[00133] In an example embodiment, person A gives person B an X amount of principal; in return, person B posts collateral as a security for the amount received. These two entries can be entered as DSL as in example input string 402, 404, 406, 408, 410, or 412. The user borrower can initiate these transactions using a client device 130 subject to the conditions imposed on those input strings 304. The user borrower (person A) can propose depositing collateral, of a specific type, as in the smart contract request in the example input string 402. The user borrower has an obligation to pay the lender a principal amount seen in the smart contract request in example input string 404 and interest as in the smart contract request in example input string 406. Person B receives, on a client device 130 e.g. client devices 130-3 and 130-4 for user lender, the principal in smart contract request in example input string 408 and the interest in smart contract request in example input string 410.

[00134] Consequently, a smart contract can be broken into multiple conditional smart contract requests. For instance, another example input string 412 may allow person B to claim the collateral, precisely a value of interest plus principal from the collateral, if the conditions of paying principal in example input string 404 and interest in example input string 406 are not met.

[00135] Processing input strings 304 on-chain makes both agreement upgrades and the respective transaction recorded on-chain transparent to all parties; thus, the due diligence for agreement condition upgrades becomes almost automated and recorded as an obligation or allowance on the agreement contract balance sheet. Further, such transactions become immutable records on the blockchain; thus, the entered agreement conditions and associated data cannot be altered or removed. In addition, the transparency of agreement upgrades occurs immediately upon submission of a smart contract request. With a transparent balance sheet along with existing obligations and allowances also recorded on-chain, parties to an agreement can perform immediate audits of the agreement smart contract and its updates to validate agreement condition upgrades with respect to their existing security interests in the agreement. As such, the existing agreement conditions of an agreement contract can be upgraded by providing another input string 304, similar to a signed addendum placed on a legal financial contract. The upgrade can be performed since the conditions are not hardcoded; instead, conditions are broken down into segments, and smart contracts representing the conditions are called, making conditions immutable and transparent despite not being hardcoded.

by the EVM. Once conditions of the example input strings are satisfied, effective double entries are created, one for the lender and one for the borrower, changing asset and liability balances in the capital stack. The example input string 404 may be recorded in the balance sheet as balance sheet entry 808 and example input string 406 may be recorded in the balance sheet as balance sheet entry 810.

[00143] FIG. 9 is a flowchart of a method 900 for computing device sending an input string to the smart contract in accordance with an example embodiment. The method 900 begins at step 902, at which there is a smart contract compiler 302 deployed on a blockchain. The smart contract compiler 302 of FIG. 9 may be configured as in method 700 in FIG. 7 to parse an input string into 304 an object structure, wherein the object structure is in a string format. Further, the smart contract compiler 302 may determine a byte sequence program for the object structure; and map the byte sequence program into at least one executable smart contract representation of the object structure executable by the blockchain. This smart contract compiler 302 is accessible by a client device 130, connected to nodes 120 of the blockchain. Step 902 ends and step 904 begins.

[00144] At step 904, the client device 130 sends an input string having at least one agreement condition. The input string is for a smart contract request (e.g. contract's clause, term, or instruction), such as example input strings 402, 404, 406, 408, 410, and 412. When the at least one agreement condition is satisfied, the at least one executable smart contract representation is executed and the blockchain transaction, which updates the ledger, is recorded on the blockchain.

[00145] At step 906, the client may receive in the computing device or another computing device a confirmation indicating the input string has been sent, or the executable smart contract representation has been executed, or both. Further, due to the nature of blockchain, the client can check whether the input string has been processed or the executable smart contract representation has been executed since the processing and the execution will be recorded on the blockchain. Therefore, the capability to check the recorded blockchain transactions is also a form of confirmation that the client has successfully sent an input string or the executable smart contract has been executed.

[00146] It is to be understood by a person skilled in the art that while the embodiments were taught for Ethereum blockchain and Solidity™, Ethereum and Solidity™ are not meant to

be limitations but examples. The aforementioned embodiments may be equally applicable to other blockchains such as Solana and Cardano, and other blockchains, instead of Ethereum. Further, programming languages other than Solidity™ may be used, including, Vyper™, Yup™, Python™, Rust™, Haskell™, JavaScript™, DAML™.

[00147] The various example embodiments of the systems, devices and/or processes are described using block diagrams, schematics, and examples. Insofar as such block diagrams, schematics, and examples contain one or more functions and/or operations, it will be understood by those skilled in the art that each function and/or operation within such block diagrams, flowcharts, or examples can be implemented, individually and/or collectively, by a wide range of hardware, software, firmware, or virtually any combination thereof. However, those skilled in the art will recognize that the example embodiments, in whole or in part, can be equivalently implemented in standard integrated circuits, as one or more computer programs executed by one or more computers (e.g., as one or more programs running on one or more computer systems), as one or more programs executed by on one or more controllers (e.g., microcontrollers), as one or more programs executed by one or more processors (e.g., microprocessors, central processing units, graphical processing units), as firmware, or as virtually any combination thereof, and that designing the circuitry and/or writing the code for the software and or firmware would be well within the skill of one of ordinary skill in the art in light of the teachings herein.

[00148] The described methods may be carried out by modules, routines, or subroutines of software executed by the node computing device 200. Coding of software for carrying out the steps of the methods is well within the scope of a person of ordinary skill in the art having regard to the methods of compiling smart contracts. The compiler method 700 for compiling a smart contract may contain additional or fewer steps than shown and described, and the steps may be performed in a different order. Computer-readable instructions, executable by the processor(s) of the node computing device 200, may be stored in the memory 210 of the node computing device 200 or a computer-readable medium. It is to be emphasized that the steps for compiling a smart contract need not be performed in the exact sequence as shown unless otherwise indicated. Likewise, various steps of the methods may be performed in parallel rather than in sequence.

[00149] It can be appreciated that the compiler method 700, once implemented, can be performed by the node computing device 200 in a fully automatic manner, which is convenient for users to use as no manual interaction is needed.

[00150] It should be understood by a person skilled in the art that, for the purpose of convenient and brief description, for a detailed working process of the foregoing system, apparatus, and unit, refer to a corresponding process in the foregoing method embodiments, and details are not described herein again.

[00151] In the several embodiments described, it should be understood that the described systems and methods may be implemented in other manners. For example, the described system embodiments are merely examples. Further, units or components may be combined or integrated into another system, or some features may be ignored or not performed. In addition, the displayed or discussed mutual couplings or direct couplings or communication connections may be implemented by using some interfaces. The indirect couplings or communication connections between the systems or units may be implemented in electronic, mechanical, or other forms.

[00152] The example embodiments may be embodied in other specific forms without departing from the subject matter of the claims. The described example embodiments are to be considered in all respects as being only illustrative and not restrictive. Selected features from one or more of the above-described embodiments may be combined to create alternative embodiments not explicitly described, features suitable for such combinations being understood within the scope of the example embodiments.

[00153] Also, although the example systems, devices and processes for compiling a smart contract herein may comprise a specific number of elements/components, the systems, devices, and assemblies could be modified to include additional or fewer of such elements/components. For example, although any of the elements/components disclosed may be referenced as being singular, the embodiments disclosed herein could be modified to include a plurality of such elements/components. The subject matter described herein intends to cover and embrace all suitable changes in technology.

[00154] The units described as separate parts may or may not be physically separate, and parts displayed as units may or may not be physical units, may be located in one position, or may

be distributed on a plurality of network units. Some or all of the units may be selected according to actual requirements to achieve the objectives of the solutions of the embodiments.

[00155] In addition, functional units in the example embodiments may be integrated into one node computing device 200, or each of the units may exist alone physically, or two or more units are integrated into one unit.

[00156] When the functions are implemented in the form of a software functional unit and sold or used as an independent product, they may be stored in a storage medium and include several instructions for instructing a computer device (which may be a personal computer, a server, or a network device) to perform all or some of the steps of the methods described in the embodiments of this application. The foregoing storage medium includes any medium that can store program code, such as a universal serial bus (USB) flash drive, a removable hard disk, a read-only memory (ROM), a random access memory (RAM), a magnetic disk, or an optical disc, among others.

[00157] The foregoing descriptions are merely specific implementations but are not intended to limit the scope of protection. Any variation or replacement readily figured out by a person skilled in the art within the technical scope shall fall within the scope of protection. Therefore, the scope of protection shall be subject to the protection scope of the claims.