figure 16B is a transaction table for a transaction between two parties to the system of Figure 1; and

[0053] Figure 16C is a transaction table for a transaction between two parties to the system of Figure 1.

Description of Embodiments

[0054] State of the art methods of performing common financial transactions, such as a payment into another person's bank account or a foreign currency exchange, incur cost in both transaction fees and time delay. In contrast, transactions in electronic currencies such as bitcoin can be processed at a much faster rate (i.e. minutes rather than days) and for far less cost (in the order of a few cents per transaction rather than tens of dollars).

[0055] There exists a need for a quicker and cheaper way of performing and keeping a permanent record of day-to-day transactions, both financial and non-financial. It is important to note, that the invention is not limited to use with, or advantages for, financial applications. Instead, the present disclosure generally relates to a computer implemented method and a computer system for utilising a P2P distributed ledger, such as the Bitcoin blockchain, to enable parties to offer an exchange of an entity and record the exchange of the entity which may be any type of entity of value. Exemplary methods described herein relate to the generation of a script indicative of an invitation or offer to perform an exchange of an entity. Further methods described herein relate to the enactment of the actual exchange of the entity upon matching the script to another script.

[0056] By utilising the P2P distributed ledger, such as the Bitcoin blockchain, embodiments of the present disclosure provide for a permanent record of all steps of an exchange process to be maintained. Further, each stage in the process (offer and exchange) can be secured using cryptographic locking techniques similar to those used

in transactions of cryptocurrencies. By "cryptocurrency" is meant an encrypted, electronically transferable digital asset such as, but not limited to, Bitcoin.

[0057] The exemplary methods described herein can be used to offer and exchange any type of entity. Examples of such entities include but are not limited to cryptocurrency, such as Bitcoins, fiat currencies, contracts, goods and services.

Exchanges such as Coinffeine (http://www.coinffeine.com/) which involve the use of blockchain technologies are known in the art. However, such prior art arrangements still rely on traditional models, and also rely on third-party sources, escrows, and other multi-currency non-bank accounts/processors in order to operate. These known arrangements achieve their decentralisation through their business models, rather than through technical innovations and cryptographic techniques (as per the present invention).

The present invention incorporates the use of tokenisation techniques. Contracts may be offered for exchange or exchanged by virtue of tokens.

[0058] In summary, a token is an exchangeable entity represented by/representative of a contract. The contract may take one of several forms. For example, the contract may confer a right upon the holder or denote ownership of property. The value of the token may be contractually specified and is linked to the underlying BTC amount via a 'pegging rate'. The token is exchangeable via a transaction using a cryptocurrency protocol such as the Bitcoin protocol. The Bitcoin value on the transaction acts as a token representing a rights contract in digital form. The contract itself may be stored on the transaction or may be kept in a publicly accessible location, or may be held privately by the parties to the contract depending on the particular embodiment. Where the contract is not stored on the transaction, the transaction may store a unique pointer to the contract.

[0059] Tokens may be divisible. A divisible token is one in which the value on the transaction output can be subdivided into smaller amounts which can be allocated

across multiple new tokens. Examples of divisible tokens include tokens for fiat currency or for shares in a race horse. Divisible contracts may be defined as those that specify a non-zero pegging rate. In other words, the token value is tied to the underlying bitcoin value. Alternatively, tokens may be non-divisible. A non-divisible token is a contract that specifies the holder's rights in terms of a fixed value, e.g. a contract to redeem a house or £1,000. Non-divisible tokens are therefore not linked to the value of the underlying Bitcoin.

[0060] Tokens must be digitally signed by a token issuer to be valid. The issuer may, for example, be an authority such as a Registrar of Title deeds. The issuer may issue a token to a user in return for payment. That token may then give the user the right to exercise the contract linked to the token, whether the contract represents the right to redeem fiat currency or for a service to be performed.

[0061] Examples of tokens in accordance with the above, include the following:

- A fiat currency token that is pegged to the BTC value of the transaction output by the issuer of the contract. For example "the spender of this token (bitcoin transaction) is entitled to redeem any fraction of this token for Canadian dollars (CAD) at a rate of 1 share (lOcents) for every 1000 satoshis".
- A race horse owned by several members of a syndicate.
- Any item where the ownership is by a title deed, e.g., a house or other property could be treated in this way.
- An electronic contract representing a concert ticket. This is inherently non-divisible.
- A bearer bond (non-divisible)
- A unique identifier attached to a goods/service (such as a bar code or RFID). If used, this identifier is still preferably validated by a signature of an authorised entity; without a signature it will fall into the less secure 'goods/service' category (described below).

- A contract for the right to a service to be performed. It is noted that this is not the same as the actual service itself, but only the right to have the service performed. This right can be traded. For example, a voucher from Michael's Mowing for up to 3 hours lawn mowing within the Sydney metropolitan area. The holder of this voucher (contract) can redeem it for the actual service.

[0062] Tokens must specify the value of a share, e.g., 1 share = 10 cents CAD, 1 share = 1 rupiah, or 1 share = 1% ownership of an item or property (race horse, house, etc.).

[0063] Whilst embodiments described below may refer specifically to recording transactions on the Bitcoin blockchain, it will be understood that the present disclosure may be implemented using any P2P distributed ledger. The blockchain is used below to describe examples of the present disclosure for simplicity only due to its high level of standardisation and large quantity of associated public documentation.

[0064] As is well known in the art, the Bitcoin blockchain is a transaction ledger or database which is distributed across networked nodes participating in a system based on the Bitcoin protocol. A full copy of a currency's block chain contains every transaction ever executed in the currency. Thus, a continuously growing list of transactional data records is provided. Since each transaction entered onto the block chain is cryptographically enforced, the Bitcoin blockchain is hardened against tampering and revision, even by operators of the data store nodes.

[0065] In embodiments of the present disclosure, instead of or in addition to being used in its designed function of storing a record of transactions representing payments of Bitcoins (or other cryptocurrency) from one party to another, the blockchain is used in a novel manner to enable a transfer of entities or assets between parties. The exchange transfers control and/or ownership of the digital entity from one party to another. In order to achieve this, the invention provides a mechanism for holding and recording invitations (or orders) to perform an exchange of one or more entities. The

invention thus provides a novel and advantageous communication solution which is conducted via the blockchain.

[0066] As mentioned above, any type of entity or asset is exchangeable. These may be physical, "real world" entities, or virtual, digital entities. Examples of entities which may be exchanged include cryptocurrencies such as Bitcoins, tokens (representing any type of transferrable contract), and goods and services of any type. Tokens may represent a contract conferring specified rights upon the holder to be redeemed for fiat currency (virtual banknotes), to denote ownership of property (e.g., title deeds), or grant access to an event (tickets), to name a few of many examples. Goods and services may include new or second hand products, labour (e.g., charged by the hour), complete jobs (e.g., mowing the lawn), to name a few of many examples.

[0067] Figure 1 is a network diagram of a P2P exchange system 100 according to one embodiment. The system 100 comprises a network 102 and a plurality of parties to the network. The parties include an exchange service provider 104, a first user 106, a second user 108, an escrow service provider 110, and an issuer 112. As will be described in more detail below, a combination of functions of the exchange service provider 104, the escrow service provider 110 and the issuer 112, may be undertaken by a single party. In other words, a single party could simultaneously perform the functions of each. In addition, and as will be explained in more detail below, the exchange service provided 104 and escrow service provider 110 are optional, since the invention can be carried out entirely on the P2P exchange system without the use of those service providers 104, 110.

[0068] The exchange service provider 104 provides exchange services to a plurality of users, including the first user 106 and the second user 108. The issuer 112 is optional to the network 102, as denoted by broken lines. As will be discussed in more detail below, the issuer 112 may only be required when an exchange of tokens is required.

[0069] In some embodiments, the network 102 is the Internet. Accordingly, other parties (not shown) may be party to the network 102. All parties to the network 102 are able to communicate with all other parties to the network 102. Hosted on the network 102 are a peer-to-peer distributed hash table (P2P DHT) and a peer-to-peer distributed ledger (P2P DL). It will be appreciated that some or all of the parties shown in the system 100, together with those not shown, may act as host nodes to both or either of the P2P DHT and the P2P DL.

[0070] Structure of An Invitation

An invitation may be structured include various parameters or codes. These can be used for various purposes, eg matching invitations as described in more detail below. In one or more embodiments, the following structure may be used:

Offer-type-code	
Offer-QTY-max	This is the amount of value e.g BTC that will be carried on the offerer's payment transaction. If the offerer is offering BTC than this value is simply the offerer's maximum BTC offer. If the offerer is offering tokenised currency then this value is the equivalent token value of the offerer's maximum offered currency amount (calculated based on the pegging rate specified on the tokenisation contract). The same is true if the offeror is offering a tokenised other commodity such as part ownership of a racehorse (the contract will still have a pegging rate that specifies the BTC token value for shares in the horse). If the offeror is offering a physical item then this field is ignored - but the actual bitcoin transaction will carry the minimum required amount (i.e. dust = 546 satoshis). Of course, the amount of the spend cannot exceed the total BTC value of the transaction's inputs so the offerer cannot offer more than the bitcoins owned.
Offer-QTY-min	
Offer-Item-ID	
Offer-Description	Keywords: This is a Condition that must be set if the Offer is goods/service and there is no other identifier (such as an auction site catalogue number)
Rate-min	This is the minimum rate of exchange that the offeror will accept expressed by convention as (requested units)/(offered units) Examples:

	(1) BTC offered for shares in a commodity (such as part ownership of a racehorse) rate = shares/satoshis (Note: this also applies to tokens that are non-divisible such as a concert ticket, although in this case there is only one share) (2) Token share offered for BTC rate = satoshis/share (e.g. satoshis/cent, where the tokens are for fiat amount such as CAD) (3) Token for token rate = requested-share/offered-share (e.g. cents/rupiah, where token offered represents rupiah and requested is CAD) Note that this is only a convention for the sake of consistency and convenience and will make matching easier. The rate could be just as easily be converted into requested-satoshis/offered-satoshis based on the pegging rate used on the tokenisation contract.
Rate-max	
Conditions	This will be a code to indicate up to 8 conditions that will be coded in separate metadata fields. For example: . Deadline (in Unix time) . Location (described in more detail below) The format of these metadata fields depends on the type of condition. Note that Offer-Description and Request-Description are also considered conditions. They are usually not needed as all the information required normally be can be coded into the other metadata fields. If they are needed then they need to be flagged as existing by switch on the appropriate flags in the Conditions bitfield.
Request-type-code	
Request-item-ID	
Request-description	Keywords: This is a Condition that must be set if the Request is goods/service and there is no other identifier (such as an auction site catalogue number)
Request-QTY-max	
Request-QTY-min	

[0071] One purpose of the exchange service provider 104 is to provide a gateway for users 106, 108 to place invitations or offers for an exchange of an entity on the P2P DHT and/or the P2P DL. Whilst the users 106, 108 of the network 102 may themselves place invitations on the P2P DHT and/or the P2P DL, the exchange service provider 104 provides an interface with which invitations are generated. Furthermore, the exchange service provider 104 may reduce the dangers associated with direct

handling of transactions on a distributed ledger, such as the Bitcoin blockchain (e.g., losing transactions etc.) as the skilled person will appreciate. In addition to enabling the publishing of user invitations on the P2P DHT and/or P2P DL, the exchange service provider may perform one or more of the following additional services:

- Matching invitations - As described above, an invitation may include a) details of the entities a user wishes to exchange, and b) one or more user applied options/conditions, such as a location condition, attached to the exchange. Two invitations may match when their respective entity details are mirrored and one or more of the conditions of the two invitations are compatible. In other words, a match may occur when one or more parameters or features included in a first invitation is/are also included in a second invitation. The is some commonality between one or more of the features of the invitations. An example of mirrored entity details would be when a first user (Alice) offers Bitcoins for apples and a second user (Bob) offers apples for Bitcoins. The service provider may therefore provide a matching service to match compatible invitations in order to accommodate an exchange. Matching may involve scanning the P2P DHT for one or more invitations having matching entities and/or conditions. In some embodiments, the service provider 104 may scan the P2P DHT in response to a request from a user. For example, a user may provide to the service provider 104 one or more criteria for a desired invitation. Based on the criteria provided, the service provider 104 may then search for invitations already placed on the P2P DHT which match those criteria. In other embodiments, the service provider 104 may implement a non-specific pairing algorithm which searches the P2P DHT for matching or near-matching invitations which do not relate to a specific user request. It will be appreciated that matching services may be provided by other third-party providers. One or more third-party providers may exist, whose primary purpose is to provide matching services in accordance with the above as well as match alerts as described below. In some embodiments, matching is provided by a matching service provider (MSP).

In accordance with one or more embodiments, and with reference to the table shown in the "Structure of an Invitation" section above, a matching algorithm from matching invitations between A and B such as the following may be employed:

A's Offer-type-code must match B's Request-type-code
A's Request-type-code must match B's Offer-type-code
A's Rate-Min ≤ B's Rate-max (when expressed in equivalent units)
A's Rate-Max ≥ B's Rate-min (when expressed in equivalent units)
Request-item-ID must match Offer-item-ID
A's Request-QTY-min ≤ B's Offer-QTY-max
A's Request-QTY-max ≥ B's Offer-QTY-min
A's condition(s) (if any) must be compatible with B's invitation
B's condition(s) (if any) must be compatible with A's invitation

The invention may be arranged to incorporate machine-executable rules which enforce this algorithm or a variation thereof.

- Match alerts - If a match or near-match is detected, the exchange service provider 104 may alert a user in a known manner, such as by email or via a phone or tablet app. Thus, the invention may provide a novel communication or alert mechanism.

- Generating new invitations based on matches - Where a user provides details of an invitation or offer the user wishes to place, the service provider 104 may scan the P2P DHT for one or more invitations which satisfy the conditions of the user's order. If a matching invitation is then found on the P2P DHT, the service provider 104 may generate an invitation which mirrors the identified invitation already on the P2P DHT in order to facilitate a successful match. It is noted that in order to complete the final transaction on the P2P DL, all parties to a transaction must have an invitation already published on the P2P DHT. However, not all invitations need be published on the P2P DHT. In the present example, for instance, the service provider need not publish the offer on the P2P DHT since there is no requirement for the invitation to be advertised as a

desired match has already been found. It will be appreciated, however, that the generated invitation may still be placed on the P2P DHT, for example, in case the initial match falls through.

- Executing transactions - After a pair of invitations have been successfully matched, the service provider 104 may act as a proxy to implement the final transaction. For example, on determination that two invitations provide an exact or near match, the service provider 104 may record an actual transaction, i.e. a transaction involving an *exchange of the entity*, on the P2P distributed ledger. This process may be conducted automatically without the parties express authorisation or after prompting one or more of the parties to authorise the transaction. In some embodiments, metadata in an invitation may indicate whether or not a party must be notified before an exchange is finalised.

- Verifying fulfilment of conditions - where an invitation includes a condition, such as a location condition, the exchange service provider 104 may determine whether the condition is being fulfilled before finalising the exchange. With regard to the location condition, the exchange service provider 104 may, for example, determine the current location of a user when two invitations have been matched.

- eWallet services - In addition to the above, the service provider 104 may also provide conventional eWallet services such as the holding of cryptocurrency keys etc.

[0072] A single service provider 104 is shown in the system 100 of Figure 1. However, it will be appreciated that one or more additional exchange service providers may be part of the network 102. Where more than one exchange service provider exist, users may choose an exchange service provider depending on their requirements, which may, for example, include the service provider's fee structure, location, compatibility etc. It will therefore be appreciated that, in certain circumstances, two users with matching invitations may use different exchange service providers. In such

circumstances, the users' respective exchange service providers may communicate with each other in order to facilitate matching of invitations and/or an exchange of an entity.

[0073] In addition to the exchange service providers 104, an escrow service provider 110 may be part of the network 104. The escrow service provider 110 enables a user's offer to be held (i.e. the amount offered is reserved) until a transaction is settled, or under certain conditions to cancel an order and have returned anything that was offered in an invitation. The escrow service provider 110 acts as a neutral third party, trusted by two parties of a transaction, to provide an escrow service for transactions. Thus, the system allows users participating in a final transaction to have a guarantee that the user making an offer can fulfil the offered amount (in Bitcoins or tokens).

[0074] As with exchange service provider, more than one escrow service provider may be part of the network 104. Users of the P2P exchange system 100 may also choose which escrow service provider they use, if and when they use one. In some embodiments, the services of the escrow service provider 110 may be incorporated into those of the exchange service provider 104 or vice versa. In such case, a separate escrow service provider may not be required.

[0075] In addition to the above, the system 100 may also comprise an issuer 112. An issuer 112 may be involved where the transaction involves the exchange of a token. In such circumstances, the process involves the issuer signing the token. Every transaction involving the transfer of a token may involve the issuer 112. In embodiments described herein, the issuer's signature is required in the invitation transaction, in which the token is offered and held in escrow. The issuer's signature may also be required in the exchange transaction, in which the payment of the token is made to the counterparty.

[0076] An aspect of embodiments of the present disclosure is the ability to embed metadata concerning an invitation or offer to perform an exchange of an entity in a Bitcoin transaction (or other cryptocurrency transaction) as well as the ability to embed metadata concerning an actual exchange of the entity, in a Bitcoin or other

cryptocurrency transaction. Embodiments described herein use multi-signature pay to script hash (P2SH) type transactions to enable such metadata embedding as described below.

(i) Redeem script in P2SH in general

[0077] As background, in a standard pay-to-script-hash method of the bitcoin protocol, the redeem script may take the form of:

<NumSigs PubKl PubK2 ... PubK15 NumKeys OP_CHECKMULTISIG>

where

NumSigs - is the number "m" of valid signatures required to satisfy the redeem script to unlock the transaction

PubKl, PubK2 … PubKl 5 - are the public keys that correspond to signatures that unlock the transaction (up to a maximum of 15 public keys)

NumKeys - is the number "n" of public keys (which must be 15 or less)

[0078] To redeem the above redeem script, at least a number "m" of signatures corresponding to the public keys are required. In some examples, the order of the public keys is important and the number "m" out of "n" signatures for signing must be done in sequence. For example, say that "m" is two and the number of public keys "n" is fifteen. Say that two signatures are available for use, say Sigl (corresponding to PubKl) and Sig 15 (corresponding to PubK15), the redeem script must be signed by Sigl first followed by Sig 15.

(ii) Embedding metadata in a P2SH

[0079] Metadata may be embedded in a P2SH in one or more of the 15 places available for the public keys in the redeem script.

[0080] For example, the P2SH may take the form of:

<NumSigs Metadatal Metadata2...PubKl PubK2...NumKeys OP_CHECKMULTISIG>

where

NumSigs - is the number "m" of valid signatures required to satisfy the redeem script to unlock the transaction,

Metadatal and Metadata2- each include metadata that takes place of a public key,

PubKl and PubK2 - are actual public keys, and

NumKeys - is the total number of positions taken by the metadata and the public keys (which must be 15 or less).

[0081] By placing, in a redeem script, metadata corresponding to conditions of an invitation, details of a contract associated with a token, and/or other information associated with an exchange, the hash of such information will be included in the P2P distributed ledger.

[0082] This embedding method can be summarised as:

generating a blockchain transaction (Tx) having an output (TxO) related to a portion of cryptocurrency and a hash of a redeem script which comprises:

metadata comprising a token which is a representation of, or a reference to, a tokenised entity;

and

at least one (preferably two or more) public cryptographic keys.

The tokenised entity could be a contract and/or other entity relating to the exchange. The metadata is provided in a location designated by the protocol for a cryptographic key.

[0083] Thus, the use of multi-signature P2SH bitcoin transactions in embodiments of the present disclosure offers several advantages. Firstly, it enables an invitation transaction to carry a metadata payload. Secondly, it facilitates the use of escrow services in an exchange transaction. Thirdly, where tokens are transferred in an exchange, it allows an exchange transaction to carry metadata associated with the one or more tokens being exchanged. Also, the underlying blockchain protocol is agnostic to the fact that metadata is being transmitted via the transaction. Therefore, no change is required to the blockchain protocol to convey this information.

[0084] Metadata may include a description or keyword describing an offer or request in an invitation transaction. Metadata may also include conditions associated with the invitation. For example, a location condition may be attached to the invitation which may specify a geographical location at which the exchange of the entity must take place. In addition to the location condition, a deadline condition may further be attached to the invitation. In this regard, a cancellation transaction may be generated which spends the same Bitcoin amount and contains a locktime representing the deadline for the exchange to take place. The cancellation transaction may be prevented from being distributed on the P2P DL until the locktime. If the exchange does not take place by the deadline, the cancellation transaction is added to the P2P DL and effectively pays back the payer and or service provider. If the exchange takes place before expiry of the deadline at the specified geographical location, the exchange transaction may spend the amount, creating a double-spend that hits the P2P DL ahead of the time-locked cancellation transaction, thereby blocking the cancellation transaction. In some embodiments, the metadata may not include a deadline, but instead the cancellation transaction can be solely responsible for cancelling the original invitation transaction. Alternatively, the deadline metadata condition may not automatically cause the spending of a cancellation transaction. In other words, the deadline may be a soft deadline which remains at the payer's control. This deadline

may, therefore, be extended by a party simply allowing it to lapse and still accept late matching invitations. Equally, a service provider may still attempt to match an expired order if it remains unspent.

[0085] Instead of locking in the cancellation transaction at the same time as placing the invitation transaction, a user may wait until after the deadline and manually enter the cancellation transaction if and when the user wishes.

[0086] As mentioned above, a condition may include one or more location conditions which specify, for example, that an exchange transaction is only broadcast onto the P2P DHT if the location of the transactional broadcast is within X metres of specified GPS coordinates. This ensures that the transaction can only take place at a specified location, for example, Bob's coffee shop. This is described in more detail below

[0087] A facility may exist that enables users to create their own new conditions and add them to the list of conditions by allocating them a previously unused condition-code. This facility may be resistant to abuse. For example, each service provider may simply make public their own table of conditions along with an associated condition-code and other parties to the system 100 can choose to adopt the same coding and may also add new ones of their own. Then if disputes arise due to, for example, re-use of condition code, the disputes may be resolved by the service provider or other users of the system 100.

[0088] Some examples of implementation of the present disclosure will now be described by way of example offers and transactions between the first user 106 (referred to herein as Alice) and the second user (referred to herein as Bob).

[0089] **Posting an invitation**

[0090] In a first example, Alice wants to buy some tokenised Canadian dollars (CAD) in exchange for Bitcoins. To advertise her interest, Alice contacts her exchange service provider 104 through, for example, a web interface or via an app running on a tablet or

cell phone. As shown in Figure 2, at step 202, Alice logs into the service provider's 104 web interface. At steps 204 and 206, Alice then sends details of her invitation, including entities to be exchanged (tokenised CAD for bitcoin) and conditions of the exchange and any selected options offered by the service provider, to the service provider. Alice may input this information into an interface hosted by the service provider 104, for example, using normal language which can then be translated by the service provider 104 into a valid invitation or, alternatively, Alice may simply enter the information by pre-selecting options, e.g., via drop-down selection menus.

[0091] At step 208, Alice receives from the service provider 104 a redeem script which has been generated by the service provider 104 based on her selections and includes information concerning the entities Alice wishes to exchange and any conditions associated with the invitation. Because Alice has signed up to use the particular service provider 104, the service provider 104 may already have Alice's public key. Alternatively, Alice may provide her public key to the service provider 104 either during the initial selection, or in response to a request from the service provider 104.

[0092] Alice signs the redeem script at step 210 using her private key, which is a cryptographic pair to her public key, and at step 212 sends the signed redeem script back to the service provider 104 to be distributed. This process may be supported with the use of an app, which may itself be provided by the service provider 104.

[0093] The flow diagram 300 shown in Figure 3 illustrates the corresponding process performed by the service provider 104. At step 302, the service provider 104 receives the invitation details from Alice and generates a redeem script at step 304 using Alice's public key, the entity details and the conditions of the invitation. The redeem script may be in a format suitable for a P2SH bitcoin transaction; the invitation details may be stored in metadata fields in place of 32-byte public keys usually used in multisig unlocking scripts. Figure 4 illustrates a format of the metadata for Alice's invitation, according to one embodiment. In the invitation, Alice requests tokenised CAD and offers bitcoins in return at a rate of no less than 400 CAD/bitcoin. As will be explained

below in more detail, Figure 4 also shows a deadline condition which may be added to the invitation. The deadline condition may cause the invitation to be revoked at the deadline in the event that an exchange has not been finalised based on the invitation.

[0094] The redeem script is then sent to Alice for signing. Upon receipt of a signed redeem script from Alice, at step 308 the service provider 104 generates a hash of the signed redeem script.

[0095] The service provider 104 may use the hash in two ways. Firstly, at step 310, the service provider 104 lists the invitation details along with the hash on the publicly available P2P DHT. As mentioned above, this table employs the torrent technique so it is distributed rather than centralised and hence it will remain publicly accessible and safe from alteration. Other service providers 104 are then able to access the invitation and list it on their own site (indeed, the service provider 104 may simply use the hash table as the sole repository and need not even maintain its own local database of invitations.

[0096] The second way the hash is used is to create a locking script of a Bitcoin transaction, at step 312. This transaction spends an amount of Alice's bitcoins to a P2SH script requiring 2 signatures to unlock: Alice's signature and that of her nominated escrow service provider 110 (which, as mentioned above, might or might not be the same entity as the service provider 104). The purpose of this transaction is twofold. Firstly, the invitation is logged on the P2P DL. Any user or their service provider can verify that the invitation on the P2P DHT is legitimate by ensuring that there exists a matching transaction on the P2P DL (via the matching hash values). Secondly, the transaction 'locks' the commitment made by Alice in her invitation; the amount of bitcoins Alice is offering in exchange for tokenised CAD is the amount spent by the order transaction. Thus, it can be verified that the order is backed by sufficient funds.

[0097] **Pairing matching invitations**

[0098] In a second example which refers to the above described first example, Bob wants to sell some of his tokenised CAD for BTC and has independently listed his own invitation either using the same or a different service provider to the service provider 104 used by Alice. Bob's order, too, has been listed on the hash table and embedded in a P2P DL transaction as described with reference to Figures 2 and 3. The metadata of Bob's invitation is illustrated in Figure 5.

[0099] Referring to Figure 6, a process 600 of matching Alice's and Bob's orders is described. In this example, the service provider 104 is described as performing the process. However, it will be appreciated that any exchange service provider or, indeed, any other suitable third-party may perform the process 600.

[0100] The exchange service provider 104 may run a matching algorithm operable to identify a full or partial match between Alice's and Bob's invitations. At step 602, the exchange service provider 104 scans the P2P DHT for matching entity details. During the scan, the service the provider 104 checks for a match between the entity details of Alice's and Bob's invitations. If a match is not found at step 604, then the process returns to step 602 and the exchange service provider 104 continues to scan the P2P DHT for matching entity details. If a match is found at step 604, the process 600 continues to step 606 where a check is made for a match between one or more of the conditions of each of Alice's and Bob's invitations. If no match is found at step 606, the process returns to step 602. If a match between one or more conditions is found, then the process moves to step 608 in which the exchange service provider 104 attempts to create and finalise a transaction between Alice and Bob.

[0101] A direct match of all conditions in the two invitations may not be required at step 606 for a positive match to be confirmed. Indeed, the process 600 may only require that some of the conditions match. Additionally or alternatively, one or more of the conditions need not fully match. For example, if conditions being compared are exchange rates proposed in each condition, the process 600 may confirm a positive match provided the rates are within a predetermined threshold range of each other. For example, if Alice proposes a minimum rate condition of 4×10^{-5} tokenised

CAD/satoshi and Bob's equivalent minimum proposed rate is 3.9 x 10^{-5} tokenised CAD/satoshi, the process may still confirm a condition match even though Bob's offered rate doesn't quite satisfy Alice's original requirements. In such circumstances, upon match, Alice may be given an option to accept. It will be appreciated that if Bob's equivalent minimum proposed rate is 4.1 x 10^{-5} tokenised CAD/satoshi, then the condition would be satisfied. In another example, the conditions might be respective values for goods and services proposed in an offer and request. The process 600 may again confirm a positive match provided the two values are within a predetermined threshold range of each other. In each case, the predetermined threshold range may be, for example, a discrete value or a percentage of the offer value or request value.

[0102] As stated previously, the transaction metadata for each or both of Bob's and Alice's invitations may further comprise one or more location conditions which may specify, for example, that a transaction is only broadcast onto the P2P DL if the location of the transactional broadcast is within X metres of specified coordinates. This ensures that the transaction can only take place at a specified location, for example, Bob's coffee shop.

[0103] Once a match is found, and before completing the transaction, one or more intervening steps may be performed. These may include an alert to the parties that a match has been found, followed by a request to each or both parties for confirmation that they wish to proceed, etc. For example, as discussed above, where a condition is nearly but not fully met by one or more users, a match may still be recorded but not finalised until all parties are happy with the conditions of the invitations. This process may lead to counter offers to negotiate a final agreement on conditions, which may then lead to the generation of further invitations in accordance with processes described above.

[0104] The final exchange may be executed by creating one or more bitcoin transactions that spend the outputs of each invitation transaction. The inventors have found several methods of completing the transaction, which may depend on circumstances including but not limited to: the users involved in the transaction, the

entities being exchanged, and the service providers and issuers involved in the transaction. Some examples of these methods are described below.

[0105] Following on from the examples described above with reference to Figures 2 to 6, a transaction table 500 for the separate transactions for Alice-Bob and Bob-Alice is shown in Figure 7 and a schematic diagram 600 of the transaction flow is shown in Figure 8. As with the metadata values shown in Figures 4 and 5, values provided in the transaction table 500 are shown. In this example, Alice's bitcoins in her invitation transaction are spent to Bob, and Bob's CAD-tokenised bitcoin in his invitation transaction are spent to Alice.

[0106] Referring first to the Alice-Bob transaction, the input 602 of this transaction is provided from the output of the invitation transaction which was placed on the P2P DL with Alice's invitation. Like the first transaction, the input script is signed by both Alice and the escrow service provider 110 (assuming Alice is happy for the transaction to proceed). The script unlocks the spent bitcoins which can then be output to a) Bob as his payment in return for the tokenised CAD (604), b) the exchange service provider 104 as payment for the exchange (606), and c) to Alice as change if any is due (608).

[0107] Referring now to the Bob-Alice transaction, this transaction has two inputs. The first input 610 of the transaction is provided from the output of the invitation transaction which was placed on the P2P DL with Bob's invitation. Because the input to this transaction is tokenised, the input script needs to be signed by both Bob and his token issuer. In this circumstance, the token issuer may also operate as an escrow, thereby withholding the funds until Bob (and optionally Alice) are satisfied with the transaction. The signed script unlocks the spent tokens which can then be output to a) Alice as payment in return for the BTC (612), and b) back to Bob as a change token to the value of the original token less the value sent to Alice (614). The second input 616 is from a previous bitcoin transaction of Bob. This input is unlocked and output to a) the service provider 104 as payment for the exchange, b) the bitcoin miner as a fee for the exchange transaction and c) to Bob as change in bitcoins to the value of the original bitcoin input value less the service provider's 104 fee and the miner's fee.

[0108] The service provider's 104 fee for each transaction may be a slice of the value of the transaction. Alternatively or additionally, the fee may be a slice of the exchange rate spread between corresponding rate conditions presented in the two invitations. For example, where the offered rates overlap, the service provider 104 may fulfil both sides of the exchange at the asking rate of each and retain the difference as the fee. Alternatively or additionally, a flat fee (in satoshis, tokenised currencies or other) may be taken by the service provider 104.

[0109] Once the transaction has been completed, Bob's and Alice's respective service providers may remove their invitation entries from the P2P DHT or enter a further entry which nullifies the original entry. For example, a service provider may simply leave the entry on the P2P DHT since the entry corresponds to a spent transaction, which implies that the invitation is no longer valid. Alternatively, a service provider may mark the transaction with a field that specifies that it has been spent. This may be a separate field on the DHT which corresponds to a particular entry but does not change the actual metadata associated with the invitation (this ensures that the script hash still matches the hash on the transaction). Alternatively, a service provider may delete the entry from the P2P DHT. However, a benefit of the P2P DHT is a permanent audit control for transaction using the system 100. Preferably, therefore, deletion of entries from the P2P DHT is prevented or deleted entries are archived so as to maintain a record of the entry. In one example, deleted entries are archived.

[01 10] In the above example transaction, no puzzle is exchanged. In other words, the two transactions (Alice-Bob and Bob-Alice) are completely separate and discrete. It may be preferable in some cases, however, for the two transaction either to be valid or invalid. Figure 9 shows an alternative transaction example in which a puzzle is exchanged in Alice's transaction (Alice-Bob). By doing so, the two transactions are locked together so that one cannot be spent without the other also being spent. This prevents a transaction going through from one party to another without the opposite transaction also going through.

[0111] In the above two example two bitcoin transactions are performed to complete the exchange. Where possible, however, it is preferable to consolidate the above two transactions into a single bitcoin transaction. Doing so automatically locks together the two parts of the exchange and leads to a reduction in the overall fees paid by Alice and Bob for the transaction.

[01 12] Figure 10 shows a transaction table 700 for the single transaction between Alice and Bob. The transaction flow for the exchange is the same as that of the previous two examples, i.e. as shown in Figure 6. However, the exchange is consolidated into a single multi-input-multi-output (MIMO) transaction. It is noted that in Figure 8, two separate fees are paid to the exchange service provider 104. However, if the exchange service provider 104 is the same for both Bob and Alice, these two fees may be consolidated into a single transaction, paid by Bob or Alice, or both Bob and Alice.

[0113] **Posting an invitation**

[01 14] In a third example, Bob offers an exchange of a gold nugget for a specific amount of Bitcoins. To publish his offer, Bob contacts a first third party, which in this case represents his exchange service provider 104. In this particular example, the exchange service provider 104 provides a communication interface that is implemented by a software application running on Bob's cell phone. Thus, Bob can enter his offer for the exchange of the gold nugget using the software application on his cell phone as will be described in more detail below with reference to the process 700 shown in Figure 11.

[0115] As shown in Figure 11, at step 702, Bob logs onto the exchange service provider's 104 software application on his cell phone and enters at steps 704 and 706 details of his offer into the application. The details of the offer in this particular example include the entity offered for exchange, i.e. the gold nugget, an expected amount of Bitcoins in exchange for the gold nugget, and a location condition.

[0116] The location condition may include one or more of the following information:

a) location information indicative of a geographical location or area for the exchange of the gold nugget;

b) location format information indicative of a format of the location information;

c) range information indicative of a range limit on the geographical location or area relating to the exchange; and

d) range format information indicative of a format of the range information.

[01 17] In this particular example, the location condition of Bob's offer specifies an area which is defined as range of 20 metres from Bob's store "the Gold Emporium". The range of 20 metres may be specified to allow for known limits to accuracy of GPS readings by Bob's cell phone in the specified area. However, it will be appreciated that a range may be specified for various reasons. For example, the location condition may specify a country or a city in which the exchange needs to take place.

[0118] Bob may input the above mentioned information into an interface hosted by his service provider's 104 software application, for example using normal language which can then be translated by the service provider 104 into a valid offer or, alternatively, Bob may simply enter the information by pre-selecting options, e.g., via drop-down selection menus.

[01 19] At step 708, a redeem script is generated using Bob's public key and Bob's selections which include information concerning the entities Bob is offering for exchange and any conditions including the above location condition associated with the offer. Bob's public key is part of an asymmetric cryptographic pair which comprises a private key and the public key. In addition, the generated redeem script includes a further public key that is the service provider's 104 public key.

[0120] At step 710, the software application facilitates Bob to sign the generated redeem script with his private key which corresponds to the above mentioned public

key. The generated redeem script may be in a format suitable for a P2SH bitcoin transaction; the invitation details may be stored in metadata fields in place of 32-byte public keys usually used in multisig unlocking scripts as described above.

[0121] At step 712, a hash of the signed redeem script is generated. At step 712, the script along with the hash is published on the publicly available P2P DHT. As mentioned above, this table employs the torrent technique so it is distributed rather than centralised and hence it will remain publicly accessible and safe from alteration. The invitation can be accessed by other users and exchange service providers.

[0122] Figure 12 illustrates a format of the metadata for Bob's script, according to one embodiment. In the invitation of the script, Bob requests Bitcoins in exchange for a gold nugget. As will be explained below in more detail, Figure 12 also shows the above mentioned location condition which is added to the invitation.

[0123] The general format of condition metadata blocks in the P2P Exchange model is shown in Figure 13. Metadata blocks are 32 bytes long. The first two bytes of a Condition metadata block are a numeric code which defines the condition, as specified via a separate look-up table such as distributed hash table. The format for the remaining 30 bytes of a Condition metadata block will depend on the Condition.

[0124] As shown in the example for Bob's script in Figure 12, the location condition in this particular scenario is allocated the value 0004.

[0125] The general format of a location condition code is shown in Figure 14 in which explanations for the fields are as follows:

LocationCode	2 byte numeric code indicating the format of the Location. A separate look-up table defines the codes and their specification (see for example paragraph [111]).
RangeCode	Optional value indicating the units of measurement to specify an acceptable range subcondition. If set to 0x00 then no range subcondition is specified.

	A separate look-up table defines the codes and their specification (see for example paragraph 112).
Range	Optional value indicating the acceptable range subcondition. This value is irrelevant (ignored) if RangeCode = 0x00
Location	The actual location, specified in the format required by the LocationCode

[0126] As will be appreciated, a location condition can be specified in different formats. Some specific, non-limiting examples are given below:

LocationCode	Description	Byte Format
0x0101	GPS Co-ordinates	01 - 01: 'N'/'S'
		02 - 08: degrees (NN) minutes (NN.NNN)
		09 - 09: 'E'/'W
		10 - 17: degrees (NN) minutes (NN.NNN)
0x0001	USA Zip Code	01 - 05: NNNNN
0x0061	Australia Postcode	01 - 04: NNNN
0x1044	UK Building Name and Level	01 - 23: first 23 characters of building name
		24 - 26: level number (NNN)

[0127] As mentioned above, the location condition may further specify a range from the specified location. The range may allow for known limits to accuracy of GPS readings or may define an area in which the exchange of the entity must take place for the exchange to be finalised. Some examples for the format of the range code is given below:

RangeCode	Description
0x00	No Range specified
0x01	Metres

0x02	Kilometres
0x11	Within adjoining category [e.g.: if LocationCode = 0x0001 then this would mean 'adjoining USA postcode'; if LocationCode = 0x1044 then this would mean 'next floor above or below'; etc.]

Matching an Offer

[0128] In a fourth example which refers to the above described third example, Alice has seen Bob's offer on the P2P DHT and goes to Bob's Gold Emporium as specified in the location condition of Bob's script to inspect the gold nugget.

[0129] On examining the gold nugget and negotiating an amount of Bitcoins in exchange for the gold nugget, Alice creates a matching offer using a software application that is running on her cell phone. In particular, the software application facilitates Alice to enter details of the entities to be exchanged, i.e. one gold nugget in exchange for the negotiated amount of Bitcoins. Using the entered details, a redeem script is generated for Alice which is then hashed and published together with the hash of the redeem script on the P2P DHT. The metadata of Alice's offer is illustrated in Figure 15.

[0130] In this particular example, Bob's script as shown in Figure 12 does not specify a minimum or maximum rate of Bitcoins for the exchange to be finalised. Thus, when the exchange service provider 104 runs the matching algorithm as described above to identify a match between Bob's and Alice's offers, Bob will be given an option to accept Alice's offer.

[0131] In this particular example, the process of matching Alice's and Bob's scripts further includes a step of verifying that the location condition is fulfilled which is specified in Bob's script as shown in Figure 12. This verification step may be performed in various ways.

[0132] For example, the exchange service provider 104 may communicate with Bob's and/or Alice's cell phone to determine the location of the cell phone. This location may then be compared with the location and the range specified in the location condition of Bob's script.

[0133] If the location is specified as a building name, such as Bob's Gold Emporium, the exchange service provider 104 may determine GPS coordinates associated with this location by communicating with a proprietary database or a further software application, such as Google maps. The determined GPS coordinates may then be compared with the GPS coordinates of Bob's or Alice's cell phone.

[0134] In a further example, the exchange service provider 104 may facilitate manual input by Bob and/or Alice to confirm that the location condition is fulfilled. This step may, for example, be conducted together with the step of accepting Alice's offer.

[0135] If the location condition is found to be fulfilled, the exchange service provider 104 will attempt to create and finalise a transaction between Alice and Bob. In one example, a transaction is only broadcast onto the P2P DL if the location of the transactional broadcast is within X metres of specified coordinates. This ensures that the transaction can only take place at a specified location, for example, Bob's Gold Emporium.

[0136] The final exchange may be executed by creating one or more bitcoin transactions that spend the outputs of each invitation transaction as described with reference to the first and second examples shown in Figures 2 to 10.

Transactions involving more than two parties

[0137] Transactions described above have been in relation to exchanges between two parties and two entities. It will, however, be appreciated that in some examples more than two parties and/or entities may be involved in an exchange. Consider, for example, the following scenario. Alice wishes to exchange bitcoins for apples, but will

only accept a minimum of 1000 apples. Bob wants to exchange apples for bitcoins, but can only supply 500 apples. Carol wants to exchange apples for bitcoins but can only supply 600 apples. In such a circumstance, the conditions of Alice's invitation cannot be satisfied by Bob or Carol individually. Together, however, Bob and Carol have 1100 apples and so can satisfy Alice's invitation conditions.

[0138] In another example, Alice wishes to exchange tokenised CAD for tokenised GBP, Bob wishes to exchange tokenised GBP for tokenised AUD, and Carol wishes to exchange tokenised AUD for tokenised CAD. No direct match exists between any two of the three parties, but combined, each of the invitations can be satisfied - Alice's tokenised CAD can go to Carol, Bob's tokenised GBP can go to Alice, and Carol's tokenised AUD can go to Bob. Figures 16A to 16C show exemplary transaction tables for transactions between Alice, Bob and Carol.

[0139] Referring first to Figure 16A, a transaction table is shown for the payment from Alice to Carol. Alice has $1500 of tokenised CAD and needs tokenised GBP500 from Bob. The transaction pays tokenised CAD 1000 to Carol from Alice and Alice pays herself the remaining tokenised CAD500 (1500-1000). Using regular BTC, Alice may pay the service provider's fee (which may be a flat fee as is shown in Figure 11A or a fee dependent on the value of the transfer) and pays herself the change minus 1000 satoshis for the miner.

[0140] Referring now to Figure 16B, a transaction table is shown for the payment of tokenised GBP to Alice from Bob. Bob has tokenised GBP750 and needs tokenised AUD from Carol. The transaction pays tokenised GBP500 to Alice from Bob and Bob pays himself the remaining tokenised GBP250 (750-500). Using regular BTC, Bob may pay the service provider's fee (which may be a flat fee as is shown in Figure 16B or a fee dependent on the value of the transfer) and pays himself the change minus 1000 satoshis for the miner.

[0141] Referring now to Figure 16C, a transaction table is shown for the payment of tokenised AUD to Carol from Bob. Carol has tokenised AUD1500 and needs tokenised

CAD from Alice. The transaction pays tokenised AUD1000 to Bob from Carol and Carol pays herself the remaining tokenised AUD500 (1500-1000). Using regular BTC, Carol may pay the service provider's fee (which may be a flat fee as is shown in Figure 11C or a fee dependent on the value of the transfer) and pays herself the change minus 1000 satoshis for the miner. It will be appreciated that where an exchange is composed of two or more separate transactions (e.g. 1: Alice transfers to Bob; and 2: Bob transfers to Alice), the transactions may be linked so as to ensure that either every party receives their entitlement or none of them do. This may be achieved by satisfying the following condition (using an example of two parties, A and B, but easily extendable to three or more parties): The transaction output transferring from A to B exists and is spendable by B only if the transaction output transferring from B to A also exists and is spendable by A at the same time, and vice versa. It is noted that party A and party B refer to the required set of signatories for each transaction, not just Alice and Bob (for example, it may include token Issuers, escrow, etc.).

[0142] **Receiving a selection from a user**

[0143] In a variation of the example exchange described above with reference to Figure 6, instead of a service provider parsing the P2P DHT for matching orders, users themselves may scan or browse the P2P DHT to view the current invitations as described with reference to the above fourth example. Browsing may be facilitated by a third-party, such as the exchange service provider 104. The third-party may provide an interface within which users can browse, scan and search for invitations which they might be interested in.

[0144] Users may then skip the process of entering their own prospective invitation on the P2P DHT, but instead choose to create an invitation which matches or nearly matches the order which they're interested in.

[0145] For example, Bob may find Alice's invitation on the P2P DHT via a browsing or searching interface. In which case Bob may enter his own invitation to match that of Alice's. Bob may do this in one of several ways. In one example, there may be a

facility on the interface which displays Alice's order to 'Accept' her order. If Bob is a client of the same exchange service provider 104 as that which Alice used for the invitation, they may already have access to Bob's eWallet (public key etc.) and can thus create the matching order based on such information. Accordingly, the exchange service provider 110 may generate a redeem script for the matching invitation, send this to Bob for signing, receive the signed redeem script and enter the order onto the P2P DHT in preparation for the transaction. If Bob is not a client of Alice's exchange service provider 104, a facility may be provided to allow Bob to enter the required information and authorisations that can then enable the Service provider to create Bob's matching order. The same process as that described above with reference to Figures 7 and 8 may then follow.

[0146] The above example describes the exchange of BTC for tokenised CAD. However, it will be appreciated that the system 100 works for any type of exchange including, for example, BTC for token of any type (i.e. representing any contract, not just for currency contracts), token of any type for token of any other type, BTC for goods/services, tokens for goods/services, or goods/services for goods/services. Additionally and theoretically, the above process can be modified to the exchange of BTC for BTC, although such an exchange has no real meaning.

[0147] **Exchanging goods/services**

[0148] A slight variation of the above described transaction process is required when goods/services are involved in the exchange.

[0149] In such a case, the transaction (of the goods and/or services) comprises a description of the goods or services involved in the exchange. Unlike a token, which is represented by a contract or title deed, the description does not form a contract.

[0150] The description may or may not uniquely identify the item. For example, where a physical item is involved in the transaction, the description may refer explicitly to a unique identifier associated with that physical item. Additionally or alternatively,

description metadata may include one or more of the following: a) a general description of a desired item offered or requested, e.g., "dishwasher, < 3 years", b) reference to a specific item for sale on an auction website, e.g., "second hand product put up for sale on an auction site", c) any number of an item type, e.g., advertise 15 t-shirts for sale that can be bought as singles or any quantity up to 15, d) reference to cash, in any specified currency, e) description of labour and payment per single job completion or for regular lawn mowing (repeating job) with payment per repetition or hour, or f) one or more keywords, e.g. "dishwasher".

[0151] With regard to services, services may be backed by a contract as with a token. As such, a service may be divisible into shares, and non-divisible services may be considered as one-time jobs, i.e. divisible but comprising a single share (1 share). If the service is non-divisible, it may be treated as a token for the purpose of invitation and exchange. Where an item is backed by a token it is treated as a token for the purpose of both invitation and exchange and is exchanged in the same manner as other tokens, such as tokens for fiat currency.

[0152] An example of a transaction involving a physical item associated with a unique identifier will now be described. Similar to previous examples, in the present example, Alice has used her exchange provider to place an invitation on the P2P DL and P2P DHT. The invitation includes a description that she will buy a physical item having unique identifier XYZ123, which may relate to Rafael's masterpiece *Deposition of Christ*, for no more than 2500 BTC. Equally, Bob may have placed a matching invitation that he will sell item XYZ123 for no less than 2400 BTC. Alice may have browsed the P2P DL and found the item with item number XYZ123 and placed a matching order based on this information or, alternatively, Alice may have placed a generic invitation which is then matched by a third party, e.g., an exchange service provider, and subsequently a new invitation including the catalogue item number and description is made to match Bob's order.

[0153] For transactions involving unique IDs, it will be appreciated that such IDs must be unique not only to a particular exchange service provider, but also across the

entire P2P DL. Therefore, if the unique identifier is not unique to the device (e.g. the devices serial number) then the exchange service provider may generate a unique identifier for the device. To ensure that each identifier is unique to the entire P2P DL, each exchange service provider may, for example, have their own unique code prefixed to numbers they use to uniquely identify products being advertised on the P2P DL.

[0154] Once an agreement has been reached between Alice and Bob, the transactions may take place in accordance with the example transaction processes described above with reference to Figures 7 to 10.

[0155] A further example of a transaction involving a physical item will now be described. In this example, however, the item does not have a unique identifier associated therewith.

[0156] Where an invitation includes an offer to sell a plurality of similar items, metadata may be required to describe the maximum and minimum quantity of items which can be bought with any one transaction. For example, Alice may place an invitation inferring that she will sell up to 15 *Dead Lizard* 2015 concert tour t-shirts for 0.025 BTC each - at least 5 per transaction. In this case, metadata values may include a minimum rate (0.025BTC / 15 items); a maximum quantity (Offer-QTY-max (15)) and a minimum quantity (Offer-QTY-min (5)). The table below summarises the metadata associated with the invitation.

Field	Sub-field	Value	Comments
A	ContractType	OxOOOOFFOl	Indicates P2P Exchange Order
	ContractPointer		IPv6 address of the actual contract file location
	OfferRequestCodes		Coded value indicating Offer-type (4 bits) + Request-type (4 bits)
	Conditions	000000112	Bitfield - flags indicating the presence of extra conditions in metadata fields
	Rate-m in	0.025	BTC / 15 items
	Rate-max		
C	Offer-QTY-max	15	Maximum quantity of t-shirts per sale
	Offer-QTY-min	5	Minimum quantity of t-shirts per sale
	Request-QTY-max		

	Request-QTY-min		
D	Offer-Item -ID	01245D2SA	Unique ID related to item
	Request-item-ID		

[0157] The actual BTC value on the payment transaction will then be calculated by an exchange service provider. Because this transaction simply represents an invitation to perform an exchange, the actual value of the transaction may be small, e.g. as small as dust (546 satoshis). Alternatively, as described below, the value may be a nominal amount required by a service provider to guarantee an invitation (e.g. so Alice is incentivised not to pull out).

[0158] In a further example, goods in the form of hard currency (cash) may be exchanged. For example, Alice may place an invitation to sell bitcoin for Canadian dollars (hard currency instead of tokenised) with a maximum purchase of 150 BTC. The invitation may additionally include a location condition that the exchange must take place at her shop address: 371 Whimsy Avenue, Brentford. After placing a matching invitation, in order to finalise the transaction, Bob may then bring the cash to Alice's shop to hand over in exchange for a payment in bitcoin transaction. The actual digital transaction of bitcoins to Bob and of the digital record of transfer of hard cash to Alice, may then take place once Bob and Alice have met in her shop for the physical transfer.

[0159] In the case of a transaction involving goods/services swapped for other goods/service, it will be appreciated that the transaction on the P2P DL exists as a record only and not for exchanging any value between parties (apart from any fees to service providers etc.). Users may use the system and opt to pay the nominal service fee for entry of the transaction onto the P2P DL so that the exchange is permanently recorded.

[0160] It is noted that the original invitation transactions act as invitations only and not as value transfer or as records of events. If a goods-for-goods exchange occurs, such that the exchange involves physical items only, the final exchange need not be

recorded on the P2P DL, since the P2P DL is not required to complete any transactions in the final exchange. Notwithstanding, if parties to the exchange of physical items wish to record the exchange on the P2P DL, they may each spend their invitation transactions to one another subject to a fee for the miner for doing so. If the parties do not wish to record the final exchange on the P2P DL, they may each spend their invitation transactions back to themselves, or even leave them unspent on the P2P DL.

[0161] In the case of an exchange involving BTC for goods, or tokens for goods, at least one transaction is spent on the P2P DL to transfer the value of the BTC or token. In this case, the invitation transaction offering the goods may or may not be spent since the value of the exchange (goods) is not transferred by spending that invitation transaction. However, again, parties may decide to spend the transaction nonetheless, in order to provide a permanent record of the transfer (e.g., a receipt of sale).

[0162] The amount spent on the above transactions might not represent the offered amount in some cases, particularly when Alice's offer is not bitcoin or tokens but goods/service. Instead, the Service Provider might require a 'deposit' by Alice of an amount representing the value of the goods, or might only require a nominal amount if Alice is able to 'guarantee' the offer in another way, or (iii) might itself provide the bitcoins on Alice's behalf (she might not have any) and cover this funding cost by whatever means they charge a fee to the client.

[0163] In embodiments described above, users' invitations are published on a P2P DHT. In some embodiments, however, a user's invitation (e.g. the script and script hash) may be published on a website, or sent directly to another user.

[0164] In some embodiments, a user's invitation may be stored locally by a service provider. For example, the service provider may host a private auction in which only certain users can access the user's invitation details.

[0165] It will be appreciated by persons skilled in the art that numerous variations and/or modifications may be made to the above-described embodiments, without

departing from the broad general scope of the present disclosure. The present embodiments are, therefore, to be considered in all respects as illustrative and not restrictive.

[0166] The steps, features, integers, compositions and/or compounds disclosed herein or indicated in the specification of this application individually or collectively, and any and all combinations of two or more of said steps or features.

CLAIMS:

1. A computer implemented method of offering an exchange of an entity, the method comprising:

generating a first script, the first script comprising:

a first set of metadata associated with a first invitation for the exchange of a first entity by a first user, the first set of metadata comprising an indication of the first entity to be offered for exchange and a first location condition for the exchange,

a first user public key (P1A) associated with the first user, wherein the first user public key (PI A) is part of an asymmetric cryptographic pair comprising the first user public key (P1A) and a first user private key (VIA),

hashing the first script to generate a first script hash; and

publishing the first script and the first script hash on a distributed hash table (DHT).

2. The method of claim 1 wherein the first script further comprises

a first third-party public key (PIT) associated with a first third-party, wherein the first third-party public key (PIT) is part of an asymmetric cryptographic pair comprising the first third-party public key (PIT) and a first third-party private key (V1T).

3. The method of claim 1 or 2, comprising:

generating a first invitation transaction for inclusion on a peer-to-peer (P2P) distributed ledger, the first invitation transaction comprising an indication of a second entity to be transferred in exchange for the first entity, and the first script hash.

4. The method of claims 1 to 3 wherein the P2P distributed ledger is the Bitcoin block chain.

5. The method of any one of the preceding claims, wherein the first entity that is offered for exchange and/or the second entity to be transferred in exchange for the first entity may be one or more of the following:

a) a cryptocurrency;

b) a contract;

c) goods;

d) services.

6. The method of claim 4 wherein the contract is for one or more of the following:

a) a fiat currency;

b) title deeds;

c) goods;

d) services.

7. The method of any one of the preceding claims, wherein the first location condition comprises one or more of the following:

a) location information indicative of a geographical location or area for the exchange of the first entity;

b) location format information indicative of a format of the location information;

c) range information indicative of a range limit on the geographical location or area relating to the exchange; and

d) range format information indicative of a format of the range information.

8. The method of any one of the preceding claims comprising:

matching the first invitation of the first script to a second script that is published on the DHT by comparing at least the first entity specified in the first script with a second entity specified in the second script,

wherein the second script comprises a second set of metadata associated with a second invitation for an exchange of a second entity by a second user, the second set of metadata comprising an indication of the second entity to be offered in exchange for the first entity,

a second user public key (P2A) associated with the second user, wherein the second user public key (P2A) is part of an asymmetric cryptographic pair comprising the second user public key (P2A) and a second user private key (V2A), and

a second third-party public key (P2T) associated with a second third-party, wherein the second third-party public key (P2T) is part of an asymmetric cryptographic pair comprising the second third-party public key (P2T) and a second third-party private key (V2T).

9. The method of step 8 comprising:

determining whether the location condition is fulfilled by identifying a location of the first user and/or the second user.

10. The method of claim 8 or 9 wherein the second third party is the same as the first third party.

11. The method of claim 8, 9 or 10, further comprising:

publishing the second script on the DHT, comprising:

generating the second script:

hashing the second script to generate a second script hash; and

publishing the second script and the second script hash on the DHT.

12. The method of claim 11, further comprising:

generating a second transaction invitation for inclusion on the P2P distributed ledger, the second invitation transaction comprising an indication of the first entity to be transferred in exchange for the second entity; and the second script hash.

13. The method of any one of the preceding claims, further comprising:

generating a first exchange transaction for inclusion on the P2P distributed ledger, the first exchange transaction comprising:

the first script,

the first user public key (PI A),

the first third-party public key (PIT),

a first input provided from an output of the first invitation transaction, and

a first output indicating a first quantity of a first entity to be transferred to the second user; and

recording the first exchange transaction on the P2P distributed ledger.

14. The method of claim 12 and 13, further comprising:

generating a second exchange transaction for inclusion on the P2P distributed ledger, the second exchange transaction comprising:

the second script,

the second user public key (P2A),

the second third-party public key (P2T),

a second input provided from an output of the second invitation transaction, and

a second output indicating a second quantity of a second entity to be transferred to the first user; and

recording the second exchange transaction on the P2P distributed ledger.

15. The method of claim 12, further comprising:

generating a first exchange transaction for inclusion on the P2P distributed ledger, the first exchange transaction comprising:

the first script,

the first user public key (PI A),

the first third-party public key (PIT),

the second script,

the second user public key (P2A),

the second third-party public key (P2T)

a first input provided from an output of the first invitation transaction,

a second input provided from an output of the second invitation transaction,

a first output indicating a first quantity of a first entity to be transferred to the second user, and

a second output indicating a second quantity of a second entity to be transferred to the first user; and

recording the first exchange transaction on the P2P distributed ledger.

16. The method of claim 13 or 15, wherein generating the first exchange transaction comprises:

facilitating signing of the first script with the first user private key (VIA); and

faciliating signing of the first script with the first third-party private key (PIT).

17. The method of claim 14 or 15, wherein generating the second exchange transaction comprises:

facilitating signing of the second script with the second user private key

(VIA); and

facilitating signing of the second script the second third-party private key (P2T).

18. The method of any one of claims 13 to 15, further comprising facilitating one or more of the first and second users to accept the exchange before generating or recording the first and/or second exchange transactions.

19. The method of any one of the preceding claims, wherein one or more of the first exchange transaction, and the second transaction is a pay-to-script-hash (P2SH) transaction.

20. A computer system for offering an exchange of an entity, the computer system comprising:

a first processing device configured to:

generate a first script comprising:

a first set of metadata associated with a first invitation for the exchange of a first entity by a first user, the first set of metadata comprising an indication of the first entity to be offered for exchange and a first location condition for the exchange,

a first user public key (PI A) associated with the first user, wherein the first user public key (P1A) is part of an asymmetric cryptographic pair comprising the first user public key (P1A) and a first user private key (VIA)

hash the first script to generate a first script hash; and

publish the first script and the first script hash on a distributed hash table (DHT).

21. The method or system of any preceding claim where the first and/or second set of metadata is be provided in a redeem script,

22. The method or system of any preceding claim, where the first and/or second set of metadata is provided in the script at a location which is designated in a blockchain protocol as a location for a cryptographic key.

23. A processor or group of processors operable to perform the method of any of claims 1 to 18.

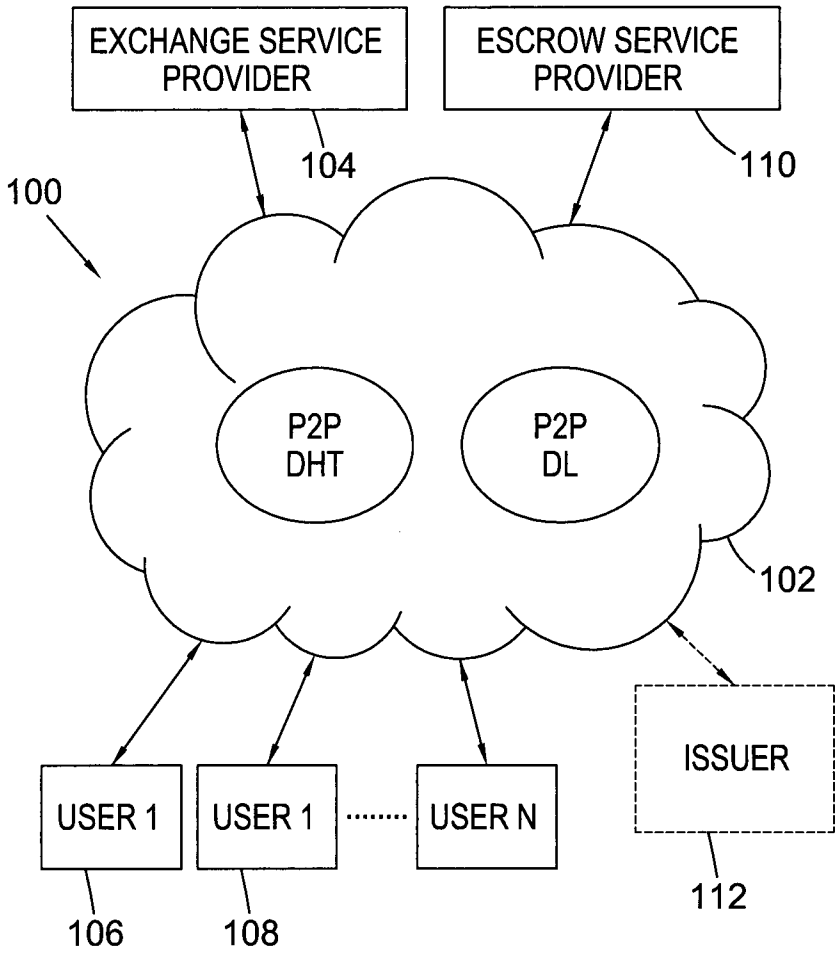
EXCHANGE SERVICE PROVIDER
ESCROW SERVICE PROVIDER
104
110
100
P2P DHT
P2P DL
102
ISSUER
USER 1
USER 1
USER N
106
108
112

Fig. 1

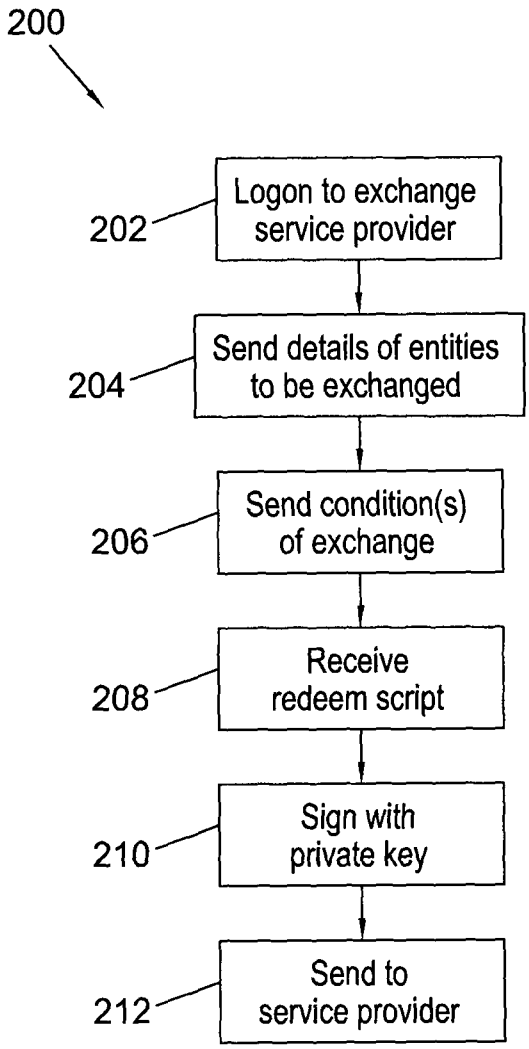
200
202
Logon to exchange service provider
204
Send details of entities to be exchanged
206
Send condition(s) of exchange
208
Receive redeem script
210
Sign with private key
212
Send to service provider

Fig. 2

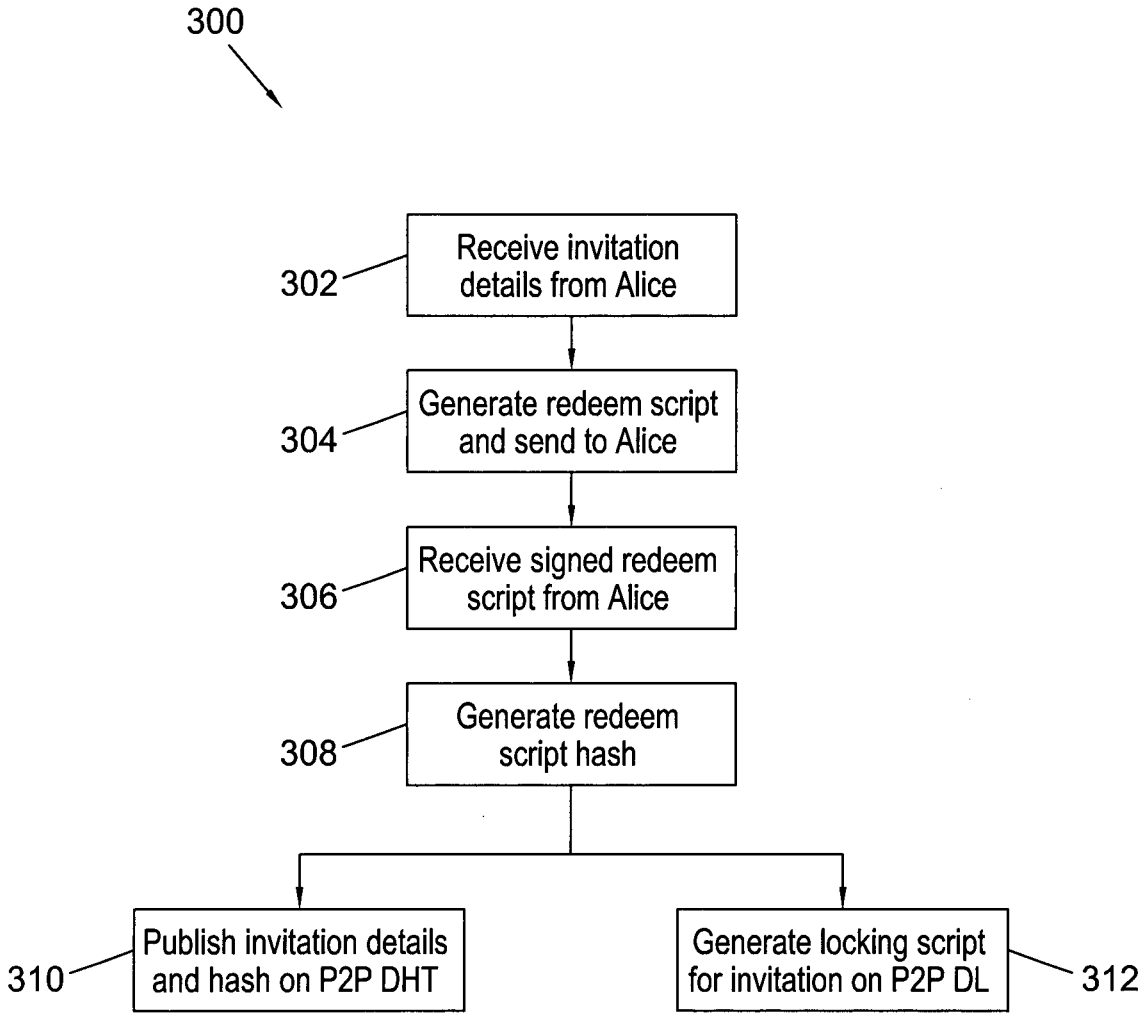
300
Receive invitation details from Alice
302
Generate redeem script and send to Alice
304
Receive signed redeem script from Alice
306
Generate redeem script hash
308
Publish invitation details and hash on P2P DHT
310
Generate locking script for invitation on P2P DL
312

Fig. 3

ALICE

Offer:	BTC
Request:	Token
Offer-type-code	0000_2
Offer-QTY-max	NULL
Offer-QTY-min	NULL
Offer-item-ID	
Rate-min	$4.00000000 \times 10^{-5}$ shares/satoshi
Rate-max	NULL (no maximum - i.e. infinity)
Request-type-code	0001_2
Request-item-ID	CAD1234
Request-QTY-max	10000 shares
Request-QTY-min	NULL (no minimum - i.e. 0)
Offer-Deadline	midnight 31-Dec-2016
Explanation of the order	I will buy tokenised CAD for BTC up to $1000 at a rate of no less than $400/BTC. Deadline is midnight 31-Dec-2016

Fig. 4

BOB

Offer:	TOKEN
Request:	BTC
Offer-type-code	0001_2
Offer-QTY-max	15000 shares
Offer-QTY-min	NULL (no minimum - i.e. 0)
Offer-item-ID	CAD1234
Rate-min	2.43902439×10^2 satoshis/share
Rate-max	NULL (no maximum)
Request-type-code	0000_2
Request-item-ID	
Request-QTY-max	NULL
Request-QTY-min	NULL
Offer-Deadline	NULL
Explanation of the order	I will sell tokenised CAD up to $1500 at a rate of no more than $410/BTC

Fig. 5

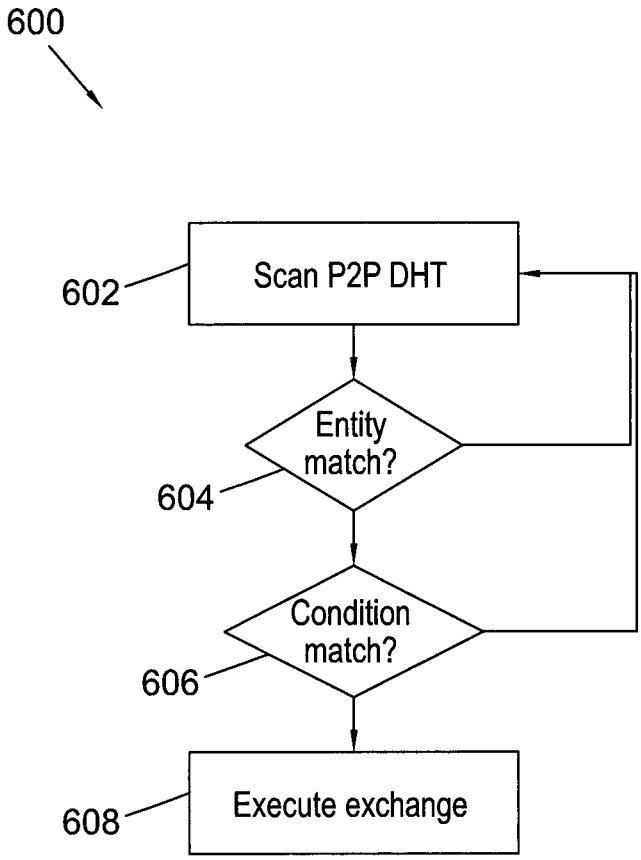
600
602
Scan P2P DHT
Entity match?
604
Condition match?
606
Execute exchange
608

Fig. 6

Alice sends BTC to Bob	
ALICE00040	Trancaction-ID
Version number	Version number
1	Number of inputs
ALICE00010	Prev Trans Output
IDX-00	Prev Trans Output index
Script length	Script length
Sig-Alice Sig-EscrowA < OP_2 metadataA metadataB metadataC metadataD metadataE PubK-Alice PubK-EscrowA OP_7 OP_CHECKMULTSIG>	ScriptSig
Sequence number	Sequence number
3	Number of outputs
246,913,580 satoshi	Output value
Output script length	Output script length
OP_DUP OP_HASH160 <PubK-Bob hash> OP_EQUALVERIFY OP_CHECKSIG	Output script
2,469,136 satoshi	Output value
Output script length	Output script length
OP_DUP OP_HASH160 <PubK-Server hash> OP_EQUALVERIFY OP_CHECKSIG	Output script
3,116,284 satoshi	Output value
Output script length	Output script length
OP_DUP OP_HASH160 <PubK-Alice hash> OP_EQUALVERIFY OP_CHECKSIG	Output script
LockTime	LockTime

Bob sends token to Alice	
BOB00030	Trancaction-ID
Version number	Version number
2	Number of inputs
BOB00010	Prev Trans Output
IDX-00	Prev Trans Output index
Script length	Script length
Sig-Bob Sig-Issuer < 2 metadata1 metadata2 PubK-Bob PubK-Issuer 4 OP_CHECKMULTSIG>	ScriptSig
Sequence number	Sequence number
BOB00010	Prev Trans Output
IDX-01	Prev Trans Output index
Script length	Script length
Sig-Bob PubK-Bob	ScriptSig
Sequence number	Sequence number
4	Number of outputs
10,000,000 satoshi	Output value
Output script length	Output script length
OP_HASH160 <redeem script hash>OP_EQUAL	Output script
5,000,000 satoshi	Output value
Output script length	Output script length
OP_HASH160 <redeem script hash>OP_EQUAL	Output script
10,000 satoshi	Output value
Output script length	Output script length
OP_DUP OP_HASH160 <PubK-Server hash> OP_EQUALVERIFY OP_CHECKSIG	Output script
880,000 satoshi	Output value
Output script length	Output script length
OP_DUP OP_HASH160 <PubK-Bob hash> OP_EQUALVERIFY OP_CHECKSIG	Output script
LockTime	LockTime

Fig. 7

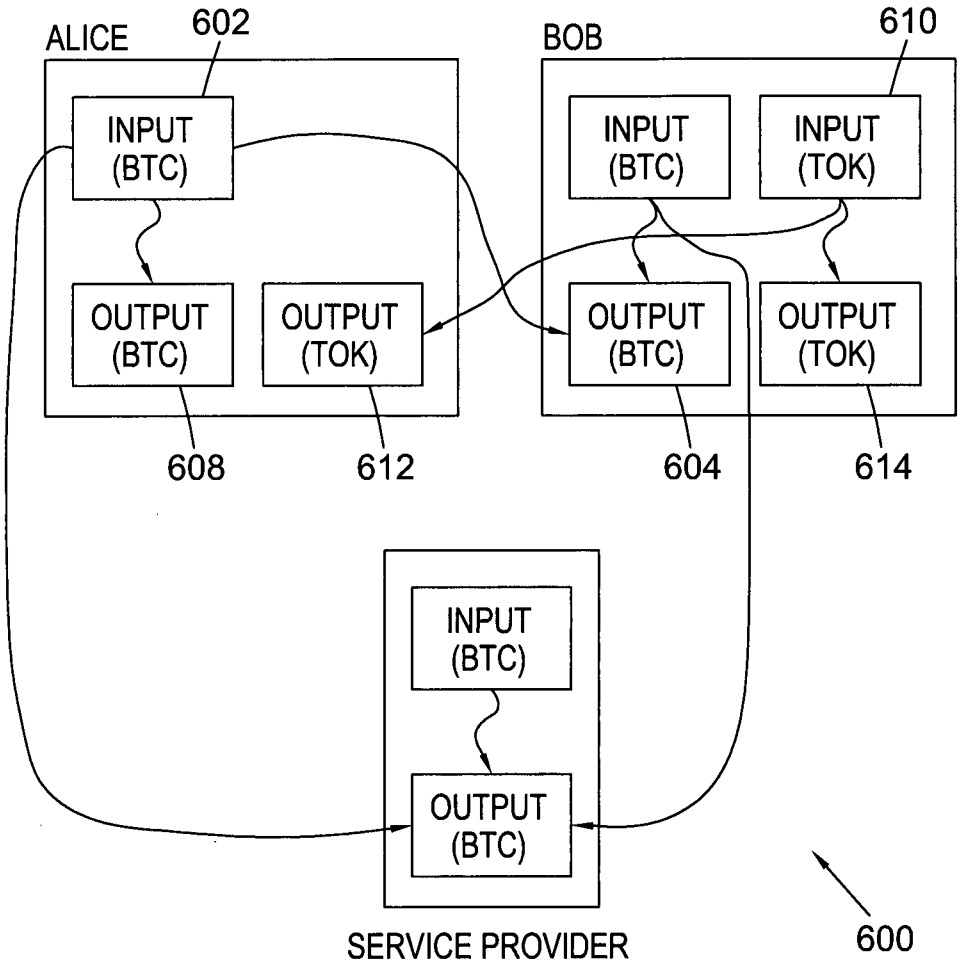
ALICE
602
INPUT (BTC)
OUTPUT (BTC)
OUTPUT (TOK)
608
612
BOB
610
INPUT (BTC)
INPUT (TOK)
OUTPUT (BTC)
OUTPUT (TOK)
604
614
INPUT (BTC)
OUTPUT (BTC)
SERVICE PROVIDER
600

Fig. 8

Alice sends BTC to Bob	
Transaction-ID	Trancaction-ID
Version number	Version number
1	Number of inputs
ALICE00010	Prev Trans Output
IDX-00	Prev Trans Output index
Script length	Script length
Sig-Alice PubK-Alice	ScriptSig
Sequence number	Sequence number
3	Prev Trans Output
246,913,580 satoshi	Prev Trans Output index
Output script length	Script length
OP_DUP OP_HASH160 <PubK-Bob hash> OP_EQUALVERIFY OP_CHECKSIG	ScriptSig
2,469,136 satoshi	Sequence number
Output script length	Number of outputs
OP_DUP OP_HASH160 <PubK-Server hash> OP_EQUALVERIFY OP_CHECKSIG	Output value
50,616,284 satoshi	Output script length
Output script length	Output script
OP_DUP OP_HASH160 <PubK-Alice hash> OP_EQUALVERIFY OP_CHECKSIG	Output value
LockTime	Output script length

Bob sends token to Alice	
Transaction-ID	Trancaction-ID
Version number	Version number
2	Number of inputs
BOB00010	Prev Trans Output
IDX-00	Prev Trans Output index
Script length	Script length
Sig-Bob Sig-Issuer < 2 metadata1 metadata2 PubK-Bob PubK-Issuer 4 OP_CHECKMULTSIG>	ScriptSig
Sequence number	Sequence number
BOB00010	Prev Trans Output
IDX-01	Prev Trans Output index
Script length	Script length
Sig-Bob PubK-Bob	ScriptSig
Sequence number	Sequence number
4	Number of outputs
10,000,000 satoshi	Output value
Output script length	Output script length
OP_HASH160 <redeem script hash>OP_EQUAL	Output script
5,000,000 satoshi	Output value
Output script length	Output script length
OP_HASH160 <redeem script hash>OP_EQUAL	Output script
10,000 satoshi	Output value
Output script length	Output script length
OP_DUP OP_HASH160 <PubK-Server hash> OP_EQUALVERIFY OP_CHECKSIG	Output script
880,000 satoshi	Output value
Output script length	Output script length
OP_DUP OP_HASH160 <PubK-Bob hash> OP_EQUALVERIFY OP_CHECKSIG	Output script
LockTime	LockTime

Fig. 9

AB00010	Trancaction-ID
Version number	Version number
3	Number of inputs
ALICE00010	Prev Trans Output
IDX-00	Prev Trans Output index
Script length	Script length
Sig-Alice PubK-Alice	ScriptSig
Sequence number	Sequence number
BOB00010	Prev Trans Output
IDX-00	Prev Trans Output index
Script length	Script length
Sig-Bob Sig-Issuer < 2 metadata1 metadata2 PubK-Bob PubK-Issuer 4 OP_CHECKMULTSIG>	ScriptSig
Sequence number	Sequence number
BOB00010	Prev Trans Output
IDX-01	Prev Trans Output index
Script length	Script length
Sig-Bob PubK-Bob	ScriptSig
Sequence number	Sequence number
7	Number of outputs
246,913,580 satoshi	Output value
Output script length	Output script length
OP_DUP OP_HASH160 <PubK-Bob hash> OP_EQUALVERIFY OP_CHECKSIG	Output script
2,469,136 satoshi	Output value
Output script length	Output script length
OP_DUP OP_HASH160 <PubK-Server hash> OP_EQUALVERIFY OP_CHECKSIG	Output script
50,616,284 satoshi	Output value
Output script length	Output script length
OP_DUP OP_HASH160 <PubK-Alice hash> OP_EQUALVERIFY OP_CHECKSIG	Output script
10,000,000 satoshi	Output value
Output script length	Output script length
OP_HASH160 <redeem script hash>OP_EQUAL	Output script
5,000,000 satoshi	Output value
Output script length	Output script length
OP_HASH160 <redeem script hash>OP_EQUAL	Output script
10,000 satoshi	Output value
Output script length	Output script length
OP_DUP OP_HASH160 <PubK-Server hash> OP_EQUALVERIFY OP_CHECKSIG	Output script
880,000 satoshi	Output value
Output script length	Output script length
OP_DUP OP_HASH160 <PubK-Ali hash> OP_EQUALVERIFY OP_CHECKSIG	Output script
LockTime	LockTime

Fig. 10

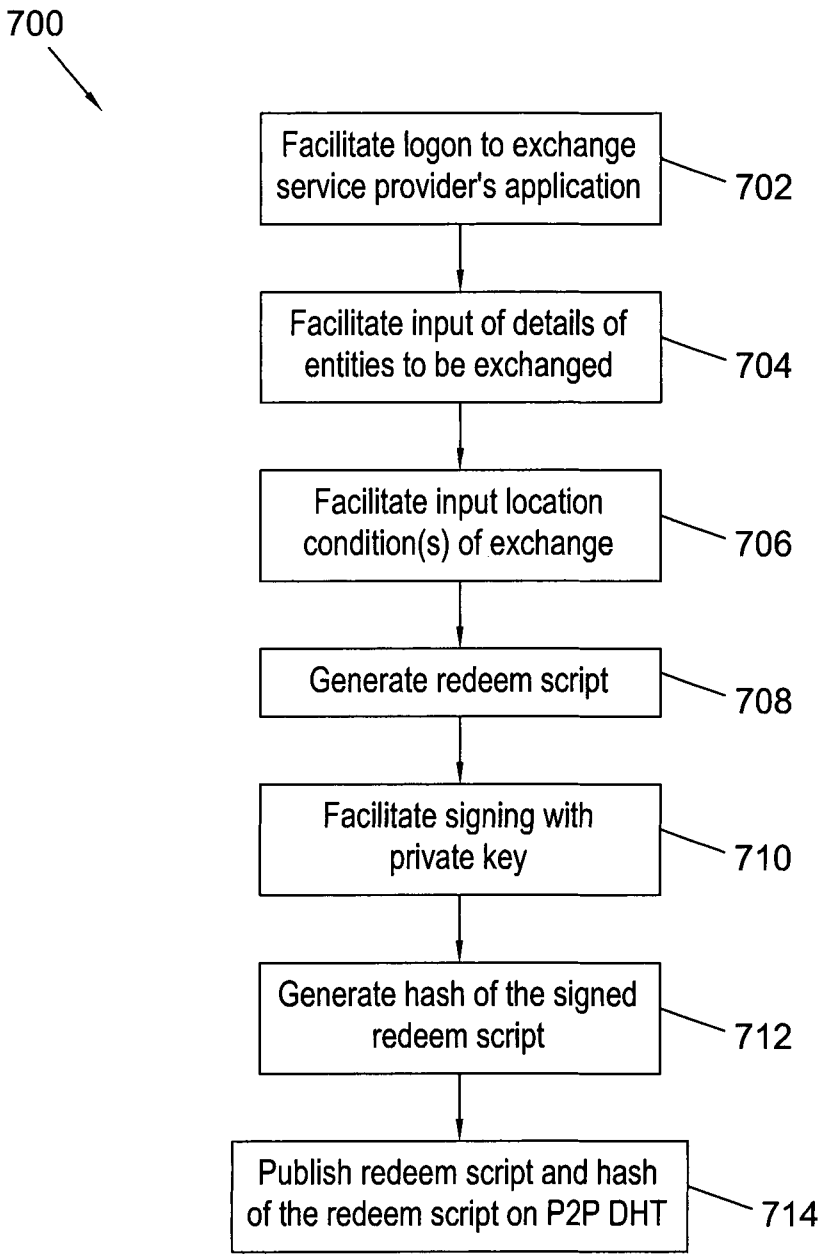
700
Facilitate logon to exchange service provider's application
702
Facilitate input of details of entities to be exchanged
704
Facilitate input location condition(s) of exchange
706
Generate redeem script
708
Facilitate signing with private key
710
Generate hash of the signed redeem script
712
Publish redeem script and hash of the redeem script on P2P DHT
714

Fig. 11

	Bob
Offer	Gold Nugget
Request	BTC
Offer-type-code	0001_2
Offer-QTY-max	1
Offer-QTY-min	1
Offer-item-ID	GN1234
Rate-min	NULL (no minimum)
Rate-max	NULL (no maximum)
Request-type-code	0000_2
Request-item-ID	
Request-QTY-max	NULL
Request-QTY-min	NULL
Condition	0004
Location Code	0101
Range Code	01
Range	20
Location	52.600120, -1.349945
Explanation of the offer	I am offering one gold nugget in exchange for Bitcoins at my store the Gold Emporium

Fig. 12

Field	Sub-field	Bytes	Value	Comments
MetadataX	ConditionCode	2	0x0000	Value indicates the condition as defined by a DHT of condition codes
	Data	30		Format depends on the Specific Condition code

Fig. 13

Field	Sub-field	Bytes	Value	Comments
MetadataX	ConditionCode	2	0x0004	Value indicates LOCATION specific exchange
	LocationCode	2	0x0101	Example: Value indicates location is expressed as GPS coordinates
	RangeCode	1	0x01	Example: Value indicates range is in metres
	Range	1	0x64	Integer value expressed in binary. Example: 100 metres
	Location	26		Location - expressed as required by the LocationCode

Fig. 14

	Alice
Offer	BTC
Request	Gold Nugget
Offer-type-code	0000_2
Offer-QTY-max	1
Offer-QTY-min	1
Offer-item-ID	
Rate-min	1.00×10^2 satoshi/gold nugget
Rate-max	NULL (no maximum)
Request-type-code	0001_2
Request-item-ID	GN1234
Request-QTY-max	1
Request-QTY-min	1
Explanation of the offer	I will buy one gold nugget in exchange for 100 BTC

Fig. 15

Alice sends token to Carol	
ALI00010	Trancaction-ID
Version number	Version number
2	Number of inputs
ALI00010	Prev Trans Output
IDX-00	Prev Trans Output index
Script length	Script length
Sig-Ali Sig-Issuer < 2 metadata1 metadata2 PubK-Ali PubK-Issuer 4 OP_CHECKMULTSIG>	ScriptSig
Sequence number	Sequence number
ALI00010	Prev Trans Output
IDX-01	Prev Trans Output index
Script length	Script length
Sig-Ali PubK-Ali	ScriptSig
Sequence number	Sequence number
4	Number of outputs
10,000,000 satoshi	Output value
Output script length	Output script length
OP_HASH160 <redeem script hash>OP_EQUAL	Output script
5,000,000 satoshi	Output value
Output script length	Output script length
OP_HASH160 <redeem script hash>OP_EQUAL	Output script
10,000 satoshi	Output value
Output script length	Output script length
OP_DUP OP_HASH160 <PubK-Server hash> OP_EQUALVERIFY OP_CHECKSIG	Output script
880,000 satoshi	Output value
Output script length	Output script length
OP_DUP OP_HASH160 <PubK-Ali hash> OP_EQUALVERIFY OP_CHECKSIG	Output script
LockTime	LockTime

Alice pays Carol $1000 in tokenised CAD (expecting her GBP from Bob at a rate of AUD$2/GBP)
Alice pays herself the remainder of tokenised CAD (i.e. change) = $1500-$1000
Using regular (untokenised BTC) Alice pays the Service Provider's fee (say, flat rate = 10000 sat)
Using regular (untokenised BTC) Alice pays herself change from 900,000 leaving 1000 for miner

Fig. 16A

Bob sends token to Alice	
BOB00030	Trancaction-ID
Version number	Version number
2	Number of inputs
BOB00010	Prev Trans Output
IDX-00	Prev Trans Output index
Script length	Script length
Sig-Bob Sig-Issuer < 2 metadata1 metadata2 PubK-Bob PubK-Issuer 4 OP_CHECKMULTSIG>	ScriptSig
Sequence number	Sequence number
BOB00010	Prev Trans Output
IDX-01	Prev Trans Output index
Script length	Script length
Sig-Bob PubK-Bob	ScriptSig
Sequence number	Sequence number
4	Number of outputs
10,000,000 satoshi	Output value
Output script length	Output script length
OP_HASH160 <redeem script hash>OP_EQUAL	Output script
5,000,000 satoshi	Output value
Output script length	Output script length
OP_HASH160 <redeem script hash>OP_EQUAL	Output script
10,000 satoshi	Output value
Output script length	Output script length
OP_DUP OP_HASH160 <PubK-Server hash> OP_EQUALVERIFY OP_CHECKSIG	Output script
880,000 satoshi	Output value
Output script length	Output script length
OP_DUP OP_HASH160 <PubK-Bob hash> OP_EQUALVERIFY OP_CHECKSIG	Output script
LockTime	LockTime

Bob pays Alice GBP500 in tokenised GBP (expecting his AUD at a rate of $2/GBP)
Bob pays himself the remainder of tokenised GBP (i.e. change) = GBP750-GBP500
Using regular (untokenised BTC) Bob pays the Service Provider's fee (say, flat rate = 10000 sat)
Using regular (untokenised BTC) Bob pays himself change from 900,000 leaving 1000 for miner

Fig. 16B

Carol sends token to Bob	
CAR00020	Trancaction-ID
Version number	Version number
2	Number of inputs
CAR00010	Prev Trans Output
IDX-00	Prev Trans Output index
Script length	Script length
Sig-Car Sig-Issuer < 2 metadata1 metadata2 PubK-Car PubK-Issuer 4 OP_CHECKMULTSIG>	ScriptSig
Sequence number	Sequence number
CAR00010	Prev Trans Output
IDX-01	Prev Trans Output index
Script length	Script length
Sig-Car PubK-Car	ScriptSig
Sequence number	Sequence number
4	Number of outputs
10,000,000 satoshi	Output value
Output script length	Output script length
OP_HASH160 <redeem script hash>OP_EQUAL	Output script
5,000,000 satoshi	Output value
Output script length	Output script length
OP_HASH160 <redeem script hash>OP_EQUAL	Output script
10,000 satoshi	Output value
Output script length	Output script length
OP_DUP OP_HASH160 <PubK-Server hash> OP_EQUALVERIFY OP_CHECKSIG	Output script
880,000 satoshi	Output value
Output script length	Output script length
OP_DUP OP_HASH160 <PubK-Ali hash> OP_EQUALVERIFY OP_CHECKSIG	Output script
LockTime	LockTime

Carol pays Bob $1000 in tokenised AUD (expecting her CAD at a rate of $1CAD/AUD)
Carol pays herself the remainder of tokenised AUD (i.e. change) = $1500-$1000
Using regular (untokenised BTC) Carol pays the Service Provider's fee (say, flat rate = 10000 sat)
Using regular (untokenised BTC) Carol pays herself change from 900,000 leaving 1000 for miner

Fig. 16C

INTERNATIONAL SEARCH REPORT

International application No
PCT/IB2017/050861

A. CLASSIFICATION OF SUBJECT MATTER
INV. G06Q20/38
ADD.

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)
H04L G06Q

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)
EPO-Internal , WPI Data

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	Joseph Poon ET AL: "The Bitcoin Lightning Network: Scalable Off-Chain Instant Payments" , , 14 January 2016 (2016-01-14) , XP055358006, Retrieved from the Internet: URL: https://lightning.network/lightning-network-paper.pdf [retrieved on 2017-03-23] Sections 2, 3, 3.3, 5 -----	1-23
X	US 2015/363773 A1 (RONCA JAMES G [US] ET AL) 17 December 2015 (2015-12-17) paragraphs [0057] , [0062] , [0067] , [0083] , [0179] , [0209] -----	1-23

☐ Further documents are listed in the continuation of Box C.

☒ See patent family annex.

* Special categories of cited documents :

"A" document defining the general state of the art which is not considered to be of particular relevance

"E" earlier application or patent but published on or after the international filing date

"L" documentwhich may throw doubts on priority claim(s) orwhich is cited to establish the publication date of another citation or other special reason (as specified)

"O" document referring to an oral disclosure, use, exhibition or other means

"P" document published prior to the international filing date but later than the priority date claimed

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art

"&" document member of the same patent family

Date of the actual completion of the international search	Date of mailing of the international search report
27 March 2017	10/04/2017

Name and mailing address of the ISA/	Authorized officer
European Patent Office, P.B. 5818 Patentlaan 2 NL - 2280 HV Rijswijk Tel. (+31-70) 340-2040, Fax: (+31-70) 340-3016	Frank, Mario

INTERNATIONAL SEARCH REPORT

Information on patent family members

International application No
PCT/IB2017/050861

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
US 2015363773 A1	17-12-2015	NONE	