Title: CONTROLLED RESOURCE PROVISIONING IN DISTRIBUTED COMPUTING ENVIRONMENTS

Abstract: A computer implemented method to provide allocation of one or more computing resources for a consumer computing component, each resource having a resource type and being provided by one or more resource providers, and the consumer having associated a quantity of tradeable value constraining an extent of resource consumption, the method comprising: defining one or more consumption rules for each of the resource providers, each consumption rule being associated with the consumer and defining: a type of resource; and one or more tradeable values corresponding to costs for consumption of resource, each tradeable value being associated with a particular configuration of the resource, wherein the consumption rules are recorded in a blockchain data structure; defining an optimisation rule associated with the consumer and including: a reference to each of the consumption rules associated with the consumer; a definition of one or more constraints on characteristics of resources; and a consumption optimisation function, wherein the optimisation rule is recorded in the blockchain, wherein, in use, a transaction is submitted to the blockchain to trigger consumption of one or more required types of resource and responsive to the transaction the optimisation rule determines one or more resource providers for providing resources of the required resource types and, for each determined provider, the optimisation function determines, based on the consumption rules, a particular configuration and an extent of consumption of each type of resource so as to control expenditure of the consumer's tradeable value.
Controlled Resource Provisioning in Distributed Computing Environments

The present invention relates to the provision of resources. In particular it relates to the provision of resources in distributed computing environments.

Distributed computing environments are environments in which computer systems, services and/or resources (whether hardware, software or a combination) are distributed physically and/or virtually with a dependence on communications networks for interoperability. Further, there is increasing deployment of software installations such as applications or entire virtualised computer systems to service based environments, such as Infrastructure as a Service (IaaS), Platform as a Service (PaaS) and cloud computing environments.

Software installations such as virtualised computer systems, cloud computing deployments and the like, implement functional requirements as part of the installation. For example functional requirements can include: security functionality such as anti-malware, encryption, firewall or intrusion detection; communications functionality such as network communication, encrypted or otherwise secure communication; operating system services and functions; data storage facilities including disk, memory, database and the like; middleware services such as message handling, transaction handling and the like; and other functional requirements. Functional requirements themselves employ computing resources such as storage, processors, security functions, communication resources such as networking and the like that are increasingly provided by “supplier” or “provider” entities (whether third-party or not) either as actual resources or virtualised resources. Thus in implementing or providing a functional requirement a software installation consumes requisite computing resources from resource providers. The consumption of such resources can be based on resource reservation, dynamic, on-demand and/or dynamically scalable.

Resource providers and consumers in such environments can be unrelated, anonymous, untrusted and/or unknown to each other.

A challenge arises for a resource consumer such as a cloud computing consumer having a virtual machine specification for implementation. Such a resource consumer must procure computing resource to implement functional requirements. It is necessary to identify resource providers able to satisfy the resource requirements to, in turn, satisfy the particular functional requirements. Multiple such resource providers may be identified for each resource.

Resource providers can provision resource in different ways and with different associated costs. An on-demand approach to resource provisioning provides access to resources by resource consumers as required without prior reservation. To accommodate such an
approach to resource provisioning resource providers must employ scalable architectures and may operate with costly redundancy when demand is low. Accordingly these additional burdens on resource providers are shared with resource consumers by imposing higher cost. An alternative approach to resource provisioning is reservation-based provisioning in which resource consumers arrange resource provision in advance of consumption. This provides certainty for resource providers and so the costs to consumers are lower. However, reservation-based provisioning is susceptible to over or under consumption by resource consumers where resource planning or load forecasting is not effective or accurate. These challenges for resource providers and consumers are compounded where resource demand across all consumers and the cost of providing resources varies over time.

Accordingly it would be beneficial to provide for the consumption of resource by resource consumers in a way that mitigates these challenges.

The present invention accordingly provides, in a first aspect, a computer implemented method to provide allocation of one or more computing resources for use by a consumer computing component, each resource having a resource type and being provided by one or more resource providers, and the consumer having associated a quantity of tradeable value constraining an extent of resource consumption, the method comprising: defining one or more consumption rules for each of the resource providers, each consumption rule being associated with the consumer and defining: a type of resource; and one or more tradeable values corresponding to costs for consumption of resource, each tradeable value being associated with a particular configuration of the resource, wherein the consumption rules are recorded in a blockchain data structure; defining an optimisation rule associated with the consumer and including: a reference to each of the consumption rules associated with the consumer; a definition of one or more constraints on characteristics of resources; and a consumption optimisation function, wherein the optimisation rule is recorded in the blockchain, wherein, in use, a transaction is submitted to the blockchain to trigger consumption of one or more required types of resource and responsive to the transaction the optimisation rule determines one or more resource providers for providing resources of the required resource types and, for each determined provider, the optimisation function determines, based on the consumption rules, a particular configuration and an extent of consumption of each type of resource so as to control expenditure of the consumer’s tradeable value.

Preferably the optimisation rule further includes one or more constraints on expenditure of the consumer’s tradeable value.
Preferably the optimisation rule effects consumption of resource by causing invocation of one or more consumption rules for the consumer corresponding to the one or more determined resource providers.

Preferably the invocation of each of the one or more consumption rules effects a change to the quantity of tradeable value of the consumer.

Preferably each of the one or more consumption rules are executable by miner software components operating with the blockchain data structure.

Preferably the one or more consumption rules are contracts defined in a smart contract blockchain system.

Preferably the optimisation rule is a contract defined in a smart contract blockchain system.

Preferably the optimisation rule responsive to the transaction is executed repeatedly so as to continuously adapt resource consumption.

Preferably the optimisation rule is executable by miner software components operating with the blockchain data structure.

Preferably the optimisation rule is executed by a central computer system.

Preferably the determinations of the optimisation rule responsive to the transaction are repeated based on one or more of: changes to the consumer’s quantity of tradeable value; and changes to a tradeable value, in one or more consumption rules, corresponding to a cost for consumption of resource.

Preferably a consumption rule evaluates a tradeable value for consumption of a resource with reference to an interface to a service provider for the resource, the interface providing a dynamic tradeable value rate for the resource.

The present invention accordingly provides, in a second aspect, a computer system to provide allocation of one or more computing resources for use by a consumer computing component, each resource having a resource type and being provided by one or more resource providers, and the consumer having associated a quantity of tradeable value constraining an extent of resource consumption, the system comprising a processor and a data store wherein the processor is adapted to undertake the steps of: defining one or more consumption rules for each of the resource providers, each consumption rule being associated with the consumer and defining: a type of resource; and one or more tradeable values corresponding to costs for consumption of resource, each tradeable value being
associated with a particular configuration of the resource, wherein the consumption rules are recorded in a blockchain data structure; defining an optimisation rule associated with the consumer and including: a reference to each of the consumption rules associated with the consumer; a definition of one or more constraints on characteristics of resources; and a consumption optimisation function, wherein the optimisation rule is recorded in the blockchain, wherein, in use, a transaction is submitted to the blockchain to trigger consumption of one or more required types of resource and responsive to the transaction the optimisation rule determines one or more resource providers for providing resources of the required resource types and, for each determined provider, the optimisation function determines, based on the consumption rules, a particular configuration and an extent of consumption of each type of resource so as to control expenditure of the consumer’s tradeable value.

The present invention accordingly provides, in a third aspect, a computer program element comprising computer program code to, when loaded into a computer system and executed thereon, cause the computer to perform the steps of a method as described above.

A preferred embodiment of the present invention will now be described, by way of example only, with reference to the accompanying drawings, in which:

Figure 1 is a block diagram of a computer system suitable for the operation of embodiments of the present invention;

Figure 2 is a component diagram of an exemplary computer system arrangement for provisioning resource to a consumer computing component in accordance with an embodiment of the present invention;

Figure 3 is a component diagram illustrating the exemplary arrangement of Figure 2 in use in accordance with an embodiment of the present invention;

Figure 4 is a schematic illustration of a consumption rule in accordance with an embodiment of the present invention; and

Figure 5 is an exemplary flowchart of a method to provide computing resources for use by a consumer in accordance with an embodiment of the present invention.

Figure 1 is a block diagram of a computer system suitable for the operation of components in embodiments of the present invention. A central processor unit (CPU) 102 is communicatively connected to storage 104 and an input/output (I/O) interface 106 via a data bus 108. The storage 104 can be any read/write storage device such as a random access memory (RAM) or a non-volatile storage device. An example of a non-volatile storage device
includes a disk or tape storage device. The I/O interface 106 is an interface to devices for the input or output of data, or for both input and output of data. Examples of I/O devices connectable to I/O interface 106 include a keyboard, a mouse, a display (such as a monitor) and a network connection.

Figure 2 is a component diagram of an exemplary computer system arrangement for provisioning resource to a consumer computing component 202 in accordance with an embodiment of the present invention. The consumer 202 is a software, hardware, firmware or combination component adapted to consume computing resource. For example, the consumer 202 is a virtual machine specification or a virtual machine for instantiation and operation in a virtualised computing environment and requiring computing resources for execution, such as processor, storage, networking, input/output or other computing resource. Consumption of resource is compensated by expenditure of a tradeable value 203 associated with the consumer 202. Thus the consumer 202 has associated a quantity of tradeable value 203, such as a numerical representation of value or a balance. Consumption of a resource involves expenditure of the tradeable value 203 and thus the quantity of tradeable value 203 associated with the consumer 202 constitutes a constraint on an extent of resource consumption by the consumer 202. In one embodiment the tradeable value 203 is a cryptocurrency value or a fiat value.

The consumer 202 is connected to a computer network 200 such as a wired, wireless, physical or virtual network. Accordingly the consumer 202 is communicatively connected to the other components of Figure 2 as described below. The network 200 is depicted as a single continuous entity though it will be appreciated that the network 200 could have any communications network topology and/or arrangement. In some embodiments the network 200 is a series of communicatively connected networks or a logical arrangement of networks or subnetworks operating over potentially numerous underlying physical networks. In some embodiments the network 200 is the internet.

The arrangement of Figure 2 further includes one or more resource provider computer systems 204 each connected to the network 200. Providers 204 are hardware, software, firmware or combination components adapted to provide computing resources 214 for consumption by the consumer 202. For example, a provider 204 can be a cloud computing service offering processor, memory, storage, database, networking or other computing resources. Each resource 214 has a resource type as a class or type of resource. For example, multiple different microprocessor resources or virtualised processing component resources could have the type "processor". Notably each resource provider 204 may provide multiple resources 214 which may be of the same or different types or a mixture. Such
resources may not be physical resources and may alternatively be virtualised such as virtual storage. Such virtualisation can include simulation or emulation of resources or virtual implementations of physical resources. In some embodiments a resource 214 is not local to a provider 204 and the provider 204 acts as an agent, proxy, router or link to a resource. For example, in one embodiment a provider 204 is a resource agent that provides resources hosted by other computing entities communicatively connected to the provider 204 over the network 200 or a different network or communication means.

The arrangement of Figure 2 further includes a blockchain database 206 as a sequential transactional database that may be distributed and is communicatively connected to network 200. Furthermore, the arrangement of Figure 2 includes a plurality of miner computing components 212a, 212b, 212n. Distributed sequential transactional databases are well known in the field of cryptocurrencies and are documented, for example, in “Mastering Bitcoin. Unlocking Digital Crypto-Currencies.” (Andreas M. Antonopoulos, O’Reilly Media, April 2014). For convenience, such a database is herein referred to as a blockchain 206 though it will be appreciated that other suitable databases, data structures or mechanisms possessing the characteristics essential for embodiments of the present invention could alternatively be used. The blockchain 206 is a distributed chain of block data structures accessed by a network of nodes, referred to here as a network of miners 212. Each block in the blockchain 206 includes a plurality of transaction data structures. For example, in a preferred embodiment each blockchain includes a Merkle tree of hash or digest values for transactions included in the block to arrive at a hash value for the block, which is itself combined with a hash value for a preceding block to generate a chain of blocks (blockchain). A new block of transactions is added to the blockchain by miner 212 software, hardware, firmware or combination systems in the miner network. The miners 212 are communicatively connected to sources of transactions (such as the consumer 202) and access or copy the blockchain 206. A miner 212 undertakes validation of the substantive content of a transaction (such as the criteria defined therein) and adds a block of new transactions to the blockchain when a challenge is satisfied, typically such challenge involving a combination hash or digest for a prospective new block and a preceding block in the blockchain and some challenge criterion. Thus miners 212 in the miner network may each generate prospective new blocks for addition to the blockchain 206. Where a miner 212 satisfies or solves the challenge and validates the transactions in a prospective new block such new block is added to the blockchain 206. Accordingly the blockchain 206 provides a distributed mechanism for reliably verifying a data entity such as an entity constituting or representing the potential to consume a resource.
While the detailed operation of blockchains and the function of miners 212 in the miner network is beyond the scope of this specification, the manner in which the blockchain 206 and network of miners 212 operate ensures that only valid transactions are added within blocks to the blockchain 206 in a manner that is persistent within the blockchain 206. Transactions added erroneously or maliciously are not verifiable by other miners 212 in the network and do not persist in the blockchain. This attribute of blockchains is exploited by embodiments of the present invention to provide a distributed and reliable assurance for resource providers 204 and resource consumer 202 to improve efficiency of resource consumption. Thus transactions submitted for recordal in the blockchain 206 are passed to the miner network for validation by miners 212 as prospective new blocks. Validated blocks are added to the blockchain 206 by the miner network. Blocks added to the blockchain 206 that are invalid (due to error or malice) do not persist in the blockchain in favour or blocks verifiable by other miners in the network. Thus after a period of time (the length of which can be tailored by, for example, adapting the complexity of the challenge required to demonstrate proof of work by the miners 212 as part of the creation of new blocks), a new block is confirmed in the blockchain 206 at which time entities utilising the blockchain 206 can operate with certainty that transactions in the confirmed block are valid and verifiable.

In one embodiment the arrangement of Figure 2 is implemented within one or a number of closely coupled computer systems and the network 200 is provided as a communication means between the various components of Figure 2 such as a communications bus or the like. In some embodiments the arrangement of Figure 2 is implemented in a controlled environment (for example a non-public environment) such that the blockchain is accessible to the other components that also execute in the controlled environment. For example, the arrangement could be implemented within a single organisation or a group of collaborating organisations using an intranet of the organisation(s).

The arrangement of Figure 2 is arranged to provide at least some of resources 214 for use by the consumer 202 in such a way as to control the expenditure of the consumer’s tradeable value as will be described with respect to Figure 3. Figure 3 is a component diagram illustrating the exemplary arrangement of Figure 2 in use in accordance with an embodiment of the present invention. Many of the elements of Figure 3 are identical to those described above with respect to Figure 2 and these will not be repeated here.

Initially the consumer 202 defines one or more consumption rules 208 as a data structure for storage in the blockchain 206. Each consumption rule 208 is provided in response to an agreement or determination that a provider 204 can provide access to a resource 214. Such determination includes a definition of, or a mechanism for determining, a cost of consumption
of the resource 214. Thus, provider 204a may agree to supply consumer 202 with access to a virtualised processor resource having a particular processing performance (e.g. processing throughput). The provider 204a may define a fixed cost for consumption of the processor resource and a consumption rule 208 can be defined to identify the resource and the cost. Alternatively, in a context where the cost of consumption of the processor resource is variable, the provide 204a may provide an interface such as an application programming interface via which an up-to-date cost for consumption of the resource can be obtained. For example, cost can vary temporally, based on resource load, provider capacity and the like. Furthermore, the provider 204 may offer variants of the virtualised processor resource, such as a virtualised processor resource having different configurations. For example the provider 204 may offer multiple virtualisations of different physical processors and/or access to actual physical processors, with all such resources being processors and so belonging to the same resource type. Each configuration of the resource may be have a different associated cost that can be reflected in consumption rules 208.

Accordingly a consumption rule 208 defines logic for validating consumption by a consumer 202 ("CON") of one or more resources of a resource provider 204 ("PRV"). The validation is undertaken by the definition of logical rules within the consumption rule 208 that reflect the requirements of the consumer 202, such as preferences of the consumer 202 or a user of the consumer 202 or policy or other requirements of the consumer 202, such as organisational policy, security policy, performance requirements and the like. In one embodiment the logical rules are implemented in executable program code and serve to test attributes of a proposal to consume a resource for satisfaction of the rules. Logical rules identify a particular type of resource ("RSC") and define a particular or range of costs for consumption of a resource (i.e. tradeable values corresponding to costs for consumption of the resource, "VAL"). For example, the consumer 202 may have a requirement to consume communication network resource. Accordingly, the consumer negotiates with a provider, such as provider 204a, and defines a consumption rule 208a identifying a resource type (communications network) and resource consumption costs imposed by the provider 204a. Such costs can be associated with one or more particular configurations of the resource.

Thus where the provider 204a offers communication network resource at multiple performance levels (e.g. multiple data transfer speeds), each performance level constituting a configuration of the resource can have associated a different cost.

Each consumption rule 208 also includes logic for effecting expenditure of tradeable value 203 when consumption is validated in response to a consumption transaction or message.

Thus, in use, a transaction submitted to the blockchain 206 for, or a message received directly by, a particular consumption rule 208a (such as by reference to an address of the
consumption rule 208a) and including transaction/message parameters indicating consumption of a particular resource, will be received and validated by the consumption rule 208a. If the parameters are validated by the consumption rule 208a (i.e. the consumption is valid and the rule is satisfied) then the consumption rule 208a effects expenditure of the consumer’s 202 tradeable value 203. Expenditure can be effected by reducing the tradeable value 203 typically with a corresponding increase in a tradeable value for a resource provider 204 providing the consumed resource. The execution of the consumption rule 208a will be undertaken by miners 212 as part of the mining process.

Figure 4 is a schematic illustration of an exemplary consumption rule 208 in accordance with an embodiment of the present invention. The consumption rule 208 includes an identification of the consumer 202 (“CON”) and a resource provider 204 (“PRV”). The consumption rule 208 further includes a list of resource types 320a, 320b, 320c agreed for provisioning by the provider 204 to the consumer 202. For example, resource types include processor, storage, communication etc. Each resource type includes a resource configuration 322 and a cost 324. Notably the cost 324 could be provided by way of an interface to real-time cost information. Multiple consumption rules 208a, 208b, 208c, 208n can be defined by the consumer 202 for resources of the providers 204. Notably, multiple consumption rules 208 for the same type of resource can be provided, such as where each rule corresponds to a different provider of the resource type.

Returning to Figure 3, the consumer 202 further defines an optimisation rule 210 as a data structure for storage in the blockchain 206. The optimisation rule 210 includes a reference, link or other mechanism for access to each of the consumption rules 208 for the consumer 202, such as an address for each of the consumption rules 208 in a storage of the blockchain system. The optimisation rule 210 includes constraint definitions such as rules defining constraints specific to the consumer 202 on the characteristics of resources, such as characteristics of particular resource types. Preferably the optimisation rule 210 further includes constraint definitions defining constraints specific to the consumer 202 on expenditure of the consumer’s 202 tradeable value 203, such as a limit to the extent or rate of expenditure or a relationship between expenditure rules and characteristics of resource types. In this way the expenditure of the consumer’s 202 tradeable value 203 by the consumption of resources 214 is bounded by the constraints.

Constraints on characteristics of resources can include definitions of minimum and/or maximum service levels, qualities, configurations or attributes of resources. Additionally or alternatively constraints on characteristics of resources can include rules defining characteristics of certain resources or resource types in dependence on characteristics of
other resources or resource types or even in dependence on an amount or proportion of tradeable value 203, such as an amount of tradeable value 203 available. For example, where the consumer 202 has resource requirements for network communication resource and data storage resource, the optimisation rule 210 can define a resource constraint that requires data storage resource of higher capacity when a network communication resource is provided having a high data rate (so as to ensure capacity to store data communicated via the communication resource, for example). In another example, a constraint on a maximum data rate for a network communication resource may cause a reduction in data rate when a quantity of tradeable value 203 of the consumer 202 falls below a predetermined threshold.

In this way a detection of a low balance of tradeable value 203 can lead to a reduction in a rate of expenditure by transitioning to lower cost resources. Other resource characteristic constraints will be apparent to those skilled in the art.

Constraints on expenditure of the consumer's 202 tradeable value 203 can be dependent on, associated with or related to resource characteristics, resource configurations, resource constraints or the quantity, balance or rate of expenditure of tradeable value itself. For example, a maximum tradeable value that can be expended on a communication resource may be defined differently for communication resources of different data rates, with higher data rates enjoying a greater maximum tradeable value.

For example, the consumer 202 may have a policy requirement that network resource must have a configuration that provides a particular performance, such as a minimum data transfer speed. Further, the consumer 202 may have a policy requirement that the network resource must not exceed a particular cost, such cost affecting the consumer 202 by expenditure of the tradeable value 203. Such requirements of the consumer 202 are reflected in the optimisation rule 210 as constraints.

The optimisation rule 210 further includes an optimisation function as executable logic for controlling expenditure of the consumer’s 202 tradeable value 203 by determining resources for consumption by the consumer 202, particular configurations of the resources and an extent of consumption of each resource. In preferred embodiments it is an overriding objective of the optimisation function to determine resources and configurations of resources for consumption by the consumer 202 that satisfy the constraints of the optimisation rule 210 (the resource constraints and, in some embodiments, the expenditure constraints) while minimising the expenditure of tradeable value 203 so as, for example, to provide for the longest duration of consumption for the tradeable value 203. Other optimisation objectives could alternatively be defined for the optimisation function as will be apparent to those skilled in the art. For the avoidance of doubt, while the term “optimisation function” and “optimisation
rule” are used in this specification, such function and rule may not obtain a definitive optimal solution and the terms “optimal”, “optimisation” and “optimum” are to be interpreted as indicating a tendency towards a local or global optimum by, for example, progressively identifying an improved configuration such as to achieve a local maxima or minima, where improved configurations en route to such local maxima/minima can be employed as optima in scenarios where optimisation algorithms are given determined stopping conditions based on, for example, duration or resource consumption. Such meaning will be apparent to those skilled in the art.

Thus in use the optimisation function operates based on the consumption rules 208 to identify particular resources (from particular resource providers 204) and particular resource configurations for consumption by the consumer 202. The optimisation function uses the references to the consumption rules 208 for the consumer 202 to determine the resource consumption options for the consumer 202 and applies the resource and (optional) expenditure constraints to the resource consumption options to determine a configuration of resource consumption so as to control expenditure of the consumer’s 202 tradeable value 203. Access to the consumption rules 208 by the optimisation function of the optimisation rule 210 can be achieved in a number of ways. In one embodiment the optimisation rule 210 is able to obtain, copy or read the consumption rules 208 such as by receiving or accessing the consumption rules 208 from the blockchain 206 as an arbitrary length byte array. Such access may be achieved by a function of the blockchain environment or database. Alternatively, each consumption rule 208 may be executable in a number of different modes such as: a first mode to validate resource consumption and effect expenditure of tradeable value 203; and a second mode to validate resource consumption without effecting expenditure of tradeable value 203 and return an extent of consumption that would be realised were the consumption to take place. A consumption 208 rule operating in such second mode of operation can provide an output (or return value) to a caller (such as the optimisation function of the optimisation rule 210) to indicate how much tradeable value 203 would be expended for the resource consumption (where, for example, zero expenditure can indicate a failure to validate the consumption). Thus in this way the optimisation function of the optimisation rule 210 is able to determine all or a subset of the consumption options for the consumer 202 as a basis for an optimisation of ressource consumption. Notably the process of obtaining relevant consumption options can, in some embodiments, be improved to provide greater efficiency and reduced overhead using various techniques. For example, a registry or database of consumption rules 208 for a consumer 202 can be generated and maintained by or for reference by the optimisation rule 210. Furthermore, such a registry or database can be supplemented by summary information or metadata indicating particular
resource types, resource configurations and/or cost information for the consumption rules 208 so as to provide a ready data structure on which to operate the optimisation function. The identification of subsets of consumption rules 208 being applicable to the resource or expenditure constraints of the consumer 202 can also be achieved, such as by reference to a registry or database as described above, by shortlisting consumption rules or excluding inappropriate consumption rules and deduplicating consumption rules that overlap in extent (e.g. excluding from consideration duplicate consumption rules differing only by cost and/or provider, with a preference for retaining lower cost consumption rules).

Once consumption options are determined for the consumer 202 the optimisation function is operable to determine a particular resource consumption configuration that satisfies the resource and (optionally) expenditure constraints defined in the optimisation rule 210. Such an optimisation function can employ any suitable optimisation algorithm such as the algorithms considered in the paper "Optimization of Resource Provisioning Cost in Cloud Computing" (Chaisiri, Lee and Niyato, IEEE Transactions on Services Computing, Vol. 5, No. 2, April-June 2002). For example, the Benders decomposition algorithm ("Partitioning Procedures for Solving Mixed Variables Programming Problems", Benders, J., Numerische Mathematik, 4, 238 – 252, 1962). can be applied to solve a stochastic programming problem for minimising the consumer’s 202 total expenditure. Such an approach involves breaking down the optimisation problem into smaller problems which can be solved independently or in parallel. Alternative optimisation algorithms could be employed such as finitely terminating algorithms, convergent (iterative) algorithms and/or heuristic approaches. For example, gradient-based methods such as conjugate gradient methods, gradient descent, hill-climbing and the like. Alternatively, the simplex algorithm or combinatorial algorithms can be employed. Any such appropriate optimisation algorithm operates on the basis of the resource consumption options identified by the optimisation rule 210 and identifies a particular resource consumption configuration for the consumer 203, such as a configuration that minimises expenditure of the tradeable value.

The determined resource consumption configuration includes an identification of resources to be consumed and the applicable consumption rule for each resource for consumption. Preferably the configuration further includes an extent or amount of each resource for consumption. On determination of a resource consumption configuration the optimisation rule 210 effects the resource consumption by transacting with, or messaging, the applicable consumption rules 208 for the resource consumption configuration. In this way consumption of each resource is validated by the optimisation rules 208 and expenditure of the consumer’s 202 tradeable value 203 is effected. To effect the resource provisioning or consumption itself the optimisation rule 210 informs the consumer 202 and/or the appropriate
resource providers 204 of the resource consumption configuration (at least insofar as it affects the particular providers 204) so that resource provisioning can take place.

The optimisation rule 210 residing in the blockchain 206 can be invoked and executable in a number of different ways. In one embodiment the optimisation rule 210 is executed solely by miners 212 in the blockchain environment in response to transactions or messages directed to the optimisation rule 210. In an alternative embodiment dedicated, specialised or designated computing resource is employed for the execution of the optimisation rule 210, such resource accessing the blockchain 206. For example, a computing system (whether distributed or not) such as a centralised, conventional, mainframe or supercomputer system can be dedicated to the execution of the optimisation rule 210 without depending on the provision of transactions directed to the optimisation rule 210 and without requiring miners 212 (such miners 212 typically requiring compensation as reward for their mining services in blockchain environments). The optimisation rule 210 is intended to be invoked multiple times in order that the optimisation function generates a new resource consumption configuration each time based on a new invocation context. The invocation context includes, for example, a quantity of tradeable value 203 remaining for the consumer; any updated or changed constraints; changes to consumption rules; resource costs determined via consumption rules; changes to resource availability or load; and other such characteristics as will be apparent to those skilled in the art. Thus the optimisation rule 210 is preferably executed initially and execution is subsequently repeatedly.

In one embodiment an initial invocation of the optimisation rule 210 is achieved by way of an initial resource consumption transaction directed to the optimisation rule 210 and submitted by the consumer 202 to the blockchain 206. In such an embodiment the optimisation rule 210 is invoked by miners 212 as part of processing the transaction. Alternatively, an initial invocation of the optimisation rule 210 can be automatic on generation of the optimisation rule 210 such that the optimisation rule 210 executes initially on first definition. Such automatic invocation can be achieved, for example, where a centralised or dedicated computer system is employed for execution of the optimisation rule 210. Subsequent invocations of the optimisation rule 210 can take place periodically, such as repeatedly after an elapsed time period. Such repeated invocations can be achieved by repeated transactions directed to the optimisation rule 210 and submitted to the blockchain 206 or, where a centralised or dedicated computer system is employed, by monitoring for elapse of a time period or the like. In one embodiment, each repeated invocation of the optimisation rule 210 is determined based on one or more of: changes to the consumer’s 202 quantity of tradeable value; and changes to the cost of consumption of one or more resources indicated in one or more consumption rules.
Thus in one embodiment invocation of the optimisation rule 210 is triggered by the consumer 202 by way of a transaction submitted to the blockchain and directed to the optimisation rule 210. Such a transaction will cause invocation of the logic of the optimisation rule 210 by miners 212. For example, the consumer 202 can submit a transaction addressed to the optimisation rule 210 for consumption of resource and the transaction can include data defining the resource and (optionally) expenditure constraints of the consumer. In some embodiments the transaction data also includes identification of or references to consumption rules 208 for the consumer 202. Thus in this way, in some embodiments, the optimisation rule 210 can be populated by constraints and consumption rule references as part of its invocation by parameters provided with a triggering transaction. Alternatively, such constraints and references to consumption rules 208 can be defined for the optimisation rule 210 when the optimisation rule 210 is initially defined. Where the optimisation rule 210 includes defined constraints and references, changes to the constraints and references can be achieved by replacing, updating, modifying or overwriting the optimisation rule 210 in the blockchain 206. Invocation of the optimisation rule

Figure 5 is an exemplary flowchart of a method to provide computing resources for use by the consumer 202 in accordance with an embodiment of the present invention. Prior to the method of Figure 5 the consumer 202 has agreed resource provisioning with resource providers 204, such as by way of a resource provisioning negotiation interface between the consumer 202 and the providers 204 such that the providers agree to provide resources to the consumer 202. Such provisioning can include definitions or indications of configurations of resources and cost of resources or references to interfaces for determining such costs. Initially, at step 502, the consumption rules 208 are defined by the consumer 202 based on the agreed resource provisioning and the consumption rules 208 are submitted to the blockchain 206 for validation and committal by miners 212. Subsequently the consumer 202 defines an optimisation rule 504. The optimisation rule 210 includes the optimisation function. In one embodiment the optimisation rule 210 further includes a definition of constraints (resource constraints and, optionally, expenditure constraints). In an alternative embodiment the optimisation rule 210 includes logic for receiving constraints as parameters on invocation. The optimisation rule 210 is submitted to the blockchain 206 for validation and committal by miners 212. Subsequently at step 506 the method determines if the optimisation rule 210 is invoked. Mechanisms for the invocation of the optimisation rule 210 are discussed above. Where the optimisation rule 210 is not invoked, the method loops. Where the optimisation rule 210 is invoked the method proceeds to step 508. At step 508 the method determines one or more resource providers 204 for providing resources according to the consumer’s 202 resource requirements. Preferably the consumer’s 202 resource requirements are encoded
as required resource types in the optimisation rule 210 by way of the resource constraints defined therein. In some embodiments the consumer’s 202 resource requirements are provided as required resource type parameters for the invocation of the optimisation rule 210. Step 508 determines resource providers 204 based on the consumption rules 208 and a current context of the consumer 202 and the resource providers 204. For example, the current context will include the available tradeable value for the consumer and the resource availability and cost reflected by the current consumption rules. Thus, at step 508 the method determines a consumption configuration for the consumer 202 including an identification of particular resources from particular resource providers 204, an identification of a particular configuration of one or more of the resources and the identification of an extent of consumption of each type of resource. Notably, the configuration of resources can include an identification of a particular resource – e.g. a particular processor or network resource – or alternatively a configuration or characteristics of a resource, such as options, settings, qualities and the like. Further notably the extent of consumption of each type of resource can be limited – such that a particular amount, quantity or time period of resource is reserved or validated for consumption - or continuous such that consumption continues until a change to the resource configuration is effected. In one embodiment the extent of consumption is controlled by a duration between invocations of the optimisation rule 210 such that each resource configuration applies only until replaced by a subsequently evaluated configuration by re-invocation of the optimisation rule 210. The resource configuration thus constitutes the identification of multiple consumption rules 208 and the definition of parameters for the consumption rules 208 defining which and to what extent resources are to be consumed. Subsequently, at step 510, the method effects resource consumption by invoking the identified consumption rules 208 with the parameters. The invocation of consumption rules 208 can be effected by transactions to the blockchain 208 or alternatively by direct messages from the optimisation rule 210 to each consumption rule. The execution of each consumption rule 208 involves the validation of parameters for the consumption of resource and effecting the consumption by expenditure of the consumer’s 202 tradeable value 203 and, optionally, corresponding changes to tradeable value associated with service providers 204.

Thus, in this way embodiments of the present invention provide a decentralised resource consumption management mechanism where consumption of resource by a resource consumer 202 is continually optimised. The consumption rules 208 and optimisation rule 210 are stored to the blockchain 206 such that they are not susceptible to repudiation and the state of the system of Figure 3 can be reconstructed and validated by the network of miners 212. Resource provisioning is dynamically altered as state changes and resource provision is
optimised based on expenditure of a tradeable value to ensure efficiency of resource provision within the constraints of the consumer’s requirements.

One blockchain-based environment suitable for the implementation of embodiments of the present invention is the Ethereum environment. The paper “Ethereum: A Secure Decentralised Generalised Transaction Ledger” (Wood, Ethereum, 2014) (hereinafter Ethereum) provides a formal definition of a generalised transaction based state machine using a blockchain as a decentralised value-transfer system.

In an Ethereum embodiment, a consumption rule 208 is implemented as an Ethereum account having associated logic for validating and effecting resource consumption so constituting Ethereum contracts (or smart contracts). Similarly, an optimisation rule 210 is implemented as an Ethereum account having logic for determining an optimised resource configuration as an Ethereum contract. Each Ethereum account has an address by which it can be referenced for messaging and transactions. The consumer 202 creates the consumption rules 208 by individual Ethereum “create contract” transactions, each returning logic required for implementation of a consumption rule 208. Such transactions, when processed by miners 212, will result in the creation of consumption rule 208 contracts in the Ethereum environment reflected in the blockchain 206. Similarly, the consumer 202 creates the optimisation rule 210 by an Ethereum “create contract” transaction returning the logic required for implementation of the optimisation rule 210. The consumption rules are referenced by the optimisation rule 210 contract by address and, in use, the optimisation rule 210 contract is invoked by transactions submitted to the Ethereum blockchain 206 (or, alternatively, invoked automatically by a centralised or designated system accessing the blockchain 206). To identify resource consumption options the optimisation rule 210 contract can read consumption rule 208 contracts directly from the blockchain 206 or alternatively messages can be passed to the consumption rule 208 contracts which operate in an optimisation mode in which no consumption is effected. Notably, in Ethereum, the optimisation rule 210 contract can communicate with the consumption rules 208 via message passing within the Ethereum virtual machine or via transactions to the blockchain 206. Where the optimisation rule 210 is executed by an Ethereum virtual machine operating in a centralised or designated computer system (so avoiding, centralising or controlling the consumption of gas value for its execution) then message passing to consumption rule 208 contracts will execute in the same Ethereum virtual machine so controlling the consumption of gas by their execution also.

Insofar as embodiments of the invention described are implementable, at least in part, using a software-controlled programmable processing device, such as a microprocessor,
digital signal processor or other processing device, data processing apparatus or system, it will be appreciated that a computer program for configuring a programmable device, apparatus or system to implement the foregoing described methods is envisaged as an aspect of the present invention. The computer program may be embodied as source code or undergo compilation for implementation on a processing device, apparatus or system or may be embodied as object code, for example.

Suitably, the computer program is stored on a carrier medium in machine or device readable form, for example in solid-state memory, magnetic memory such as disk or tape, optically or magneto-optically readable memory such as compact disk or digital versatile disk etc., and the processing device utilises the program or a part thereof to configure it for operation. The computer program may be supplied from a remote source embodied in a communications medium such as an electronic signal, radio frequency carrier wave or optical carrier wave. Such carrier media are also envisaged as aspects of the present invention.

It will be understood by those skilled in the art that, although the present invention has been described in relation to the above described example embodiments, the invention is not limited thereto and that there are many possible variations and modifications which fall within the scope of the invention.

The scope of the present invention includes any novel features or combination of features disclosed herein. The applicant hereby gives notice that new claims may be formulated to such features or combination of features during prosecution of this application or of any such further applications derived therefrom. In particular, with reference to the appended claims, features from dependent claims may be combined with those of the independent claims and features from respective independent claims may be combined in any appropriate manner and not merely in the specific combinations enumerated in the claims.
1. A computer implemented method to provide allocation of one or more computing resources for use by a consumer computing component, each resource having a resource type and being provided by one or more resource providers, and the consumer having associated a quantity of tradeable value constraining an extent of resource consumption, the method comprising:
   defining one or more consumption rules for each of the resource providers, each consumption rule being associated with the consumer and defining: a type of resource; and
one or more tradeable values corresponding to costs for consumption of resource, each tradeable value being associated with a particular configuration of the resource, wherein the consumption rules are recorded in a blockchain data structure;
   defining an optimisation rule associated with the consumer and including: a reference to each of the consumption rules associated with the consumer; a definition of one or more constraints on characteristics of resources; and a consumption optimisation function, wherein the optimisation rule is recorded in the blockchain,
   wherein, in use, a transaction is submitted to the blockchain to trigger consumption of one or more required types of resource and responsive to the transaction the optimisation rule determines one or more resource providers for providing resources of the required resource types and, for each determined provider, the optimisation function determines, based on the consumption rules, a particular configuration and an extent of consumption of each type of resource so as to control expenditure of the consumer’s tradeable value.

2. The method of claim 1 wherein the optimisation rule further includes one or more constraints on expenditure of the consumer’s tradeable value.

3. The method of any preceding claim wherein the optimisation rule effects consumption of resource by causing invocation of one or more consumption rules for the consumer corresponding to the one or more determined resource providers.

4. The method of claim 3 wherein the invocation of each of the one or more consumption rules effects a change to the quantity of tradeable value of the consumer.

5. The method of any preceding claim wherein each of the one or more consumption rules are executable by miner software components operating with the blockchain data structure.
6. The method of any preceding claim wherein the one or more consumption rules are contracts defined in a smart contract blockchain system.

7. The method of any preceding claim wherein the optimisation rule is a contract defined in a smart contract blockchain system.

8. The method of any preceding claim wherein the optimisation rule responsive to the transaction is executed repeatedly so as to continuously adapt resource consumption.

9. The method of any preceding claim wherein the optimisation rule is executable by miner software components operating with the blockchain data structure.

10. The method of any of claims 1 to 8 wherein the optimisation rule is executed by a central computer system.

11. The method of any preceding claim wherein the determinations of the optimisation rule responsive to the transaction are repeated based on one or more of: changes to the consumer's quantity of tradeable value; and changes to a tradeable value, in one or more consumption rules, corresponding to a cost for consumption of resource.

12. The method of any preceding claim wherein a consumption rule evaluates a tradeable value for consumption of a resource with reference to an interface to a service provider for the resource, the interface providing a dynamic tradeable value rate for the resource.

13. A computer system to provide allocation of one or more computing resources for use by a consumer computing component, each resource having a resource type and being provided by one or more resource providers, and the consumer having associated a quantity of tradeable value constraining an extent of resource consumption, the system comprising a processor and a data store wherein the processor is adapted to undertake the steps of:

   defining one or more consumption rules for each of the resource providers, each consumption rule being associated with the consumer and defining: a type of resource; and one or more tradeable values corresponding to costs for consumption of resource, each tradeable value being associated with a particular configuration of the resource, wherein the consumption rules are recorded in a blockchain data structure;
defining an optimisation rule associated with the consumer and including: a reference to each of the consumption rules associated with the consumer; a definition of one or more constraints on characteristics of resources; and a consumption optimisation function, wherein the optimisation rule is recorded in the blockchain,

wherein, in use, a transaction is submitted to the blockchain to trigger consumption of one or more required types of resource and responsive to the transaction the optimisation rule determines one or more resource providers for providing resources of the required resource types and, for each determined provider, the optimisation function determines, based on the consumption rules, a particular configuration and an extent of consumption of each type of resource so as to control expenditure of the consumer’s tradeable value.

14. A computer program element comprising computer program code to, when loaded into a computer system and executed thereon, cause the computer to perform the steps of a method as claimed in any of claims 1 to 12.
INTERNATIONAL SEARCH REPORT

A. CLASSIFICATION OF SUBJECT MATTER

INV. G06Q20/36  G06F21/62
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)
G06Q  G06F

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)

EP0-Internal

C. DOCUMENTS CONSIDERED TO BE RELEVANT

<table>
<thead>
<tr>
<th>Category*</th>
<th>Citation of document, with indication, where appropriate, of the relevant passages</th>
<th>Relevant to claim No.</th>
</tr>
</thead>
</table>

[X] Further documents are listed in the continuation of Box C.  

[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

[X] Document member of the same patent family

Date of the actual completion of the international search  

12 September 2016

Date of mailing of the international search report  

22/09/2016

Name and mailing address of the ISA/  

European Patent Office, P.B. 5018 Patentlaan 2  
NL - 2280 HV Rijswijk  
Tel. (+31-70) 340-3040,  
Fax: (+31-70) 340-3016

Authorized officer  

Frank, Mario
<table>
<thead>
<tr>
<th>Category</th>
<th>Citation of document, with indication, where appropriate, of the relevant passages</th>
<th>Relevant to claim No.</th>
</tr>
</thead>
<tbody>
<tr>
<td>X</td>
<td>ENAS AL KAWASMI ET AL: &quot;Bitcoin-Based Decentralized Carbon Emissions Trading Infrastructure Model&quot;, SYSTEMS ENGINEERING, vol. 18, no. 2, 28 September 2014 (2014-09-28), pages 115-130, XP055242445, US ISSN: 1098-1241, DOI: 10.1002/sys.21291 Section 2.1 Section 3.1 Section 4</td>
<td>1-14</td>
</tr>
</tbody>
</table>