US 20230047165A1

(19) **United States**
(12) **Patent Application Publication**
Melton

(10) **Pub. No.: US 2023/0047165 A1**
(43) **Pub. Date: Feb. 16, 2023**

(54) **DETERMINISTIC CREDIT SYSTEM**

(71) Applicant: **REFINITIV US ORGANIZATION LLC**, New York, NY (US)

(72) Inventor: **Hayden Melton**, Philadelphia, PA (US)

(73) Assignee: **REFINITIV US ORGANIZATION LLC**, New York, NY (US)

(21) Appl. No.: **17/886,587**

(22) Filed: **Aug. 12, 2022**

Related U.S. Application Data

(60) Provisional application No. 63/237,651, filed on Aug. 27, 2021, provisional application No. 63/233,038, filed on Aug. 13, 2021.

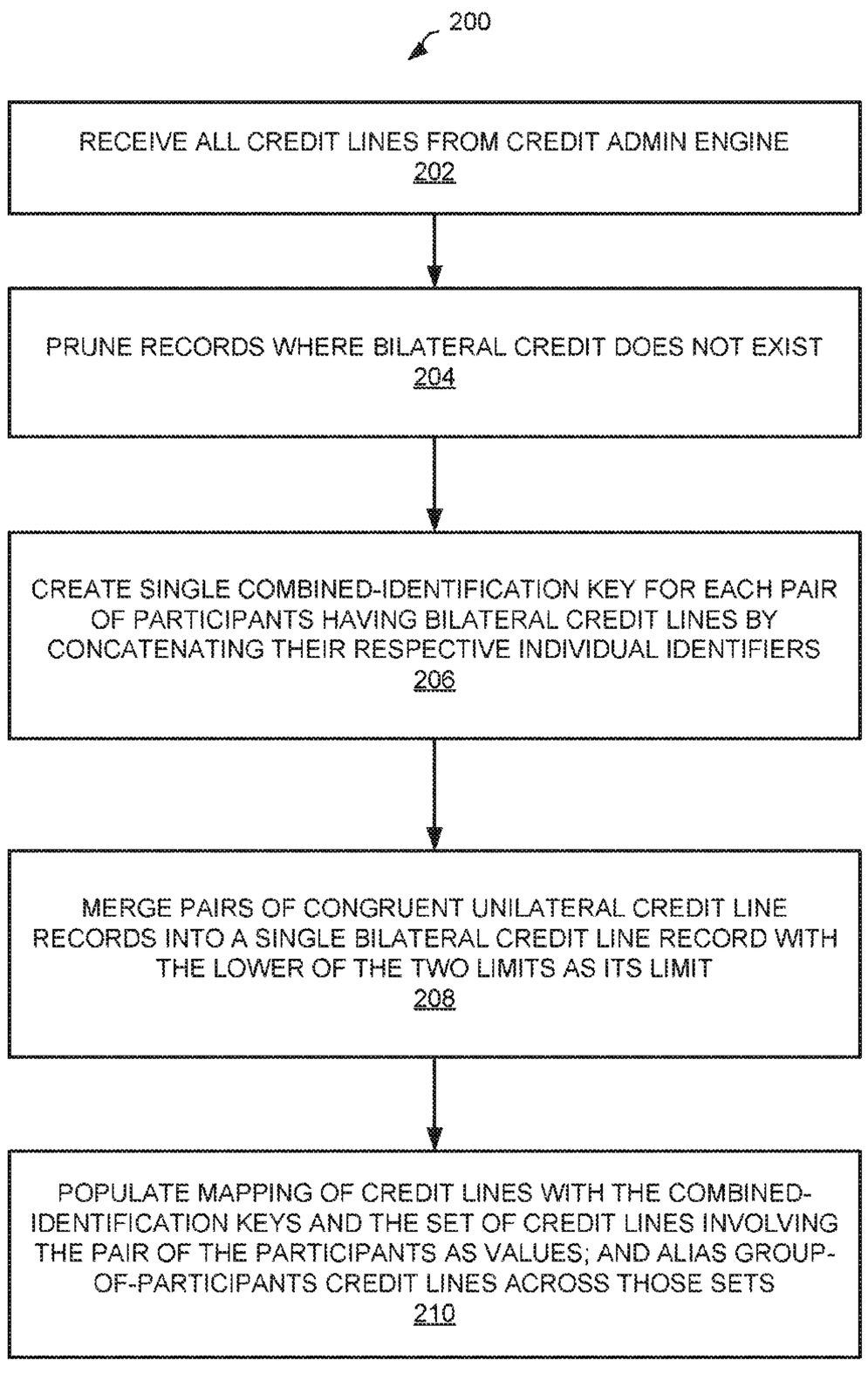
Publication Classification

(51) **Int. Cl.**
G06Q 40/02 (2006.01)
G06Q 40/04 (2006.01)

(52) **U.S. Cl.**
CPC ***G06Q 40/025*** (2013.01); ***G06Q 40/04*** (2013.01)

(57) **ABSTRACT**

Systems and methods of a deterministic credit system are disclosed to provide high-performance credit-checking while reducing the need for horizontal scaling or implementing larger system architecture. The systems and methods presented herein may achieve this in several ways, for example, by: storing credit lines in memory; optimizing the storage of those lines in memory and performing precomputations on them so as to minimize operations performed on them; and minimizing disk and network I/O by (i) sequencing and persisting inputs to the credit checking computations and (ii) ensuring those computations are deterministic, consequently eliminating the need to persist or transmit the outputs of those computations.



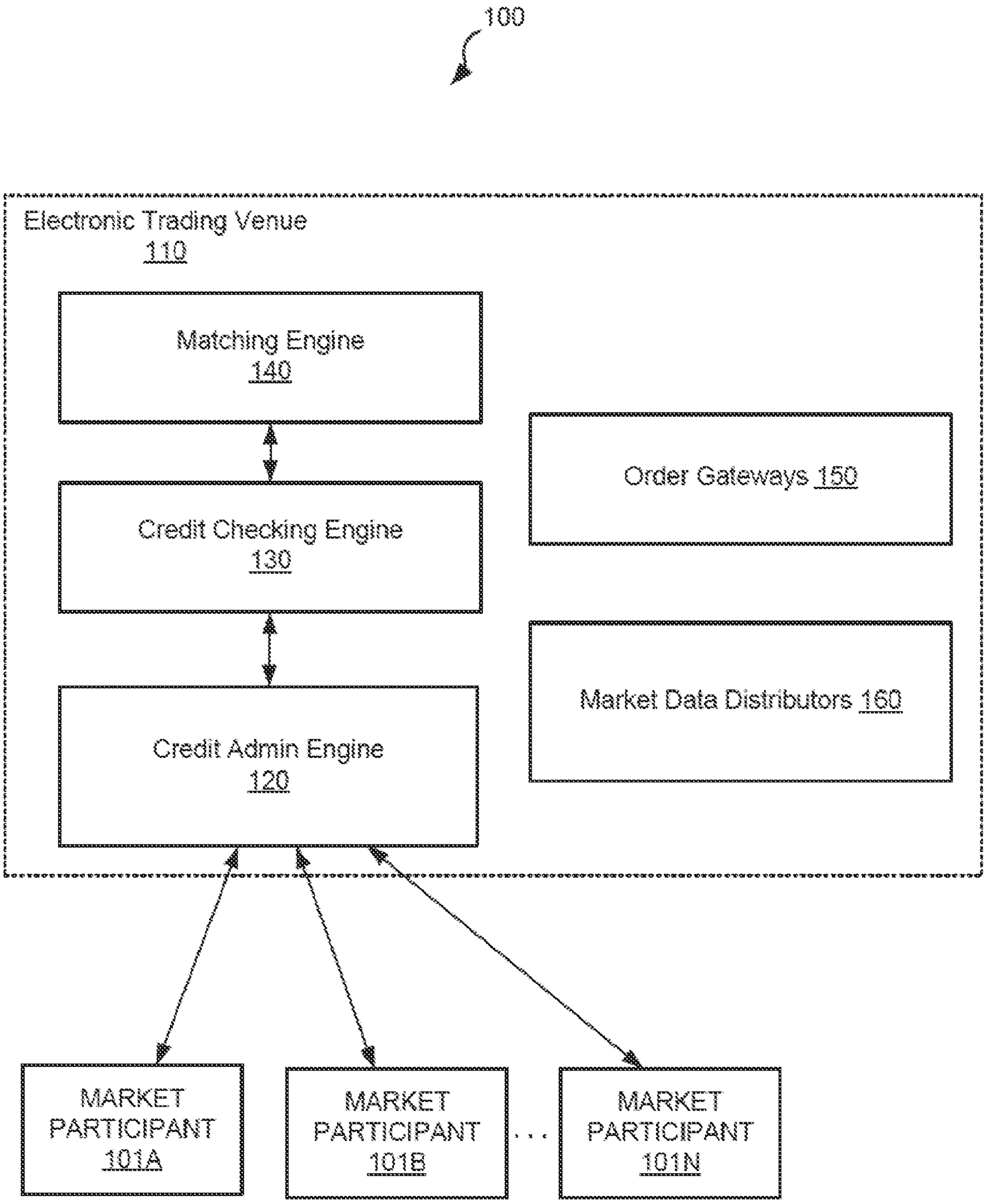

100
Electronic Trading Venue
110
Matching Engine
140
Credit Checking Engine
130
Credit Admin Engine
120
Order Gateways 150
Market Data Distributors 160
MARKET PARTICIPANT 101A
MARKET PARTICIPANT 101B
. . .
MARKET PARTICIPANT 101N

FIG. 1

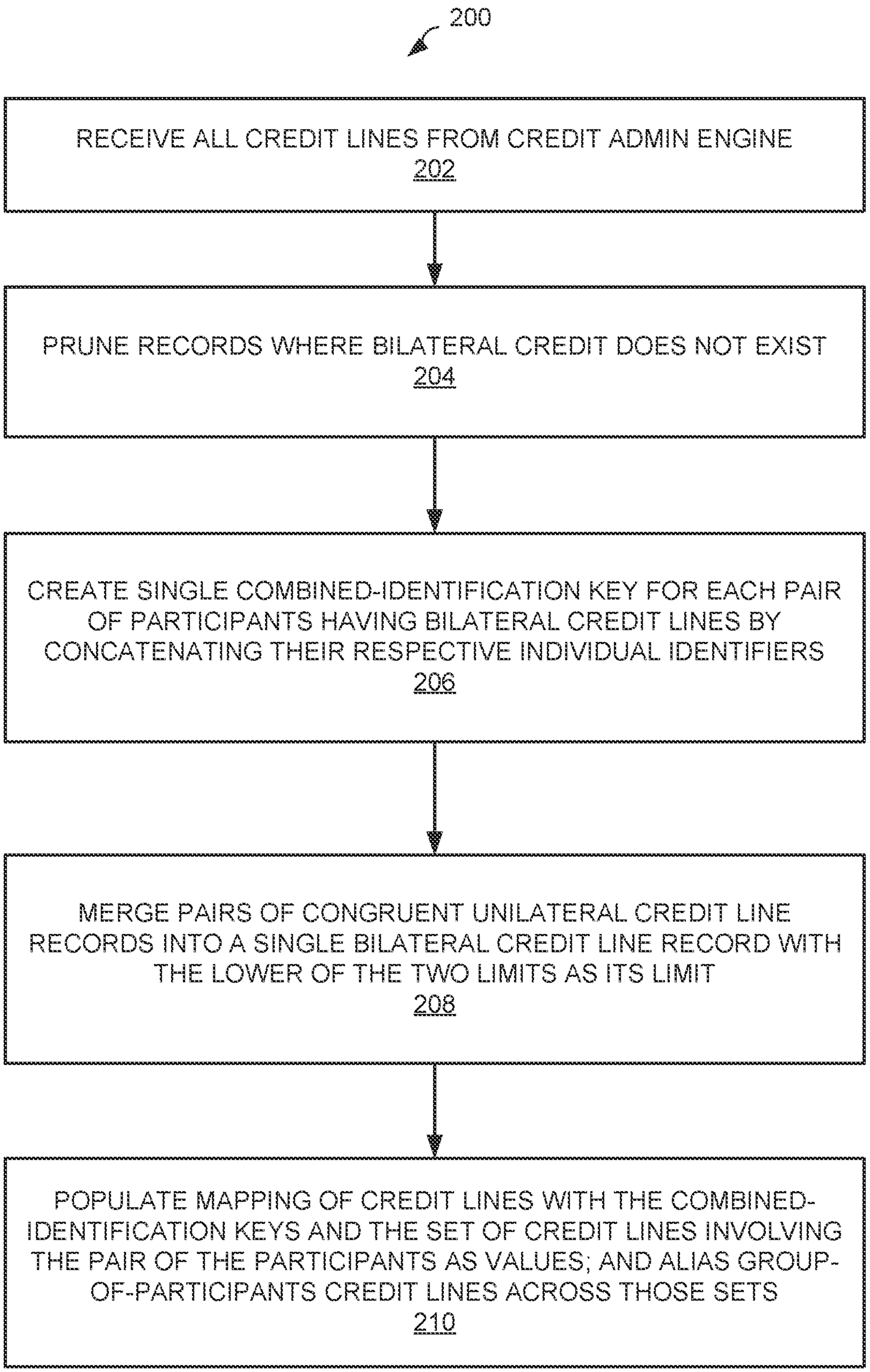

200
RECEIVE ALL CREDIT LINES FROM CREDIT ADMIN ENGINE
202
PRUNE RECORDS WHERE BILATERAL CREDIT DOES NOT EXIST
204
CREATE SINGLE COMBINED-IDENTIFICATION KEY FOR EACH PAIR OF PARTICIPANTS HAVING BILATERAL CREDIT LINES BY CONCATENATING THEIR RESPECTIVE INDIVIDUAL IDENTIFIERS
206
MERGE PAIRS OF CONGRUENT UNILATERAL CREDIT LINE RECORDS INTO A SINGLE BILATERAL CREDIT LINE RECORD WITH THE LOWER OF THE TWO LIMITS AS ITS LIMIT
208
POPULATE MAPPING OF CREDIT LINES WITH THE COMBINED-IDENTIFICATION KEYS AND THE SET OF CREDIT LINES INVOLVING THE PAIR OF THE PARTICIPANTS AS VALUES; AND ALIAS GROUP-OF-PARTICIPANTS CREDIT LINES ACROSS THOSE SETS
210

FIG. 2

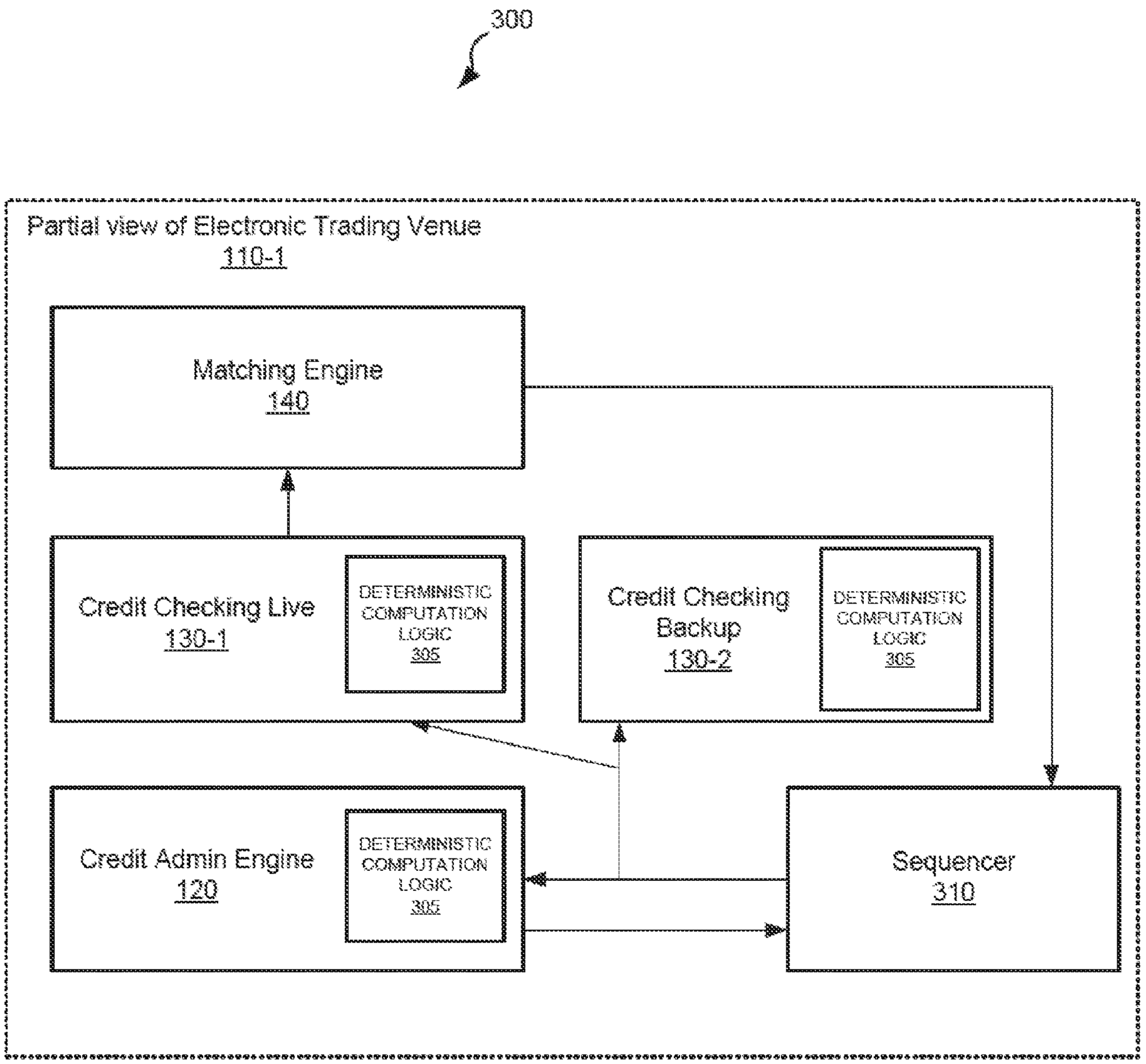

300
Partial view of Electronic Trading Venue
110-1
Matching Engine
140
Credit Checking Live
130-1
DETERMINISTIC COMPUTATION LOGIC
305
Credit Checking Backup
130-2
DETERMINISTIC COMPUTATION LOGIC
305
Credit Admin Engine
120
DETERMINISTIC COMPUTATION LOGIC
305
Sequencer
310

FIG. 3

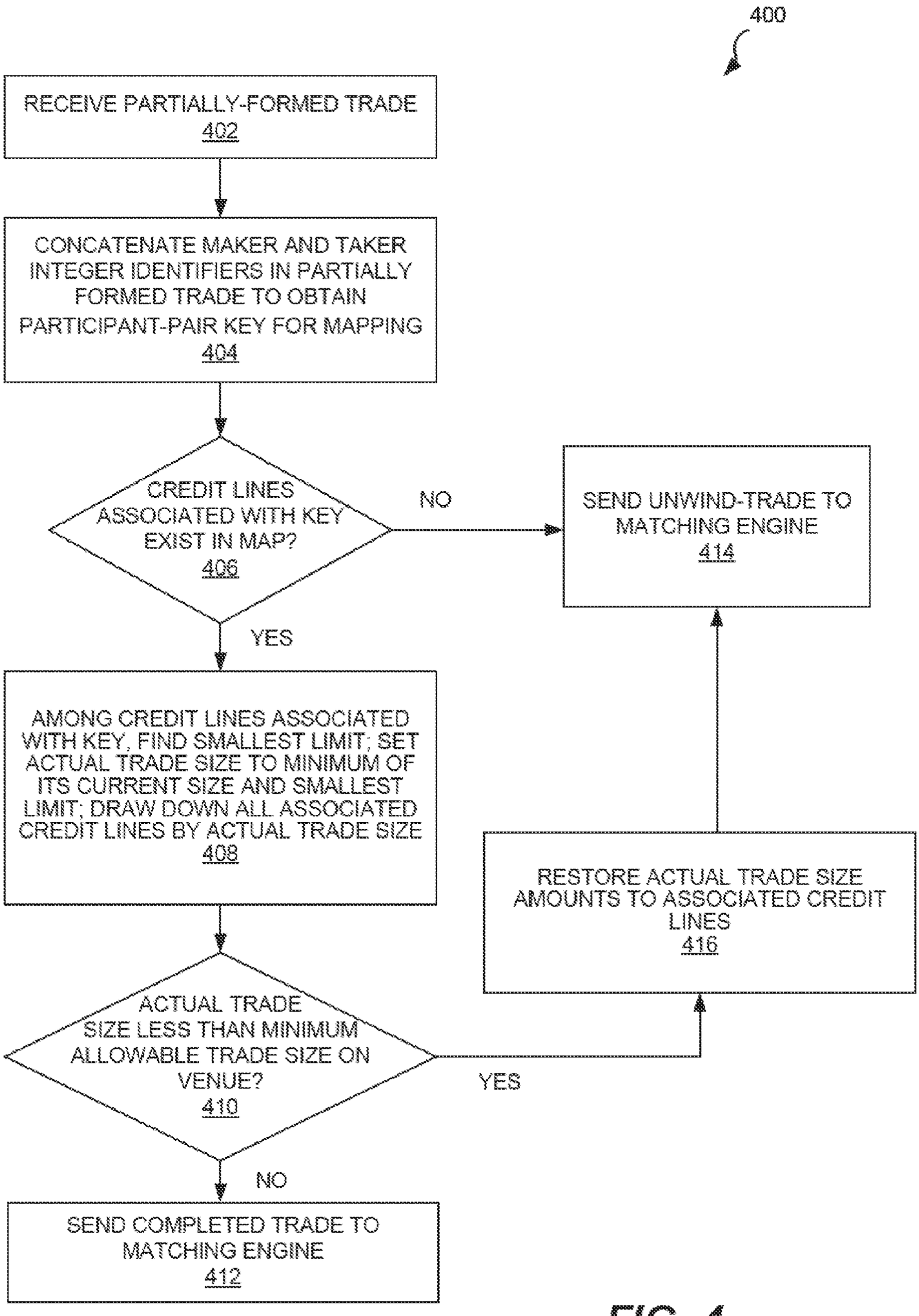

400
RECEIVE PARTIALLY-FORMED TRADE
402
CONCATENATE MAKER AND TAKER INTEGER IDENTIFIERS IN PARTIALLY FORMED TRADE TO OBTAIN PARTICIPANT-PAIR KEY FOR MAPPING
404
CREDIT LINES ASSOCIATED WITH KEY EXIST IN MAP?
406
NO
SEND UNWIND-TRADE TO MATCHING ENGINE
414
YES
AMONG CREDIT LINES ASSOCIATED WITH KEY, FIND SMALLEST LIMIT; SET ACTUAL TRADE SIZE TO MINIMUM OF ITS CURRENT SIZE AND SMALLEST LIMIT; DRAW DOWN ALL ASSOCIATED CREDIT LINES BY ACTUAL TRADE SIZE
408
RESTORE ACTUAL TRADE SIZE AMOUNTS TO ASSOCIATED CREDIT LINES
416
ACTUAL TRADE SIZE LESS THAN MINIMUM ALLOWABLE TRADE SIZE ON VENUE?
410
YES
NO
SEND COMPLETED TRADE TO MATCHING ENGINE
412

FIG. 4

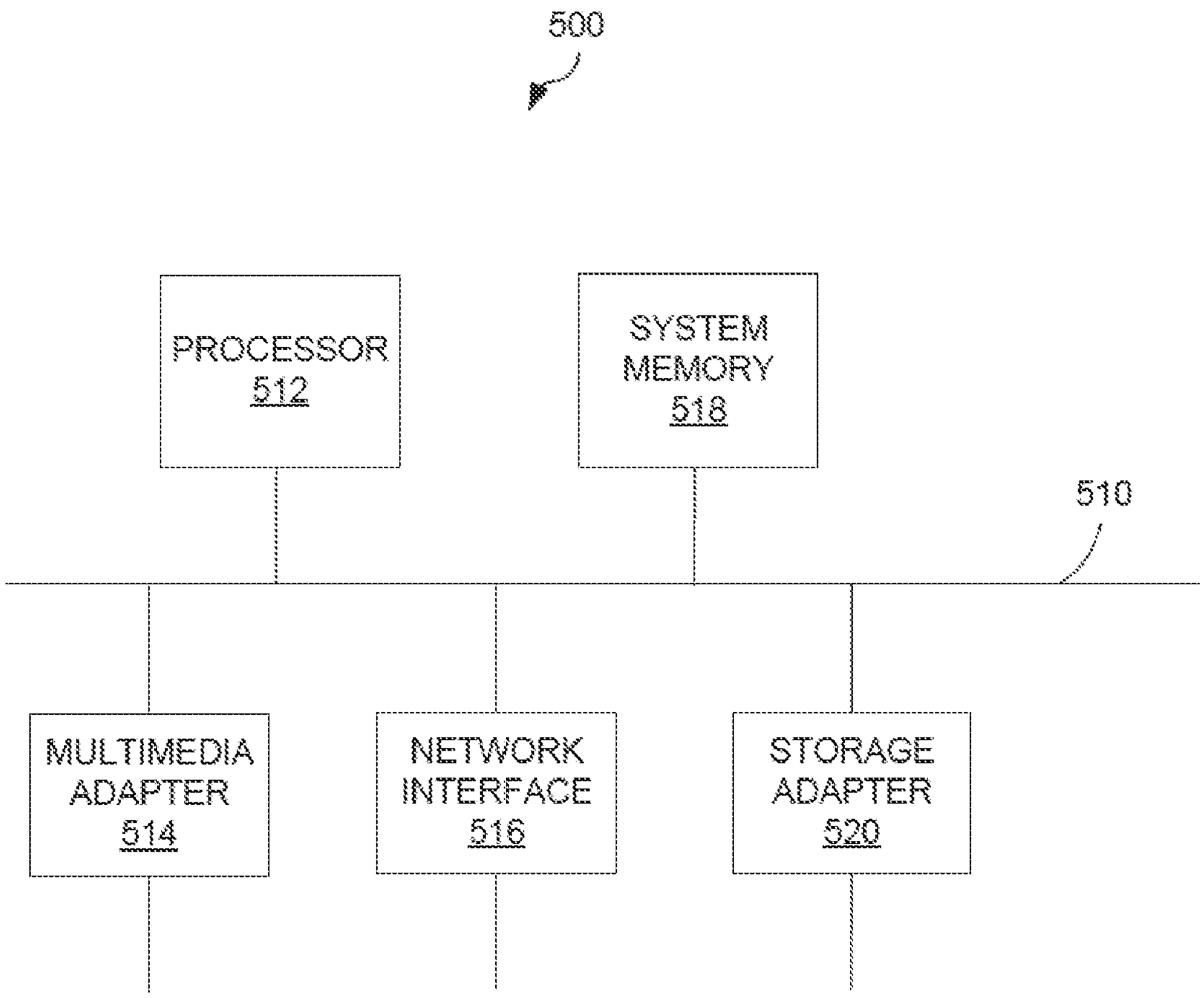

500
PROCESSOR
512
SYSTEM
MEMORY
518
510
MULTIMEDIA
ADAPTER
514
NETWORK
INTERFACE
516
STORAGE
ADAPTER
520

FIG. 5

DETERMINISTIC CREDIT SYSTEM

CROSS-REFERENCE TO RELATED APPLICATIONS

[0001] This application claims priority to U.S. Provisional Pat. Application No. 63/233,038, filed Aug. 13, 2021, entitled "Pipelined Credit Checking," and U.S. Provisional Pat. Application No. 63/237,651, filed Aug. 27, 2021, entitled "Deterministic Credit System," the contents of each of which are incorporated by reference in their entireties herein.

BACKGROUND

[0002] In certain markets, such as the spot foreign exchange (spot FX) market, trading may require the existence of a bilateral credit relationship between the two counterparties to a trade, as opposed to the existence of a central counterparty as is the norm in equities and futures markets. To that end, electronic spot FX trading venues tend to provide functionality to participants on that venue to specify credit lines they are extending to other such participants. The venue may then determine from those bilateral credit lines if a trade between two participants can occur or not and if it can drawdown those credit lines by the size of the trade, for example, by examining the lines existence, magnitude and type.

[0003] In the pursuit of performance, such as achieving low-latency and high-throughput in order processing, horizontal scaling is often employed extensively in the architecture of many electronic trading venues. Horizonal scaling may refer to the addition of computational resources such as computers and/or other devices to increase the capacity of a computer system or network. To that end, such electronic trading venues tend to be composed of a plurality of order gateway components, a plurality of matching engine components, and a plurality of market data distribution components. These components may be implemented as distinct programs, modules or engines within the electronic trading venue, each designed to perform particular tasks or functions.

[0004] For such electronic trading venues also implementing risk checking componentsthat in the context of spot FX such components may be used or otherwise repurposed to perform credit checking i.e., storing, interrogating, and drawing down bilateral credit lines-conventionally those too may be horizontally scaled.

[0005] A number of problems become apparent in attempting to horizontally scale the credit system on an electronic spot FX trading venue. First, certain types of credit limits such as net credit do not lend themselves well to being partitioned among distinct components in the required manner, such that the components can operate substantially independently of one another. Second, if the credit universe is represented as a directed graph-where the nodes of which are participants, and edges of which are credit relationships of varying type and magnitude, noting that other specific mappings to edges and to nodes may exist-then the graph tends to be connected which inhibits partitioning it across distinct components. Third, in the presence of prime brokerage relationships and in the presence of groups of limits being imposed on pluralities of currencies, instruments, or counterparties, and in conjunction with ungrouped limits, partitioning credit limits among substantially independent components is further inhibited.

[0006] These and other problems motivate the need for a high-performance credit system on electronic trading venues where trades involve bilateral credit between participants.

SUMMARY

[0007] The disclosure relates to systems, methods, and computer-readable media for a deterministic credit system to provide high-performance credit-checking while reducing the need for horizontal scaling or implementing larger system architecture. The systems and methods presented herein may achieve this in several ways, for example, by: storing credit lines in memory; optimizing the storage of those lines in memory and performing precomputations on them so as to minimize operations performed on them; and minimizing disk and network I/O by (i) persisting inputs to the credit checking computations and (ii) ensuring those computations are deterministic, consequently eliminating the need to persist or transmit the outputs of those computations.

BRIEF DESCRIPTION OF THE DRAWINGS

[0008] Features of the present disclosure may be illustrated by way of example and not limited in the following Figure(s), in which like numerals indicate like elements, in which:

[0009] FIG. **1** illustrates an example of a computer system of an electronic trading venue for performing credit administration and checking workflow among key venue components;

[0010] FIG. **2** shows a flowchart for the creation of in-memory credit lines in exemplary deterministic credit checking;

[0011] FIG. **3** illustrates how one example of the system of the present disclosure can efficiently transmit data between components to provide auditability, fault tolerance, recovery and so on, for transmission of credit-related data between the various engines of the trading venue, during a trading session;

[0012] FIG. **4** illustrates a flowchart of selected logic in the credit checking engine; and

[0013] FIG. **5** illustrates an example of a computer system illustrated in FIG. **1**.

DETAILED DESCRIPTION

[0014] Systems and methods of a deterministic credit system are disclosed to provide high-performance credit-checking while reducing the need for horizontal scaling or implementing larger system architecture. The systems and methods presented herein may achieve this in several ways, for example, by: storing credit lines in memory; optimizing the storage of those lines in memory and performing precomputations on them so as to minimize operations performed on them; and minimizing disk and network I/O by (i) persisting inputs to the credit checking computations and (ii) ensuring those computations are deterministic, consequently eliminating the need to persist or transmit the outputs of those computations.

[0015] On a conventional electronic trading venue, market participants specify credit lines, including attributes thereof

such as their size and type, that they are extending to other such participants via a credit administration engine. In conventional systems, participants may access the credit administration engine via a graphical user interface (GUI) and/or application program interface (API). Upon joining such a venue, a participant may be required by the venue operator to extend credit to a minimum number of other participants on the venue, and those credit relationships once established may persist until they are modified by either the venue operator, for example by removal of a line because a counterparty to which the participant has extended credit has left the venue, or by the participant, for example because the creditworthiness of a specific counterparty has improved and correspondingly the line extended to that participant can be increased.

[0016] Besides allowing specification of the magnitude of a credit line being extended, conventional electronic trading venues may also allow the specification of various types of lines and groups thereof. A common type of line is a gross limit. In this type of line, the magnitudes of the trades are summed without regard to their direction (buy or sell). For instance, a buy of $1.5 million and sell of $2.5 million would, under a gross limit scheme, draw down $4 million (=1.5 million +2.5 million) of credit from the line. Another common type of line is a net limit. In this type of line, the directions of trades are material in computing their drawdown. Nominally, buys are positive values and sells are negative values, so in the previous example the drawdown would be -$1 million (=1.5 million-2.5 million). To allow summation in spot FX, magnitudes of the trades may be expressed in a currency of the participant's choosing using a conversion rate; lines a participant extends may also be expressed in that same currency.

[0017] Conventional electronic trading venues may also allow participants to specify arbitrary groupings of credit lines they have extended that may or may not co-exist or even "overlap" with those ungrouped lines they have also extended. A non-limiting example that will ultimately serve to illustrate complications arising from this is provided as follows: a participant extends a single gross line to a group of counterparties all domiciled in a certain country, and further extends individual gross lines to each such counterparty in the country. The combination of these lines may allow the participant to express that they are unwilling to extend more than, for example, $100 million credit in the aggregate to all counterparties in that country. However, the sum of the individual lines extended across all such participants may actually exceed that $100 million. For example, say there are five distinct participants each extended between $30 million and $50 million each. Since, conventionally, among a plurality of credit lines extended to a counterparty it is the lowest line that is controlling, a setup like the one described can advantageously allow a participant to express their credit risk in terms of both a country and of individual counterparties domiciled there advantageously without, for example, continuously having to "rebalance" limits on the venue as trades draw them down.

[0018] Conventionally on FX venues, arbitrary grouping of credit lines on attributes in addition to counterparties may also exist, such as: currency, instrument, tenor, type and so on. These too may also co-exist and/or overlap, and given that it is the lowest line among all those extended to a counterparty that is the controlling one, other groupings may be used to similar effect. For example, Continuous Linked Settlement (CLS)-eligible currencies may be grouped together, or prime brokerage clients belonging to a single prime broker may be grouped together for the extension of shared credit among them to venue counterparties, while simultaneously imposing individual net limits on each such prime brokerage client.

[0019] Having described the specification of credit lines on a conventional electronic trading venue, attention is now given to how such a venue makes use of those lines to form trades between participants. Typically, such a venue will also include a credit checking component. In an example of the present disclosure involving a deterministic credit system, only one such component is necessary, while conventional venues may include a plurality of such horizontally-scaled components and certain parts of their functionality may even exist as subcomponents within matching engine components. Regardless of how the credit check component(s) manifest however, conventionally they receive credit line information from the venue's credit admin component. Also conventionally, trades between participants are partially-formed on the venue in the matching engine component, for example, based on price-compatibility, and these partially formed trades are sent to the credit check component to either be (i) completed as a result of a successful credit check, or (ii) unwound as a result of a failed credit check. Pertinent aspects of this workflow are illustrated in FIG. **1**.

[0020] FIG. **1** illustrates an example of a computer system **100** of an electronic trading venue **110** for performing credit administration and checking workflow among key venue components, according to examples of the present disclosure. In the examples described herein, the various components are referred to as engines that may represent separate programs for performing distinct tasks within the computer system **100**. The example electronic trading venue **110** may include a credit admin engine **120**, a credit checking engine **130**, and a matching engine **140**. The electronic trading venue **110** may also include one or more order gateways **150** and one or more market data distributors **160**. The order gateways **150** may be programmed to carry out instructions provided by a market participant during a trading session, such as buy or sell orders. The market data distributors **160** may be programmed to provide market data during a trading session.

[0021] The credit admin engine **120** may provide an interface to a plurality of market participants, such as market participants **101**A, **101**B... **101**N. The credit admin engine **120** may provide the interface to create credit lines for each market participant **101**A... **101**N initially, in order to allow participation in the electronic trading venue **110**. Later, the credit admin engine **120** may provide the interface to modify any of the credit lines for the market participants **101**A... **101**N.

[0022] To establish the credit lines, the credit admin engine **120** may send the credit line information to the credit checking engine **130**. The credit checking engine **130** may perform crediting checking using partially-formed trades against the credit lines. The partially-formed trades may be obtained by the credit checking engine **130** from the matching engine **140**. The credit checking engine **130** may also be programmed to transmit completed trades and any unwound trades to the matching engine **140**.

[0023] In an example of a deterministic credit system of the present disclosure, at the start of, or immediately prior

to, the electronic trading venue's **110** trading session, all credit line information may be transferred from the credit admin engine **120** to the credit checking engine **130**. Each such line may be transferred over a network connection, such as TCP/IP, as a single record containing: a unique identifier for the line, the counterparty extending the credit, the counterparty to which credit is being extended, the type of the limit as an enumeration, and the magnitude of the limit as a numeric value, as some examples. Upon receipt of all those credit line records and before the electronic trading venue **110** begins conducting the trading session, the credit checking engine **130** may then efficiently store them in memory, utilizing the following techniques:

[0024] 1. Prune records where bilateral credit does not exist. A successful trade requires that both counterparties have extended credit to one another. Since conventionally a counterparty only knows those to whom they have extended credit, and not those that have extended them credit, it is possible "one-sided" lines exist. These one-sided lines can be "pruned" in the sense that they may be removed from memory or otherwise ignored by the credit checking engine **130** because they can never result in trades. Such pruning can reduce the size of the data structure in which the records are stored and thus may advantageously improve performance by reducing lookup times thereon.

[0025] 2. Create a single key for each distinct pair of counterparties. To further improve lookup times on the data structure in which credit lines are stored, each distinct pair of counterparties can be assigned a unique key. For example, in the case that there are less than 2^{32} distinct counterparties on the venue and where the computer architecture uses 64-bit words, each counterparty can be assigned a unique 32-bit integer identifier. The assignment of such an integer may be done by the venue operator and its lookup and insertion into a message received from a market participant may be performed by an order gateway **150**. A single key for each distinct pair of counterparties can then be generated by concatenating the bitstrings of the two counterparty's integers, nominally using the smaller integer as the left and larger as the right in that concatenation, as one example. In one example implementation, as integer operations, this concatenation may be expressed as a cast of the smaller 32-bit integer to a 64-bit integer, a shift left on it by 32 bits, and finally by adding to it or bitwise boolean OR'ing it with the second integer.

[0026] 3. Merge congruent records where bilateral credit lines exist. Certain types of credit lines among a pair of counterparties are "congruent" when they can be merged into a single record with a single limit, which is the lesser of the two limits among the two being merged. For example, if counterparty A extends B net credit of $10 million in a single credit line, and counterparty B extends A net credit of $8 million in another single credit line, then the smaller of the two limits is controlling and this can be precomputed and stored in a single record upon receipt of the credit lines. In this way, such precomputation can reduce the time taken to perform the credit check and drawdown because instead of comparing the partially formed trade to both limits, only one such comparison operation (and drawdown operation) is necessary. The precomputation may be performed at the start of the trading session after all credit lines have been received.

[0027] 4. Aliasing of credit limits as values in the data structure. Aliasing may refer to a computer programming term where two or more references to the same object exist in a computer program's object graph. Noting that some credit lines involve groups of counterparties, rather than creating a new object for each such counterparty in the group such that multiple duplicate objects would be brought into existence, a single object representing the credit line can be created and shared by all affected counterparties. For example, if the data structure in which the credit checking engine **130** storing credit lines is a hashmap, the keys of which are pairs of counterparties stored as 64-bit integers and each value of which is a set of credit lines, then aliasing can ensure the same credit line object appears in multiple such sets. In this way, this aliasing may improve performance by reducing the number of lookup operations on the hashmap and correspondingly the number of comparison and drawdown operations when a credit check involves a group limit.

[0028] FIG. **2** shows a flowchart **200** for the creation of in-memory credit lines in exemplary deterministic credit checking, illustrating use of the above techniques. Consistent with the disclosure herein, at **202**, the credit checking engine **130** may receive credit lines from the credit admin engine **120**. At **204**, the credit checking engine **130** may pre-process the records, such as by pruning records where bilateral credit does not exist. In some examples, the credit checking engine **130** may prune the records before some or all subsequent processing. At **206**, the electronic trading venue **110** may create a single combined-identification key for each pair of participants having bilateral credit lines by concatenating their respective individual identifiers. An example of the identification key may be the 64-bit key for each pair of participants using their respective 32-bit integer identifiers as described above. This may be performed by one or more of the order gateways **150** that respond to instructions from the participants for choosing their individual identifiers. In other cases, the individual identifiers may be assigned by the credit admin engine **120** and the concatenation may be performed by the credit checking engine **130** or the matching engine **140** as some examples.

[0029] At **208**, the credit checking engine **130** may merge pairs of congruent unilateral credit line records into a single bilateral credit line record with the lower of the two limits being set as its limit. This is another further optimization to reduce latency and processing power during a trading session. At **210**, the electronic trading venue **110** may populate mapping of credit lines with the combined-identification keys and the set of credit lines involving the pair of the participants as values. The credit lines of the groups of participants may be aliased across these sets. An example of the type of mapping may be the hashmap of credit lines. This may be performed by the credit checking engine **130** as an example.

[0030] For reasons described earlier, it is clear that, even for a single pair of participants, there may exist a plurality of (bilateral) credit lines and that the process of completing or unwinding a partially-formed trade may involve checking a large number of them. It is observed then that the volume of data output from such a computation may far exceed the volume of its (incremental) input. Disadvantageously then, transmitting the output of such a computation over a computer network or from memory to disk for the purposes of auditability, fault-tolerance, recovery and so on may be slow and thus detrimental to venue performance.

[0031] To address at least these issues, in an example of the deterministic credit system of the present disclosure, inputs to the credit checking engine **130** are sequenced and computations performed thereon are specifically designed to be deterministic. Under these two conditions, the potentially voluminous outputs of computations need not be transmitted over a computer network, or from memory to disk. Instead, for the purposes of auditability, fault-tolerance, recovery and the like, it is sufficient to transmit only the sequenced inputs. Since those inputs will tend to be smaller, performance as judged by latency and throughput, for example, will advantageously be improved on venues implementing deterministic credit checking as described herein.

[0032] While the credit lines themselves are likely very voluminous inputs to the process of credit checking, the initial transmission of those lines to the credit checking engine **130** from the credit admin engine **120** does not affect venue performance. This is because, as noted earlier, the transmission of the credit lines may occur prior to the beginning of trading session. Further, if a trading session has not commenced, then participants cannot submit orders or receive market data, so the trading session may be unaffected by performance characteristics of the venue during this time.

[0033] In certain schemes such as those described in U.S. Provisional Patent 63/233,038 titled "Pipelined Credit Checking," some of the responsibility for matching of price-compatible contra-orders and for executing advanced order types, such as icebergs, is shifted from the matching engine **140** into the credit checking engine **130**. Certain aspects of that functionality may require randomization inside the credit checking engine **130**, which ostensibly may seem at odds with the determinism of computations that is required for their inputs to be transmitted rather than their outputs. A technique for achieving pseudo-randomization in this context is thus disclosed.

[0034] Pseudo-randomization for deterministic computations. In some examples of the deterministic credit system of the present disclosure, relevant computation only occurs upon receipt of an input message, such as a partially-formed trade. Consequently, randomization is only invoked when an input message is received that includes an action whose computation requires it—and crucially not at any other time. A pseudo-random number generator (PRNG) can thus be utilized that takes as its seed data exclusively derived from the input message, ultimately requiring the use of randomization in its computation. To appear random, the data chosen from the message must be substantially independent of the things in it requiring randomization. If the ordering of the two participants appearing in a partially-formed trade requires randomizing, then the identifiers of those two participants appearing in the input message would not be good candidates for input to a PRNG's seed.

[0035] Example inputs to the PRNG's seed may be a unique, monotonically increasing integer trade identifier generated by and inserted into the message by the matching engine **140**, or a monotonically increasing integer message sequence number assigned to the message and generated by an engine of the electronic trading venue **110** that sequences messages before they are sent to the credit checking engine **130**. If the size of the data desired to be used as a seed from the message exceeds that required by the PRNG, it may be hashed down to an appropriate length (for example, 64 bits) using an off-the-shelf hashing algorithm.

[0036] Having described above certain optimizations of the present disclosure before trading sessions are conducted, FIG. **3** illustrates how one example **300** of the system of the present disclosure can efficiently transmit data between components to provide auditability, fault tolerance, recovery and so on, for transmission of credit-related data between the various engines of the electronic trading venue **110**, during a trading session. In FIG. **3**, a partial view of the electronic trading venue **110-1** is shown. Here, an engine deterministic computation logic is introduced that allows each engine that contains it—in this example, the live credit checking engine **130-1**, the backup credit checking engine **130-2**, and the credit admin engine **120**—to independently perform the credit checks and drawdowns from the inputs to credit checking. Each of these components may also be populated with its own hashmap data structure of credit lines previously described in the manner previously described. The live credit checking engine **130-1** and the backup credit checking engine **130-2** may be two components within the credit checking engine **130**. The sequencer **310** introduced in FIG. **3** ensures all such engines receive all the same inputs in the same order. It may do so using a TCP/IP connection over a computer network to each, and by inserting into each message it sends a sequence number. Gaps in that sequence number could be used by the receivers of those messages to indicate a missed message, for example, as in the sequence number scheme of the FIX protocol.

[0037] Under normal operation, only the live credit checking engine **130-1** need send completed and unwound trades back to the matching engine **140**, but in a failure scenario the backup credit checking engine **130-2** may provide this function and the backup credit checking engine **130-2** may operate in live-mode for near instantaneous failover. The matching engine **140** may be programmed to send (initially) unsequenced and partially-formed trades (or a plurality of price-compatible contra orders per U.S. Provisional Patent 63/233,038) to the sequencer **310**, and then the sequencer **310** sequences and distributes them to each of the credit admin engine **120**, live credit checking engine **130-1**, and backup credit checking engine **130-2**. The credit admin engine **120** may be programmed to send, to the sequencer **310**, unsequenced changes to updated credit lines resulting from a market participant manually editing a credit line, for example, to increase it, remove it, etc., intra-session. The sequencer **310** may then also sequence and distribute these to each of the credit admin engine **120**, live credit checking engine **130-1**, and backup credit checking engine **130-2**. The sequencer **310** may also send other credit changes that may occur intra-session, such as a value date rollover for a particular instrument as updated lines.

[0038] The credit admin engine **120** contains the deterministic computation logic because, among other things, it can provide to participants a contemporaneous view of the credit they have utilized. The credit remaining across a participants lines, of course, is affected by the trades that occur during the trading session and will usually decrease as a result of those trades. The credit admin engine **120** may also provide to participants data on trades that have failed due to missing or exhausted credit lines, and this too is calculated as a matter of course by the deterministic computation logic.

[0039] Although not illustrated, it should be noted that each market data distributor **160** illustrated in FIG. **1** may include all or a portion of the deterministic computation logic **305**. For example, each market data distributor **160** may use all or a portion of the deterministic computation logic **305** to perform credit screening. Such credit screening may be performed in order to ensure that a given market participant receives appropriate market data updates in accordance with their bilateral credit with other market participants.

[0040] A flowchart **400** illustrating selected logic in the credit checking engine **130** in an example of the invention is provided in FIG. **4**. At **402**, a partially formed trade is received. The trade may be only partially formed because an agreement may be made between the maker and taker parties, but still other conditions may need to be met. For example, a credit check may need to be passed that may include ensuring that the partially-formed trade falls within the parameters of the parties' credit lines, and acceptable boundaries within the trading venue may need to be met. At **404**, the key to the hashmap containing the bilateral credit lines is obtained by combining the identifiers for the maker and taker to the partially formed trade into a single key. At **406**, the key is then used to look up the credit lines associated with that pair of participants. If no such key or associated lines are found, then at **414**, the credit check has failed and a trade-unwind message is sent to the matching engine **140**. If, however, at **408** such lines are found, then the credit checking engine **130** may find the smallest limit among the credit lines and then set the actual trade size to the minimum or lesser of its current size and the smallest credit limit. Then all of the lines associated with the key are drawn down by the minimum of two values: the size of the partially-formed trade and the minimum amount remaining among the associated credit lines. At **410**, if then the actual size of the trade is less than the venue's minimum trade size, then at **416** the credit check fails and the trade is denied. The drawdowns are reversed, and then at **414** a trade-unwind is sent to the matching engine **140**. If the actual size meets the minimum trade size, then at **412** the credit check is successful and a completed trade message is sent to the matching engine.

[0041] FIG. **5** illustrates an example of a computer system **500** that may be implemented by the computer system illustrated in FIGS. **1** and **3**. The computer system **500** may include, among other things, an interconnect **510**, a processor **512**, a multimedia adapter **514**, a network interface **516**, a system memory **518**, and a storage adapter **520**.

[0042] The interconnect **510** may interconnect various subsystems, elements, and/or components of the computer system **500**. As shown, the interconnect **510** may be an abstraction that may represent any one or more separate physical buses, point-to-point connections, or both, connected by appropriate bridges, adapters, or controllers. In some examples, the interconnect **510** may include a system bus, a peripheral component interconnect (PCI) bus or PCI-Express bus, a HyperTransport or industry standard architecture (ISA)) bus, a small computer system interface (SCSI) bus, a universal serial bus (USB), IIC (I2C) bus, or an Institute of Electrical and Electronics Engineers (IEEE) standard 1364 bus, or "firewire," or other similar interconnection element.

[0043] In some examples, the interconnect **510** may allow data communication between the processor **512** and system memory **518**, which may include read-only memory (ROM) or flash memory (neither shown), random-access memory (RAM), and/or other non-transitory computer readable media (not shown). It should be appreciated that the RAM may be the main memory into which an operating system and various application programs may be loaded. The ROM or flash memory may contain, among other code, the Basic Input-Output system (BIOS) which controls basic hardware operation such as the interaction with one or more peripheral components.

[0044] The processor **512** may control operations of the computer system **500**. In some examples, the processor **512** may do so by executing instructions such as software or firmware stored in system memory **518** or other data via the storage adapter **520**. In some examples, the processor **512** may be, or may include, one or more programmable general-purpose or special-purpose microprocessors, digital signal processors (DSPs), programmable controllers, application specific integrated circuits (ASICs), programmable logic device (PLDs), trust platform modules (TPMs), field-programmable gate arrays (FPGAs), other processing circuits, or a combination of these and other devices.

[0045] The multimedia adapter **514** may connect to various multimedia elements or peripherals. These may include devices associated with visual (e.g., video card or display), audio (e.g., sound card or speakers), and/or various input/output interfaces (e.g., mouse, keyboard, touchscreen).

[0046] The network interface **516** may provide the computer system **500** with an ability to communicate with a variety of remote devices over a network. The network interface **516** may include, for example, an Ethernet adapter, a Fibre Channel adapter, and/or other wired- or wireless-enabled adapter. The network interface **516** may provide a direct or indirect connection from one network element to another, and facilitate communication and between various network elements.

[0047] The storage adapter **520** may connect to a standard computer readable medium for storage and/or retrieval of information, such as a fixed disk drive (internal or external).

[0048] Other devices, components, elements, or subsystems (not illustrated) may be connected in a similar manner to the interconnect **510** or via a network. The network may include any one or more of, for instance, the Internet, an intranet, a PAN (Personal Area Network), a LAN (Local Area Network), a WAN (Wide Area Network), a SAN (Storage Area Network), a MAN (Metropolitan Area Network), a wireless network, a cellular communications network, a Public Switched Telephone Network, and/or other network.

[0049] The devices and subsystems can be interconnected in different ways from that shown in FIG. **5**. Instructions to implement various examples and implementations described herein may be stored in computer-readable storage media such as one or more of system memory **518** or other storage. Instructions to implement the present disclosure may also be received via one or more interfaces and stored in memory. The operating system provided on computer system **500** may be MS-DOS®, MS-WINDOWS®, OS/2®, OS X®, IOSO, ANDROID®, UNIX®, Linux®, or another operating system.

[0050] In FIG. **1**, as well as in other Figures, different numbers of entities than those depicted may be used. Furthermore, according to various implementations, the components described herein may be implemented in hardware and/or software that configure hardware.

[0051] For simplicity and illustrative purposes, the disclosure included descriptions that may refer to examples. In the description, numerous specific details have been set forth in object to provide a thorough understanding of the present disclosure. It will be readily apparent however, that the disclosure may be practiced without limitation to these specific details. In other instances, some methods and structures have not been described in detail so as not to unnecessarily obscure the present disclosure. In some instances, various method operations may be omitted and the various method operations are not necessarily performed in the object in which they are presented.

[0052] Throughout the disclosure, the term "a" and "an" may be intended to denote at least one of a particular element. As used herein, the term "includes" means includes but not limited to, the term "including" means including but not limited to. The term "based on" means based at least in part on.

[0053] Although described specifically throughout the entirety of the instant disclosure, representative examples of the present disclosure have utility over a wide range of applications, and the above discussion is not intended and should not be construed to be limiting, but is offered as an illustrative discussion of aspects of the disclosure. What has been described and illustrated herein is an example of the disclosure along with some of its variations. The terms, descriptions and figures used herein are set forth by way of illustration only and are not meant as limitations. As such, the disclosure is intended to be defined by the following claims -- and their equivalents -- in which all terms are meant in their broadest reasonable sense unless otherwise indicated.

What is claimed is:

1. A system for reducing data transmission in a network through sequenced input data that is deterministically generated into output data to avoid transmission or persistent storage of the output data and maintains auditability, fault tolerance, and/or recovery, the system comprising:

a processor programmed to:

(i) before initiation of a trading session:

(a) receive, from a credit admin engine, data indicating a plurality of credit lines; and

(b) prune credit lines among the plurality of credit lines that are not part of any bilateral credit lines, wherein each bilateral credit line comprises a first credit line extended from a first user to a second user and a second credit line extended from the second user to the first user;

(c) create a single identifying key for each pair of users in each of the bilateral credit lines;

(d) store the pruned credit lines indexed by the single identifying key for each pair of users;

(ii) after initiation of the trading session:

(a) receive a message that is sequenced from unsequenced data based on a fault-tolerant network protocol, the message indicating a matched trade between a pair of users; and

(b) generate a deterministic credit check result based on the matched trade and the stored pruned credit lines indexed by the single identifying key for the pair of users, thereby avoiding transmission of the deterministic credit check result.

2. The system of claim **1**, wherein to prune the credit lines, the processor is further programmed to:

remove credit lines that are not part of any bilateral credit lines from use before conducting the trading session.

3. The system of claim **1**, wherein to create the single identifying key for each pair of users, the processor is further programmed to:

generate a first unique identifying key for the first user of the pair of users;

generate a second unique identifying key for the second user of the pair of users; and

concatenate the first unique identifying key with the second unique identifying key.

4. The system of claim **3**, wherein the first unique identifying key and the second unique identifying key each comprise different 32-bit integers, and the single identifying key comprises a 64-bit integer.

5. The system of claim **1**, wherein to merge the credit lines within each of the bilateral credit lines, the processor is further programmed to:

determine a first net credit amount for a first line of credit in the bilateral credit line;

determine a second net credit amount for a second line of credit in the bilateral credit line; and

store the smaller of the first net credit and the second net credit as a net credit amount for the single merged credit line.

6. The system of claim **1**, wherein the processor is further programmed to store each of the plurality of credit lines in a hashmap.

7. The system of claim **6**, wherein the processor is further programmed to:

generate an alias for each credit line among the plurality of credit lines that was not pruned to a single data structure reference such that the single data structure reference is used by all users referencing the credit line, wherein the alias reduces an amount of lookup operations on the hashmap during the trading session.

8. The system of claim **1**, wherein the processor is further programmed to generate pseudo-randomization for deterministic computations.

9. The system of claim **8**, wherein to generate the pseudo-randomization, the processor is further programmed to:

receive an input message that requires an action involving randomization;

use a pseudo-random number generator, that takes as its seed, data derived from the input message that is independent of the action involving the randomization, to generate a pseudo-random number to address the action involving the randomization.

10. The system of claim **1**, wherein the processor is further programmed to:

execute a live credit checking during the trading session; and

execute a backup credit checking engine that performs credit checking during the trading session when the live credit checking engine fails.

11. The system of claim **10**, further comprising a sequencer programmed to:

receive unsequenced data from both a credit administration engine and a matching engine;

sequence the unsequenced data; and

transmit the sequenced data to the live credit checking engine, the backup credit checking engine, and a credit administration engine.

12. The system of claim **11**, wherein the unsequenced data from the credit administration engine comprises credit line changes made during the trading session.

13. The system of claim **11**, wherein the unsequenced data from a matching engine comprises information about partially-formed trades.

14. The system of claim **1**, wherein the credit checking engine is further programmed to, during the trading session:

receive a partially-formed trade from a matching engine, the partially-formed trade comprising an agreement between the first user and the second user, conditioned on passing a credit check;

concatenate a first identifier of the first user with a second identifier of the second user to obtain a participant-pair key; and

conduct the credit check using the participant-pair key.

15. The system of claim **14**, wherein the processor is further programmed to

determine that one or more credit lines stored in the system are associated with the participant-pair key;

determine a smallest credit limit among the one or more credit lines;

set an actual trade size of the partially-formed trade to be the lesser of a current trade size of the partially-formed trade and the smallest credit limit;

draw down each of the one or more credit lines by the actual trade size; and

transmit a completed trade message to the matching engine.

16. The system of claim **14**, wherein the processor is further programmed to:

determine that no credit lines are stored in the system that are associated with the participant-pair key; and

send an unwind-trade message to the matching engine.

17. The system of claim **14**, wherein the processor is further programmed to:

determine that one or more credit lines stored in the system are associated with the participant-pair key;

determine a smallest credit limit among the one or more credit lines;

set an actual trade size of the partially-formed trade to be the lesser of a current trade size of the partially-formed trade and the smallest credit limit;

determine that the actual trade size is less than a minimum allowable trade size of the system; and

send an unwind-trade message to the matching engine.

18. A method, comprising:

generating, by a processor, a plurality of credit lines each extending a line of credit from one user to another user;

pruning, by the processor, credit lines among the plurality of credit lines that are not part of any bilateral credit lines, wherein each bilateral credit line comprises a first credit line extended from a first user to a second user and a second credit line extended from the second user to the first user;

creating, by the processor, a single identifying key for each pair of users in each of the bilateral credit lines;

merging, by the processor, the credit lines within each of the bilateral credit lines to generate a single merged credit line for each bilateral credit line;

aliasing, by the processor, each credit line among the plurality of credit lines that was not pruned to a single data structure reference such that the single data structure reference is used by all users referencing the credit line; and

performing, by the processor, credit checks during a trading session using the aliases of each credit line and each single merged credit line for each bilateral credit line.

19. The method of claim **18**, further comprising:

receiving a partially-formed trade, the partially-formed trade comprising an agreement between the first user and the second user, conditioned on passing a credit check;

concatenating a first identifier of the first user with a second identifier of the second user to obtain a participant-pair key;

conducting the credit check using the participant-pair key;

determining that one or more credit lines are associated with the participant-pair key;

determining a smallest credit limit among the one or more credit lines;

setting an actual trade size of the partially-formed trade to be the lesser of a current trade size of the partially-formed trade and the smallest credit limit;

drawing down each of the one or more credit lines by the actual trade size; and

transmitting a completed trade message.

20. A non-transitory computer readable medium comprising instructions that, when executed by a processor, cause the processor to:

generate a plurality of credit lines each extending a line of credit from one user to another user;

prune credit lines among the plurality of credit lines that are not part of any bilateral credit lines, wherein each bilateral credit line comprises a first credit line extended from a first user to a second user and a second credit line extended from the second user to the first user;

create a single identifying key for each pair of users in each of the bilateral credit lines;

merge the credit lines within each of the bilateral credit lines to generate a single merged credit line for each bilateral credit line;

alias each credit line among the plurality of credit lines that was not pruned to a single data structure reference such that the single data structure reference is used by all users referencing the credit line; and

perform a credit check during a trading session using the aliases of each credit line and each single merged credit line for each bilateral credit line.

* * * * *