US 20250256666A1

(19) United States

(12) Patent Application Publication

Gaither et al.

(10) Pub. No.: US 2025/0256666 A1

(43) Pub. Date: Aug. 14, 2025

(54) **PREDICTING EFFICIENCY OF PROVIDED ELECTRICITY DURING AN OUTAGE**

(71) Applicant: **TOYOTA MOTOR NORTH AMERICA, INC.**, Plano, TX (US)

(72) Inventors: **Geoffrey David Gaither**, Ann Arbor, MI (US); **Jonathan M. Shum**, Canton, MI (US); **Allison McGowan**, Ann Arbor, CA (US)

(73) Assignees: **TOYOTA MOTOR NORTH AMERICA, INC.**, Plano, TX (US); **TOYOTA JIDOSHA KABUSHIKI KAISHA**, AICHI-KEN (JP)

(21) Appl. No.: **18/439,755**

(22) Filed: **Feb. 12, 2024**

Publication Classification

(51) **Int. Cl.**

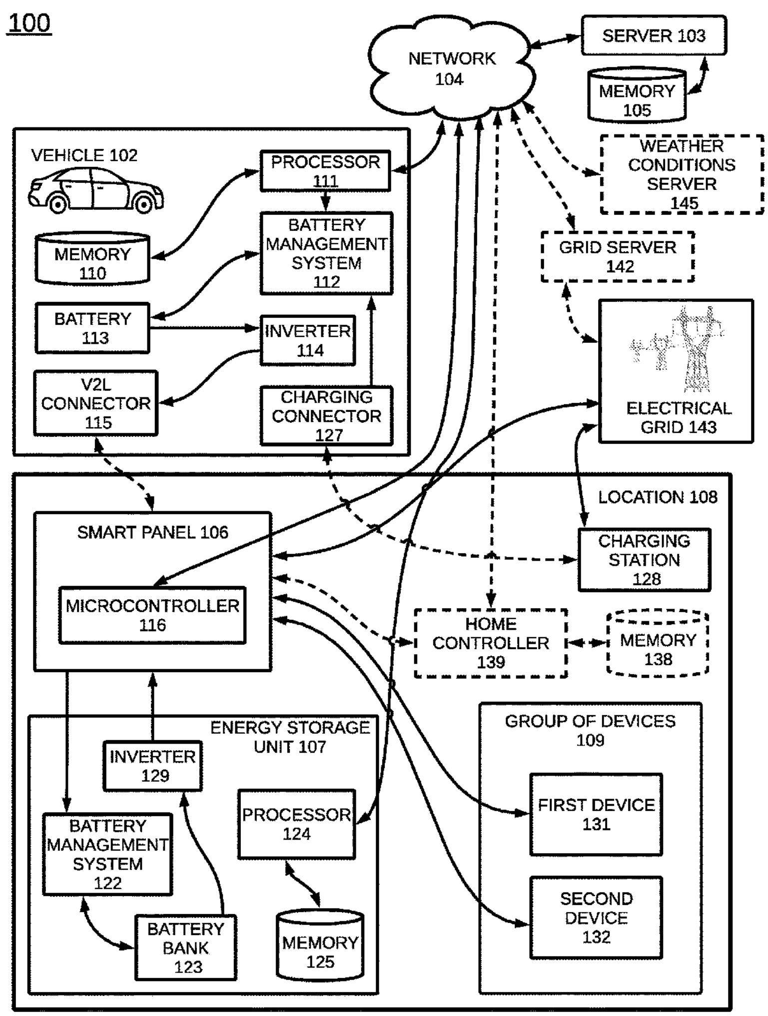
B60R 16/033	(2006.01)
B60L 1/00	(2006.01)
B60L 53/62	(2019.01)
B60L 55/00	(2019.01)
H02J 7/00	(2006.01)

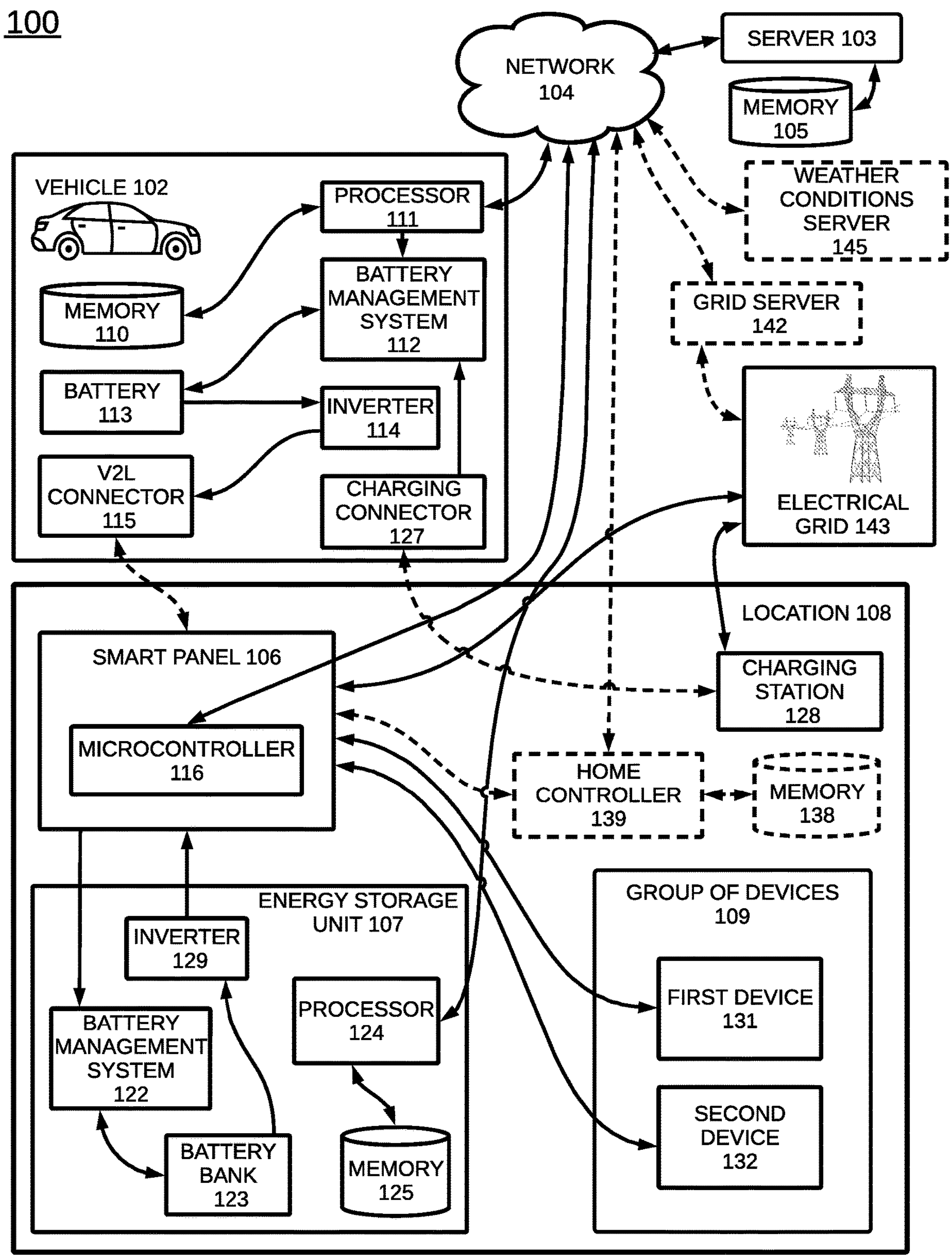
(52) **U.S. Cl.**

CPC ***B60R 16/033*** (2013.01); ***H02J 7/0063*** (2013.01); ***H02J 7/007182*** (2020.01); *B60L 1/00* (2013.01); *B60L 53/62* (2019.02); *B60L 55/00* (2019.02)

(57) **ABSTRACT**

An example operation includes one or more of predicting, by a vehicle, a duration and a severity of an electrical-related event, determining, by the vehicle, a behavior profile of a group of devices at a location, based on a need of at least one of the group of devices during the event, and an amount of energy consumption of at least one of the group of devices during the event, predicting, by the vehicle, an energy requirement during the event at the location, based on the predicted duration, the predicted severity and the determined behavior profile, directing, by the vehicle, energy to be stored prior to the event by both the vehicle and an energy storage unit (ESU) associated with the location, and directing, by the vehicle, a flow of energy to the group of devices during the event, from the ESU and the vehicle.





100
NETWORK 104
SERVER 103
MEMORY 105
WEATHER CONDITIONS SERVER 145
GRID SERVER 142
ELECTRICAL GRID 143
VEHICLE 102
PROCESSOR 111
MEMORY 110
BATTERY MANAGEMENT SYSTEM 112
BATTERY 113
INVERTER 114
V2L CONNECTOR 115
CHARGING CONNECTOR 127
LOCATION 108
SMART PANEL 106
MICROCONTROLLER 116
CHARGING STATION 128
HOME CONTROLLER 139
MEMORY 138
ENERGY STORAGE UNIT 107
INVERTER 129
BATTERY MANAGEMENT SYSTEM 122
PROCESSOR 124
BATTERY BANK 123
MEMORY 125
GROUP OF DEVICES 109
FIRST DEVICE 131
SECOND DEVICE 132

FIG. 1A

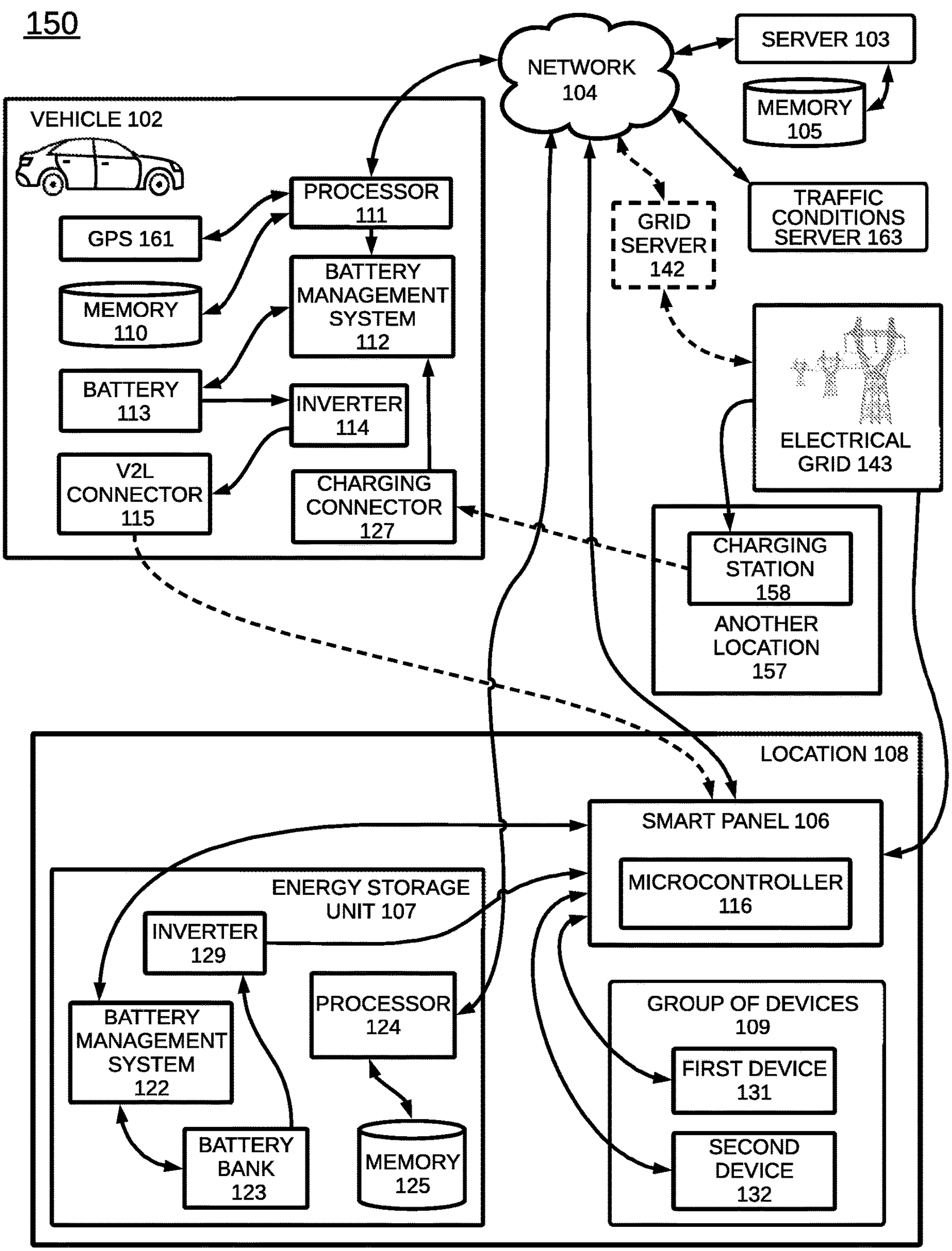

150
NETWORK 104
SERVER 103
MEMORY 105
VEHICLE 102
PROCESSOR 111
GPS 161
MEMORY 110
BATTERY MANAGEMENT SYSTEM 112
BATTERY 113
INVERTER 114
V2L CONNECTOR 115
CHARGING CONNECTOR 127
GRID SERVER 142
TRAFFIC CONDITIONS SERVER 163
ELECTRICAL GRID 143
CHARGING STATION 158
ANOTHER LOCATION 157
LOCATION 108
SMART PANEL 106
MICROCONTROLLER 116
ENERGY STORAGE UNIT 107
INVERTER 129
PROCESSOR 124
BATTERY MANAGEMENT SYSTEM 122
BATTERY BANK 123
MEMORY 125
GROUP OF DEVICES 109
FIRST DEVICE 131
SECOND DEVICE 132

FIG. 1B

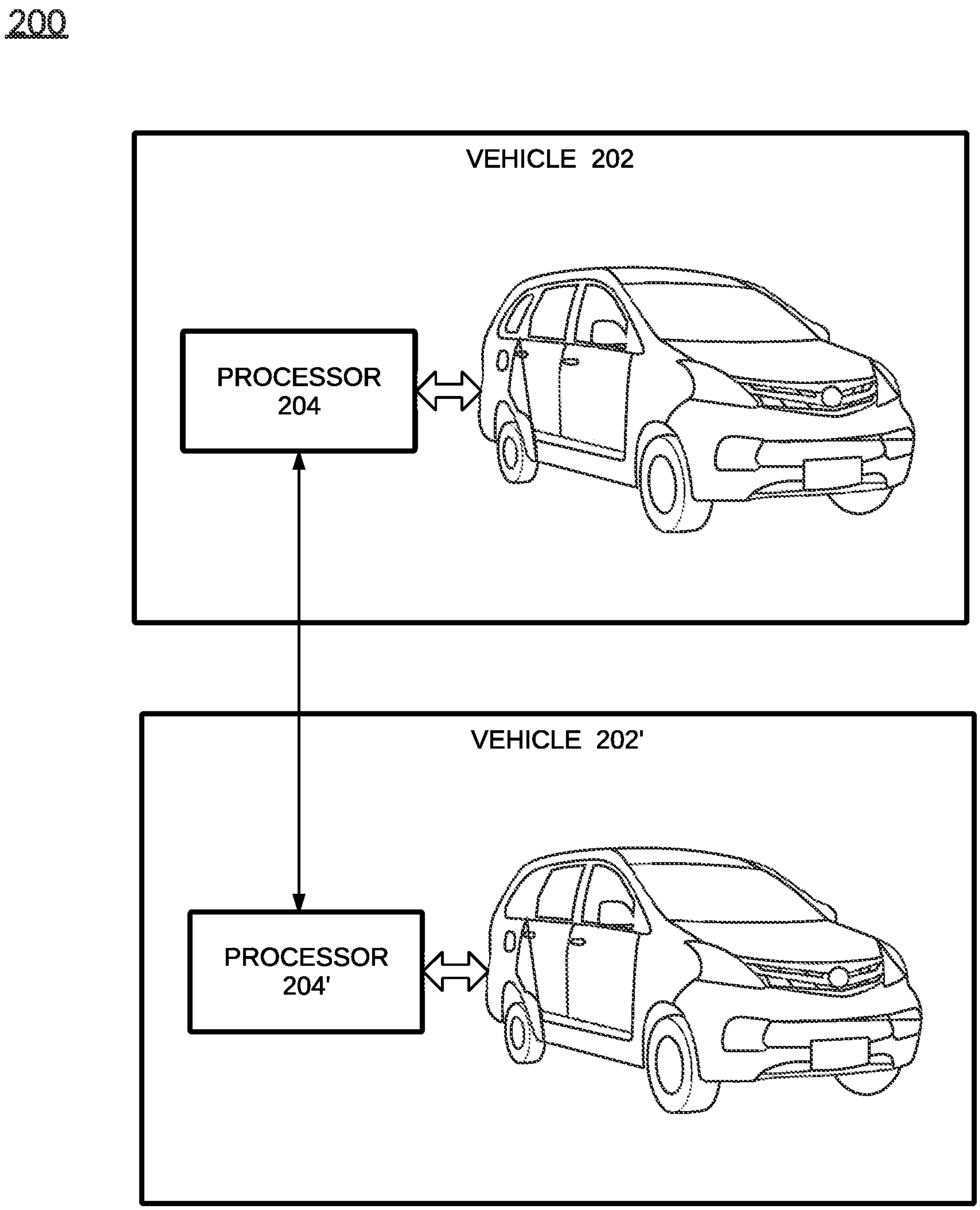


FIG. 2A

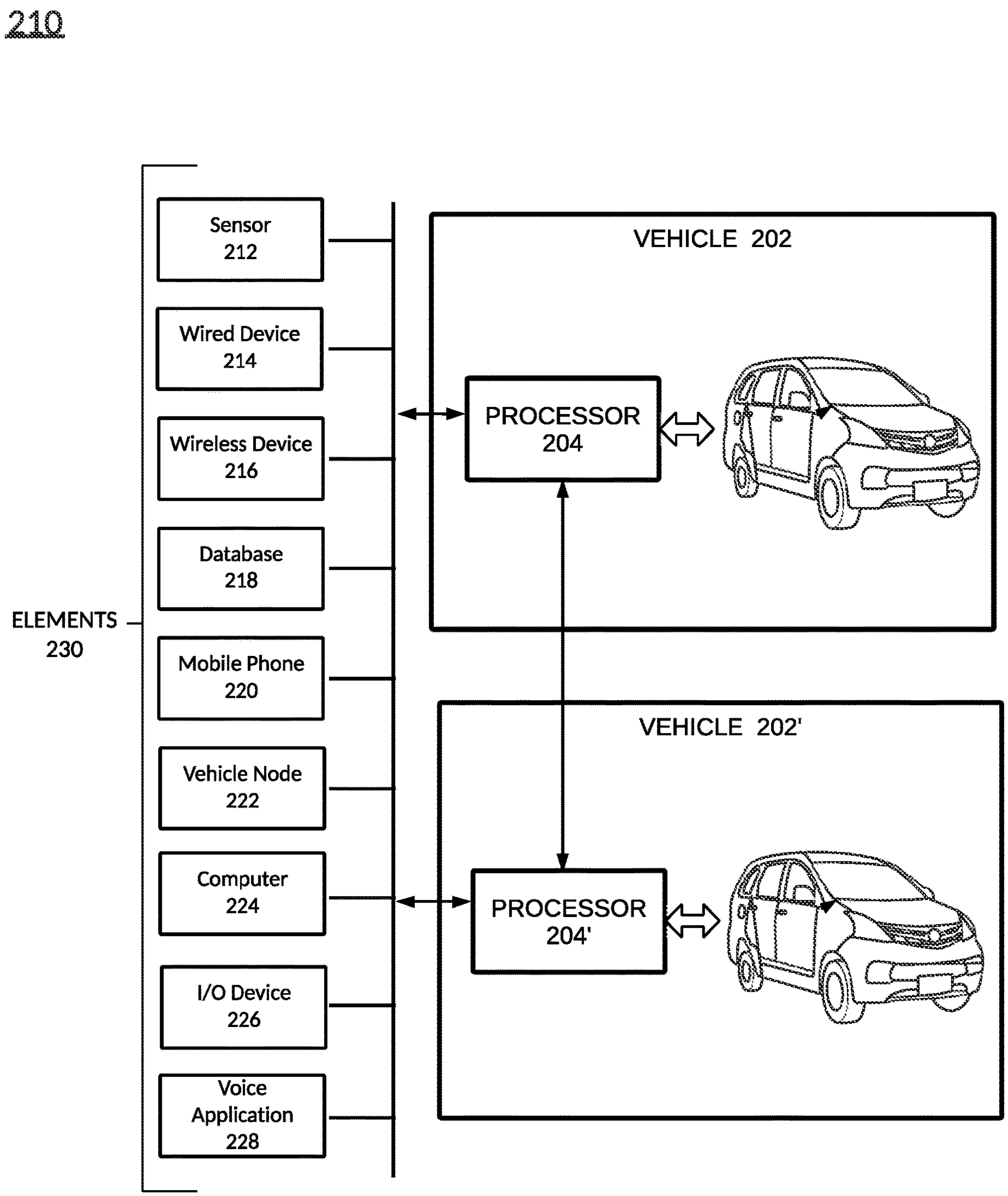

210
Sensor
212
Wired Device
214
Wireless Device
216
Database
218
Mobile Phone
220
Vehicle Node
222
Computer
224
I/O Device
226
Voice
Application
228
ELEMENTS
230
VEHICLE 202
PROCESSOR
204
VEHICLE 202'
PROCESSOR
204'

FIG. 2B

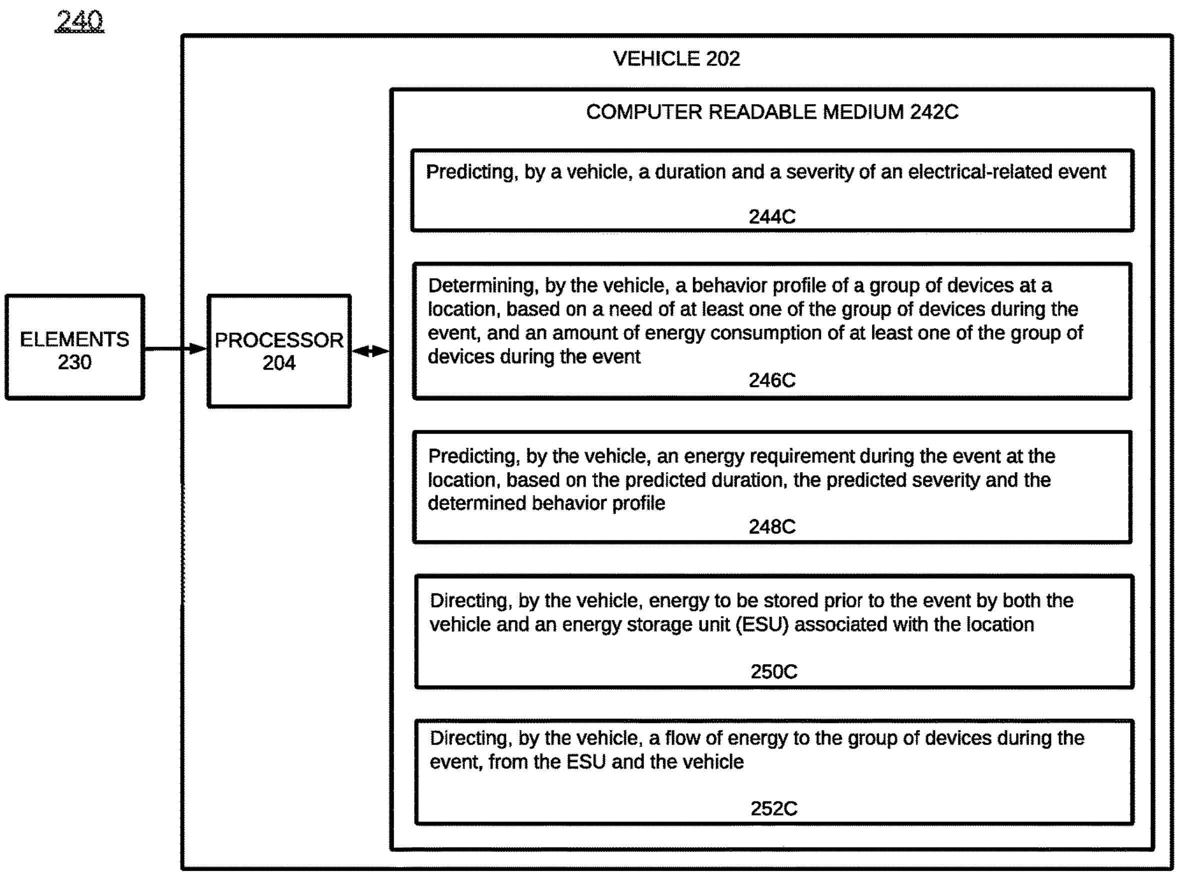

240
VEHICLE 202
COMPUTER READABLE MEDIUM 242C
Predicting, by a vehicle, a duration and a severity of an electrical-related event
244C
ELEMENTS 230
PROCESSOR 204
Determining, by the vehicle, a behavior profile of a group of devices at a location, based on a need of at least one of the group of devices during the event, and an amount of energy consumption of at least one of the group of devices during the event
246C
Predicting, by the vehicle, an energy requirement during the event at the location, based on the predicted duration, the predicted severity and the determined behavior profile
248C
Directing, by the vehicle, energy to be stored prior to the event by both the vehicle and an energy storage unit (ESU) associated with the location
250C
Directing, by the vehicle, a flow of energy to the group of devices during the event, from the ESU and the vehicle
252C

FIG. 2C

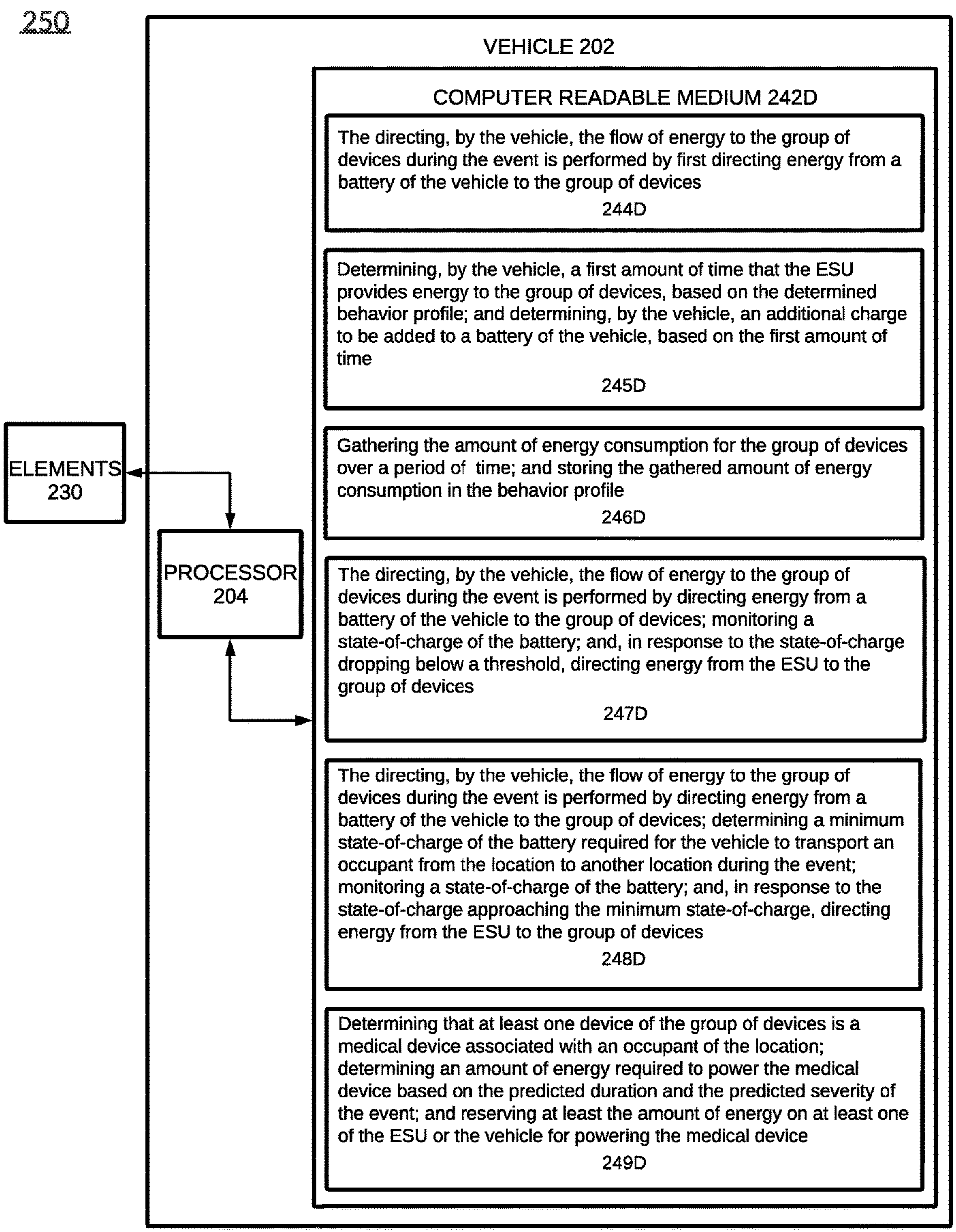

250
VEHICLE 202
COMPUTER READABLE MEDIUM 242D
The directing, by the vehicle, the flow of energy to the group of devices during the event is performed by first directing energy from a battery of the vehicle to the group of devices
244D
Determining, by the vehicle, a first amount of time that the ESU provides energy to the group of devices, based on the determined behavior profile; and determining, by the vehicle, an additional charge to be added to a battery of the vehicle, based on the first amount of time
245D
Gathering the amount of energy consumption for the group of devices over a period of time; and storing the gathered amount of energy consumption in the behavior profile
246D
The directing, by the vehicle, the flow of energy to the group of devices during the event is performed by directing energy from a battery of the vehicle to the group of devices; monitoring a state-of-charge of the battery; and, in response to the state-of-charge dropping below a threshold, directing energy from the ESU to the group of devices
247D
The directing, by the vehicle, the flow of energy to the group of devices during the event is performed by directing energy from a battery of the vehicle to the group of devices; determining a minimum state-of-charge of the battery required for the vehicle to transport an occupant from the location to another location during the event; monitoring a state-of-charge of the battery; and, in response to the state-of-charge approaching the minimum state-of-charge, directing energy from the ESU to the group of devices
248D
Determining that at least one device of the group of devices is a medical device associated with an occupant of the location; determining an amount of energy required to power the medical device based on the predicted duration and the predicted severity of the event; and reserving at least the amount of energy on at least one of the ESU or the vehicle for powering the medical device
249D
ELEMENTS 230
PROCESSOR 204

FIG. 2D

260

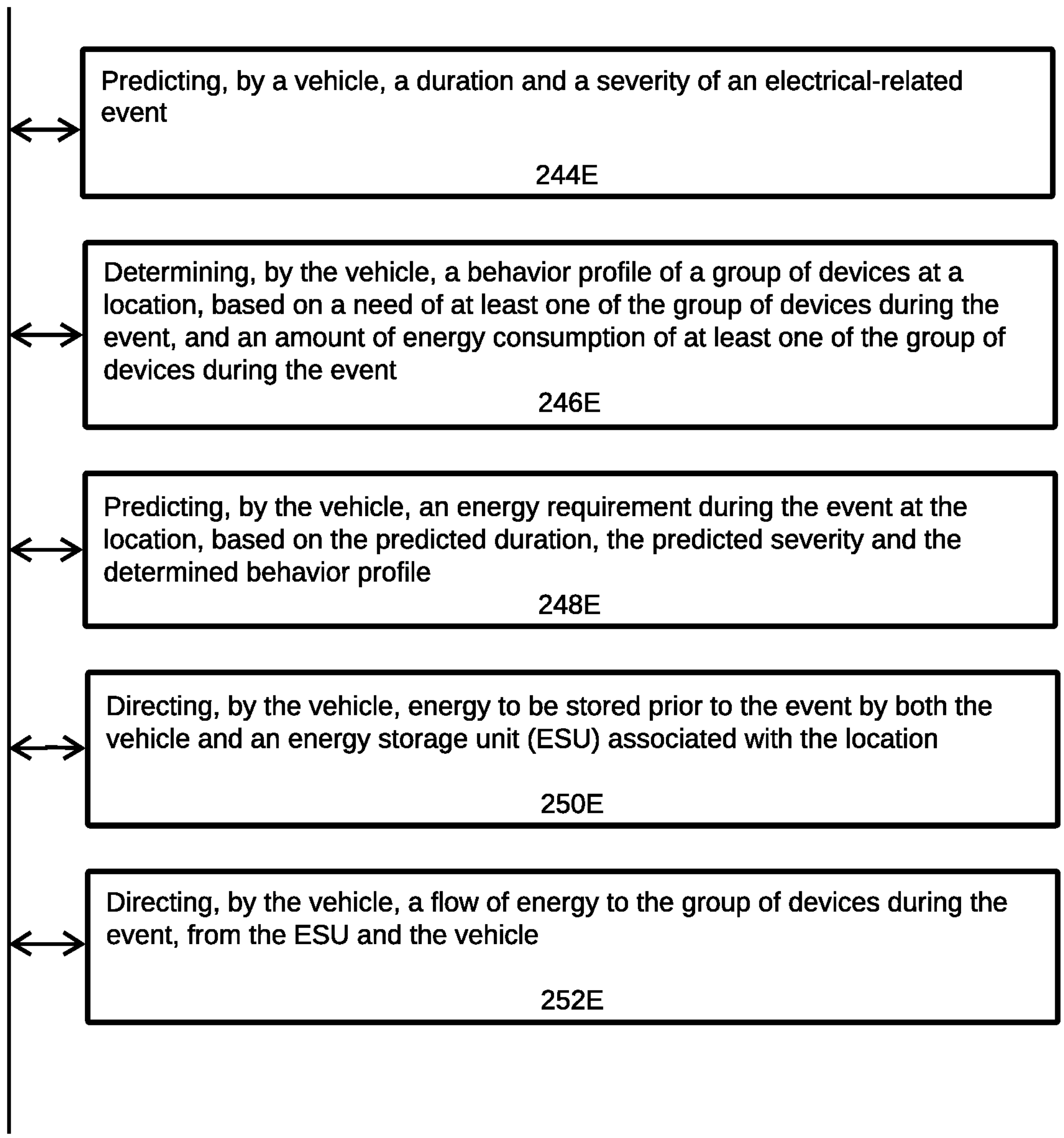

Predicting, by a vehicle, a duration and a severity of an electrical-related event
244E
Determining, by the vehicle, a behavior profile of a group of devices at a location, based on a need of at least one of the group of devices during the event, and an amount of energy consumption of at least one of the group of devices during the event
246E
Predicting, by the vehicle, an energy requirement during the event at the location, based on the predicted duration, the predicted severity and the determined behavior profile
248E
Directing, by the vehicle, energy to be stored prior to the event by both the vehicle and an energy storage unit (ESU) associated with the location
250E
Directing, by the vehicle, a flow of energy to the group of devices during the event, from the ESU and the vehicle
252E

FIG. 2E

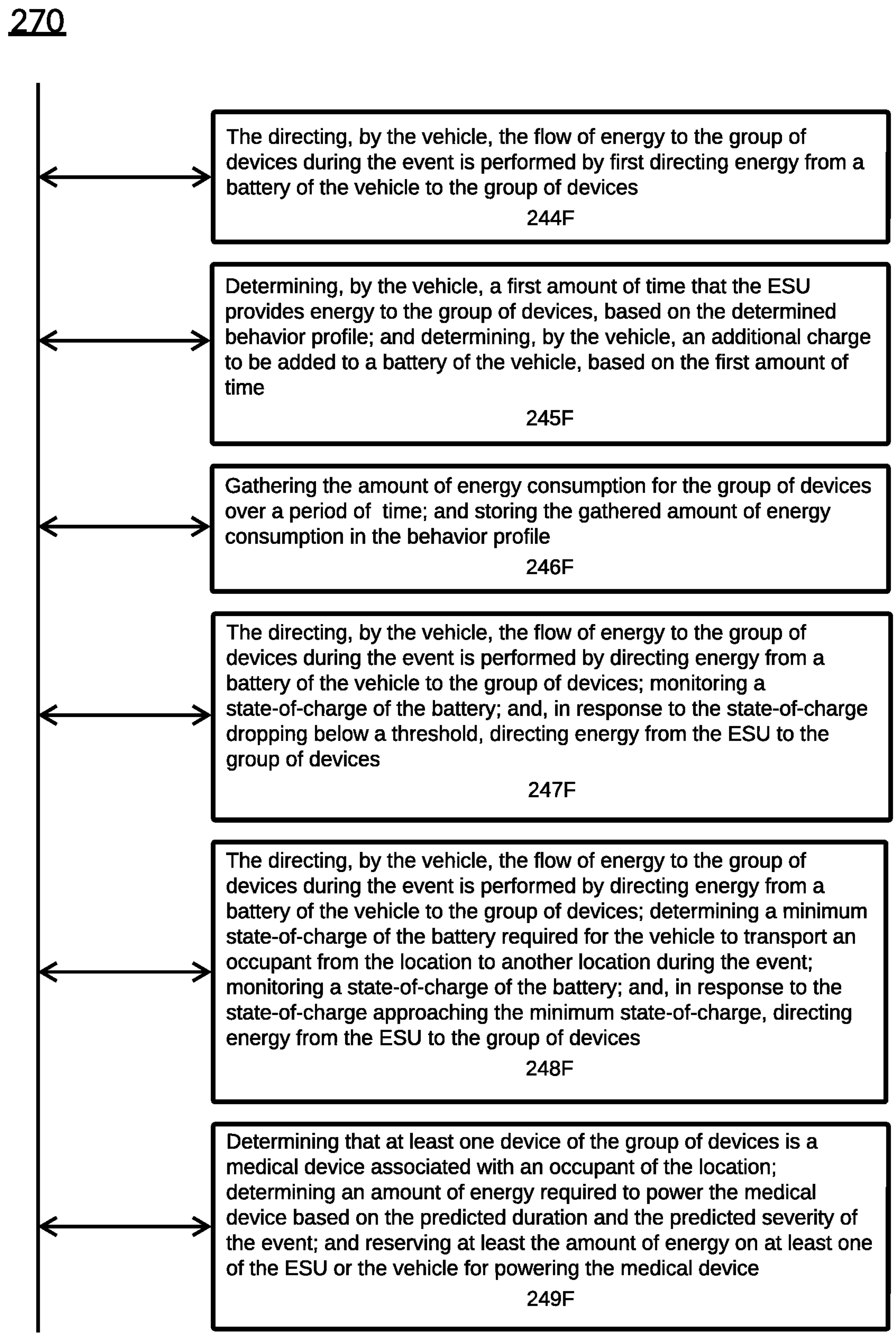

270
The directing, by the vehicle, the flow of energy to the group of devices during the event is performed by first directing energy from a battery of the vehicle to the group of devices
244F
Determining, by the vehicle, a first amount of time that the ESU provides energy to the group of devices, based on the determined behavior profile; and determining, by the vehicle, an additional charge to be added to a battery of the vehicle, based on the first amount of time
245F
Gathering the amount of energy consumption for the group of devices over a period of time; and storing the gathered amount of energy consumption in the behavior profile
246F
The directing, by the vehicle, the flow of energy to the group of devices during the event is performed by directing energy from a battery of the vehicle to the group of devices; monitoring a state-of-charge of the battery; and, in response to the state-of-charge dropping below a threshold, directing energy from the ESU to the group of devices
247F
The directing, by the vehicle, the flow of energy to the group of devices during the event is performed by directing energy from a battery of the vehicle to the group of devices; determining a minimum state-of-charge of the battery required for the vehicle to transport an occupant from the location to another location during the event; monitoring a state-of-charge of the battery; and, in response to the state-of-charge approaching the minimum state-of-charge, directing energy from the ESU to the group of devices
248F
Determining that at least one device of the group of devices is a medical device associated with an occupant of the location; determining an amount of energy required to power the medical device based on the predicted duration and the predicted severity of the event; and reserving at least the amount of energy on at least one of the ESU or the vehicle for powering the medical device
249F

FIG. 2F

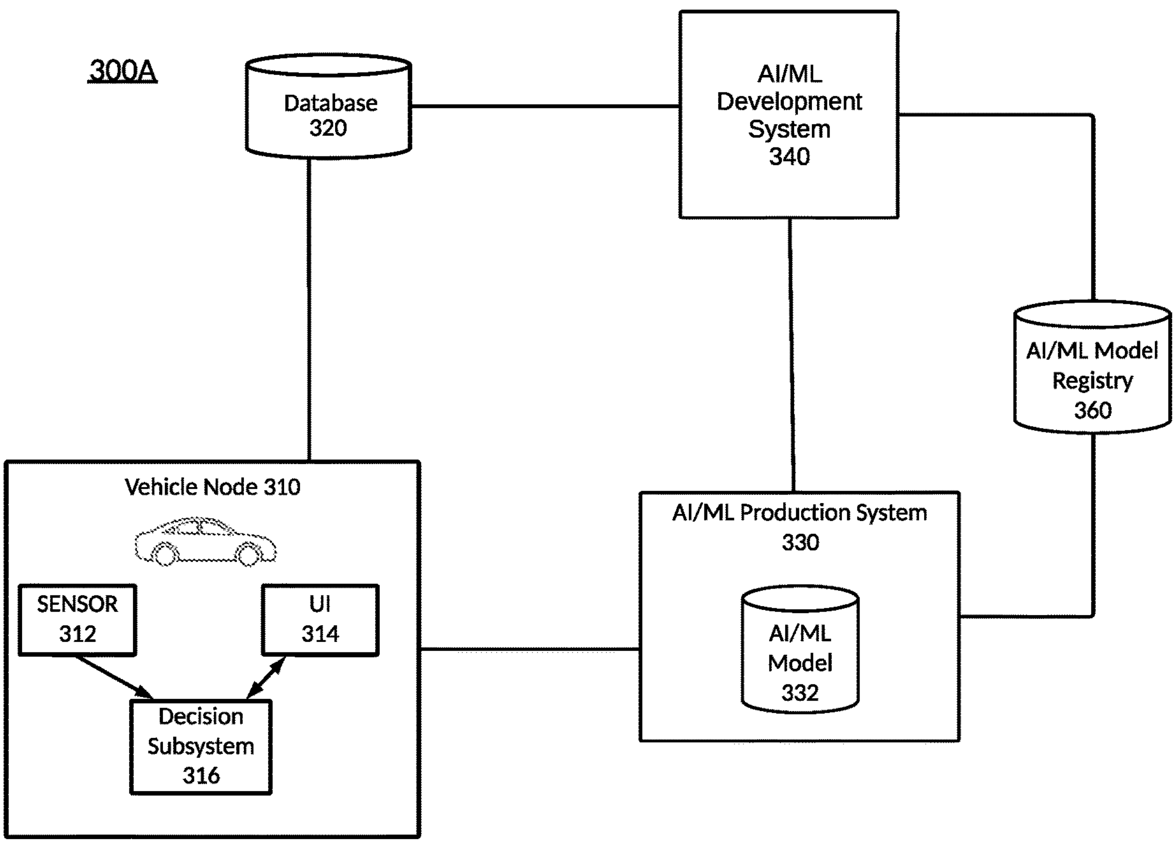

300A
Database
320
AI/ML
Development
System
340
AI/ML Model
Registry
360
Vehicle Node 310
SENSOR
312
UI
314
Decision
Subsystem
316
AI/ML Production System
330
AI/ML
Model
332

FIG. 3A

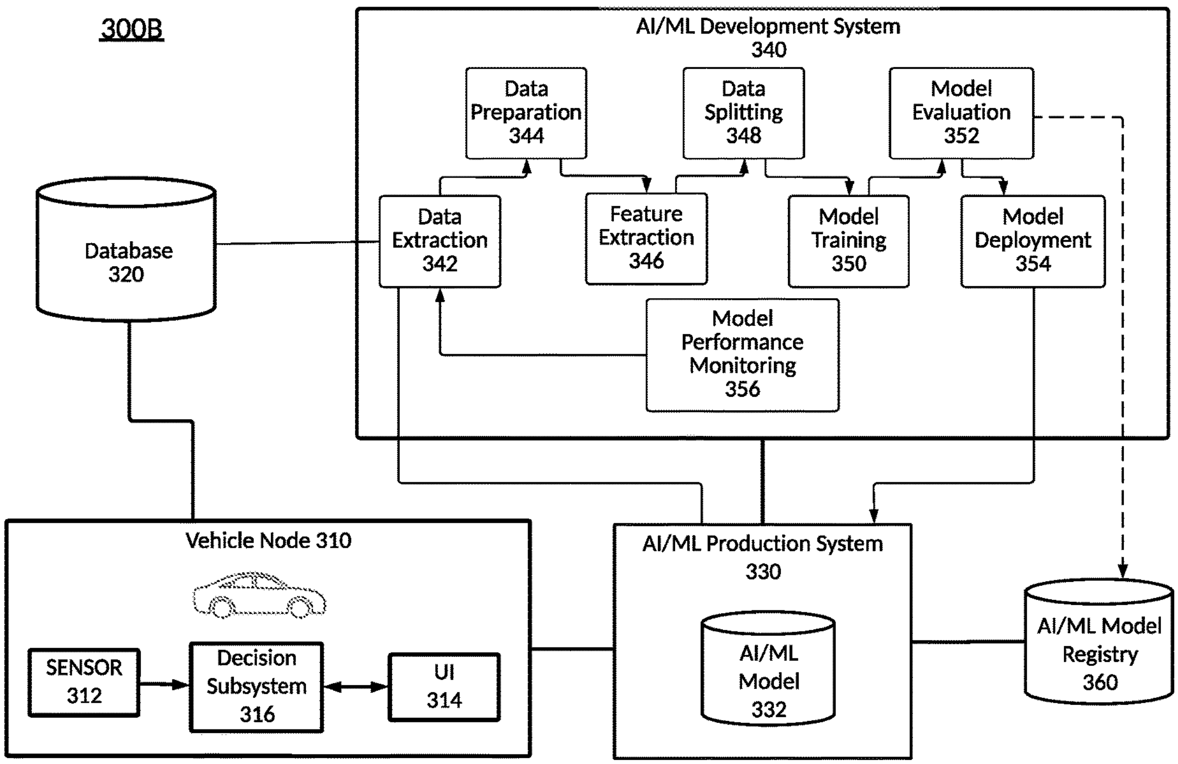

300B
AI/ML Development System
340
Data
Preparation
344
Data
Splitting
348
Model
Evaluation
352
Data
Extraction
342
Feature
Extraction
346
Model
Training
350
Model
Deployment
354
Model
Performance
Monitoring
356
Database
320
Vehicle Node 310
SENSOR
312
Decision
Subsystem
316
UI
314
AI/ML Production System
330
AI/ML
Model
332
AI/ML Model
Registry
360

FIG. 3B

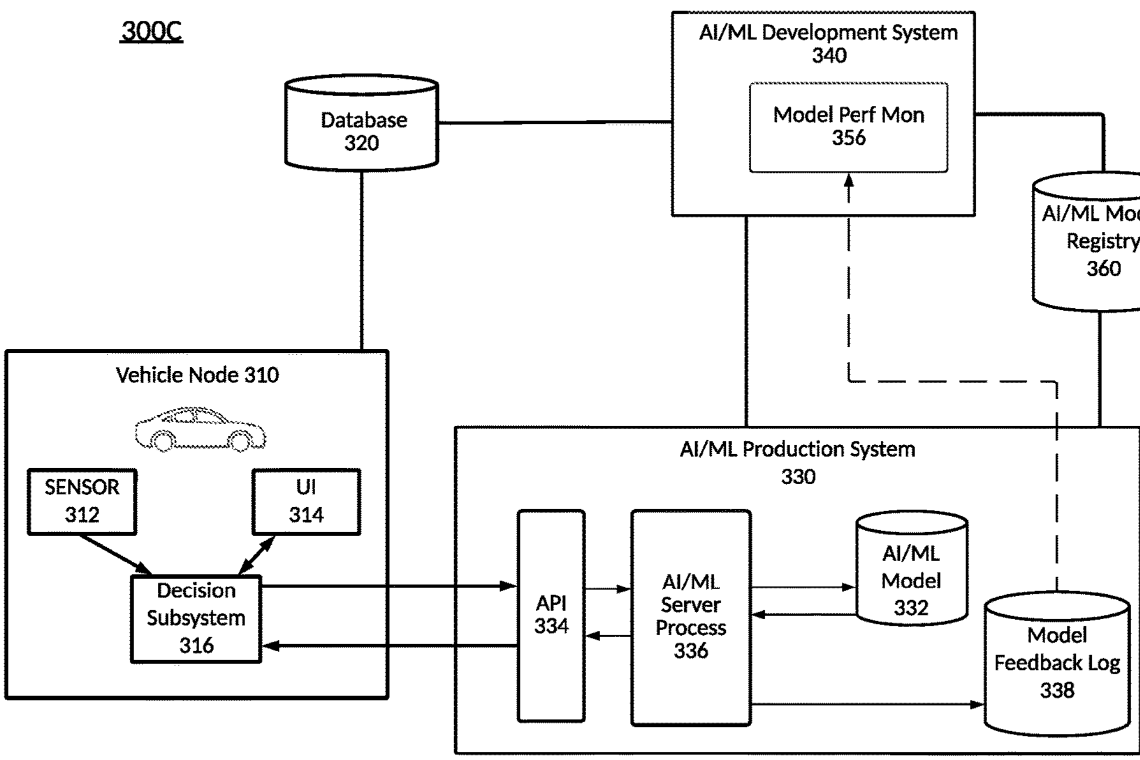

300C
Database
320
AI/ML Development System
340
Model Perf Mon
356
AI/ML Model
Registry
360
Vehicle Node 310
SENSOR
312
UI
314
Decision
Subsystem
316
AI/ML Production System
330
API
334
AI/ML
Server
Process
336
AI/ML
Model
332
Model
Feedback Log
338

FIG. 3C

300D

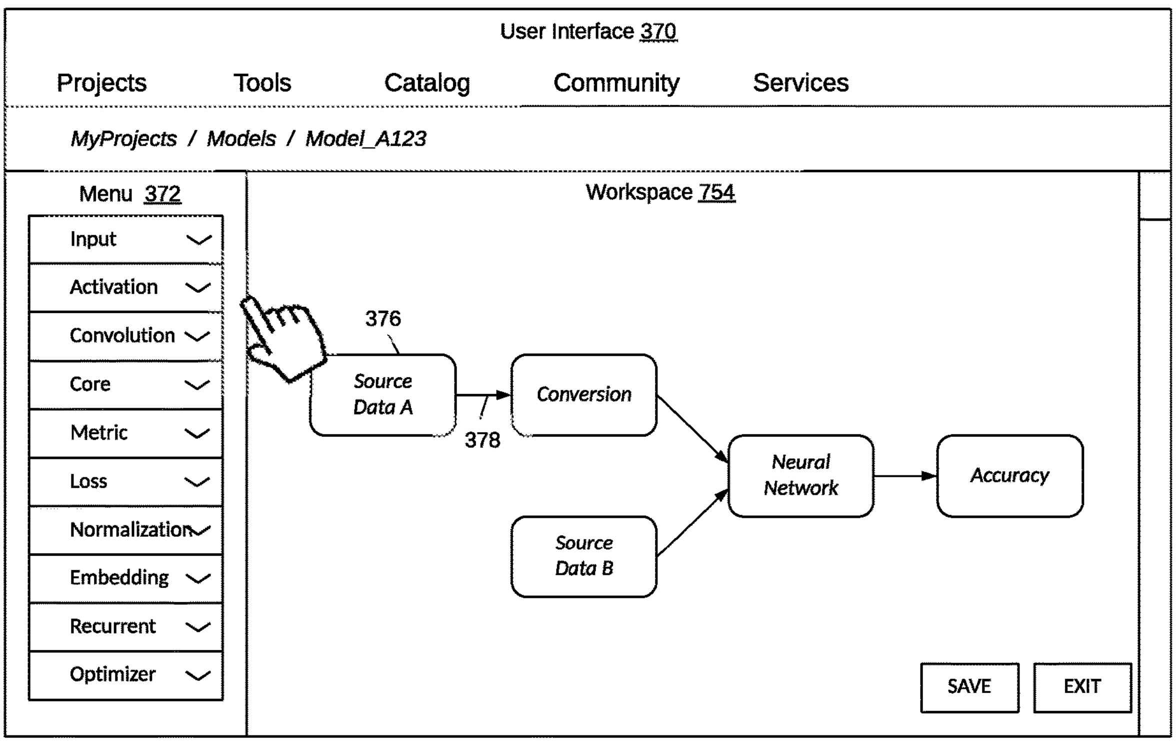

User Interface 370
Projects
Tools
Catalog
Community
Services
MyProjects / Models / Model_A123
Menu 372
Input
Activation
Convolution
Core
Metric
Loss
Normalization
Embedding
Recurrent
Optimizer
Workspace 754
376
Source Data A
378
Conversion
Neural Network
Accuracy
Source Data B
SAVE
EXIT

FIG. 3D

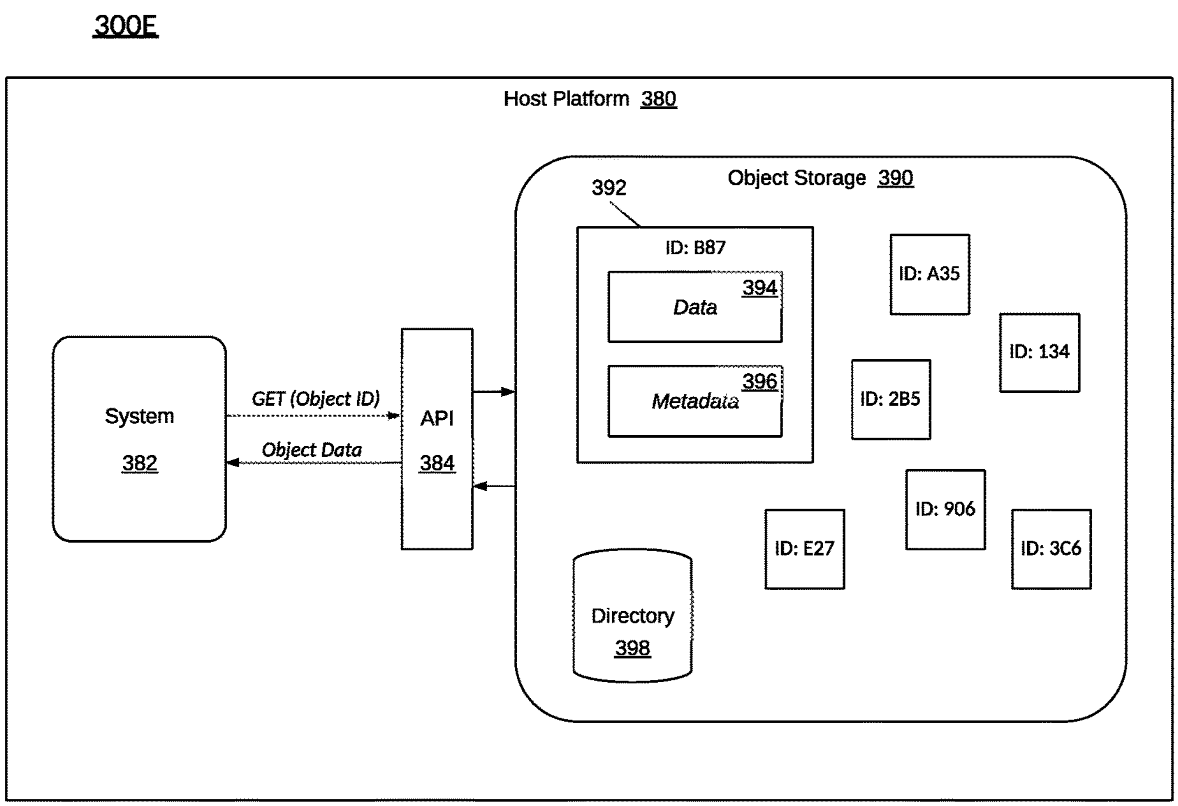

300E
Host Platform 380
Object Storage 390
392
ID: B87
394
Data
396
Metadata
ID: A35
ID: 134
ID: 2B5
ID: 906
ID: E27
ID: 3C6
Directory
398
System
382
GET (Object ID)
Object Data
API
384

FIG. 3E

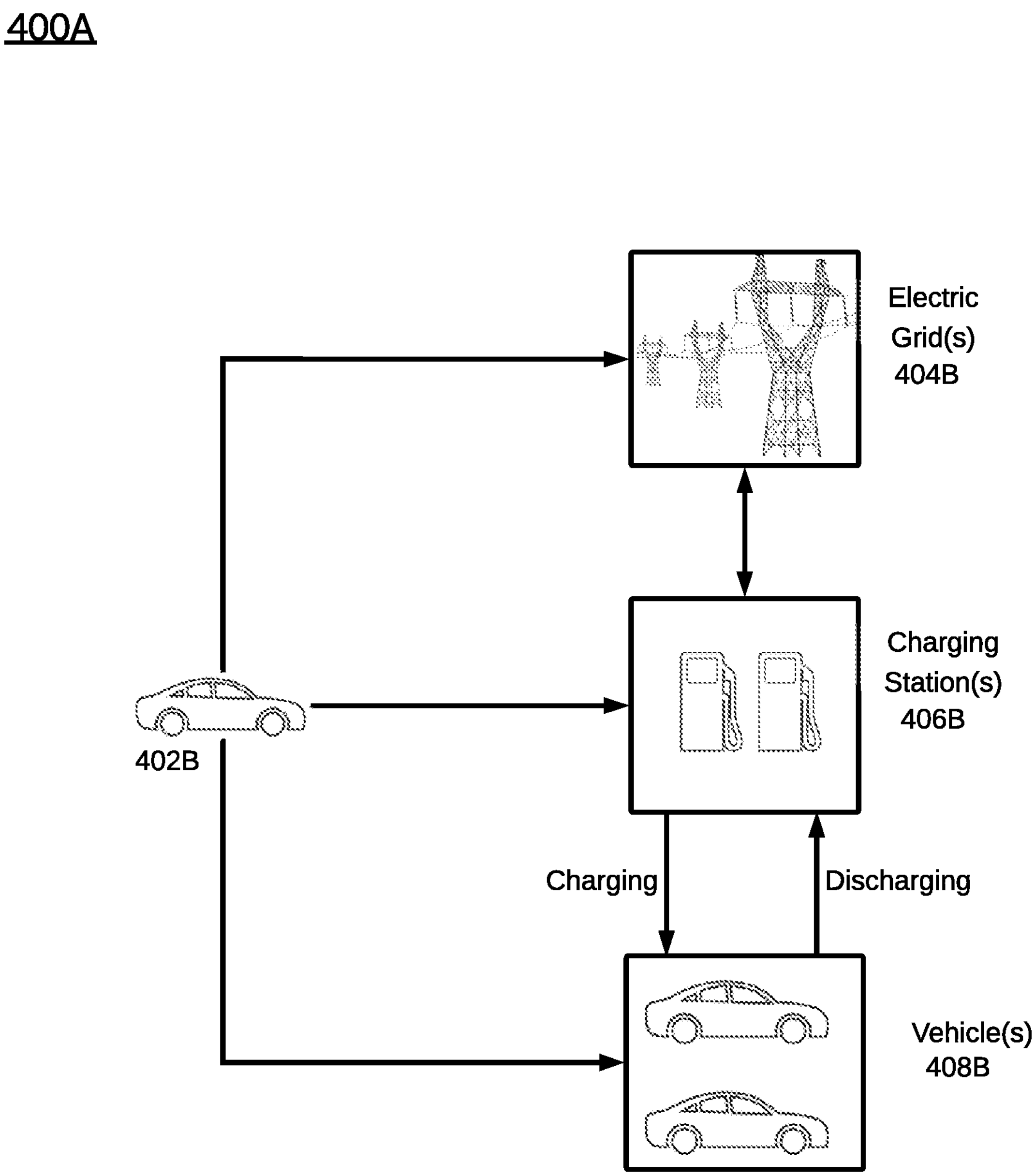

400A
Electric Grid(s) 404B
Charging Station(s) 406B
402B
Charging
Discharging
Vehicle(s) 408B

FIG. 4A

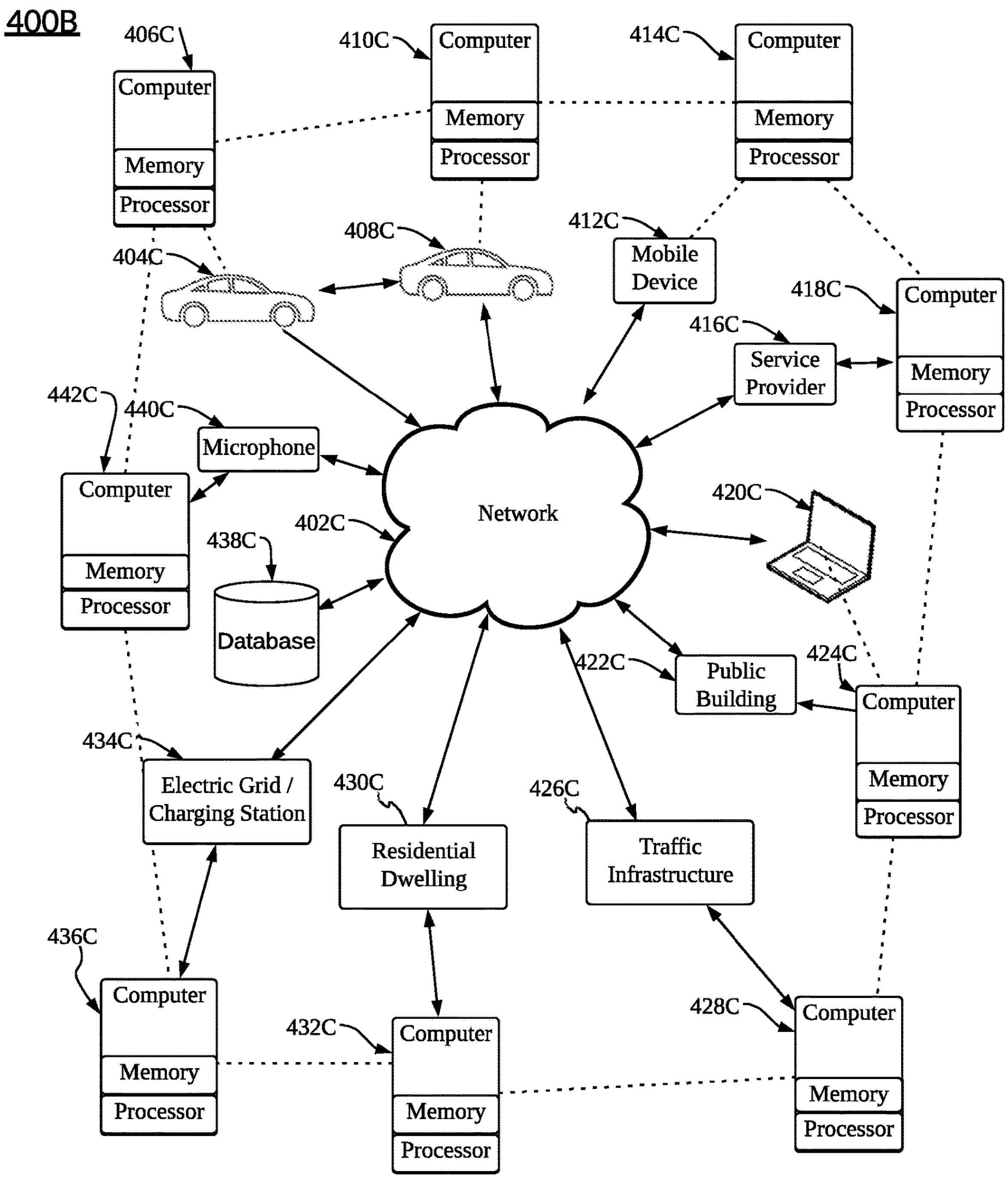

400B
406C
Computer
Memory
Processor
410C
Computer
Memory
Processor
414C
Computer
Memory
Processor
404C
408C
412C
Mobile Device
418C
Computer
Memory
Processor
416C
Service Provider
442C
440C
Microphone
Computer
Memory
Processor
Network
402C
438C
Database
420C
422C
Public Building
424C
Computer
Memory
Processor
434C
Electric Grid / Charging Station
430C
Residential Dwelling
426C
Traffic Infrastructure
436C
Computer
Memory
Processor
432C
Computer
Memory
Processor
428C
Computer
Memory
Processor

FIG. 4B

400C

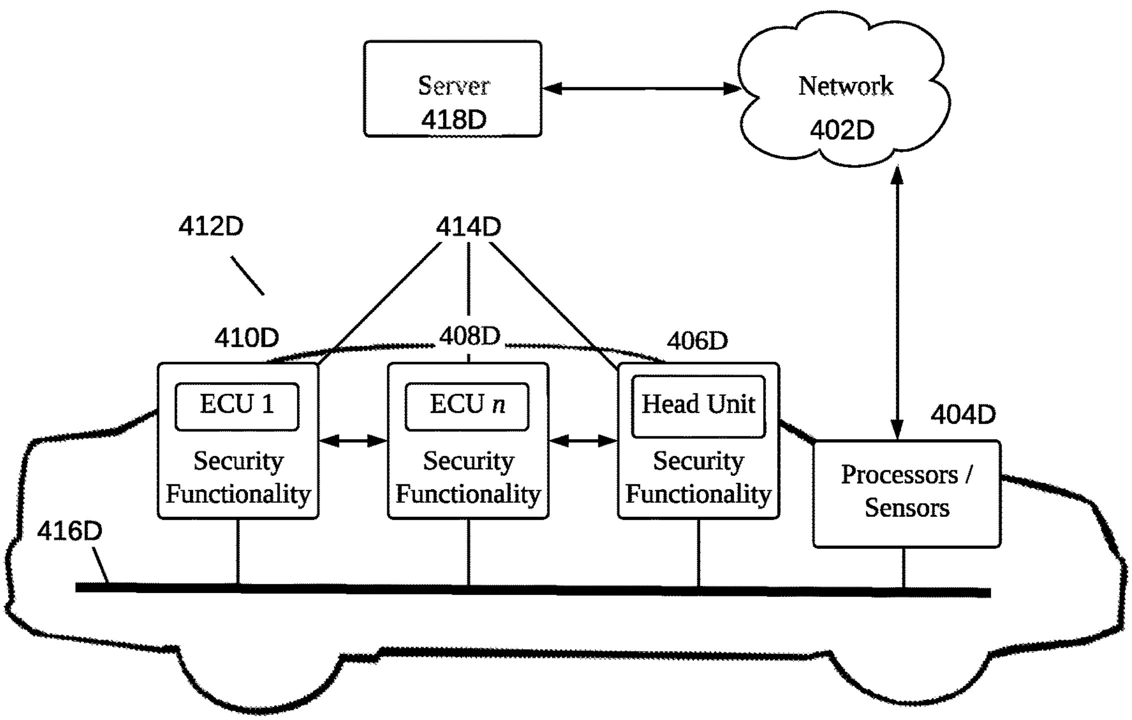

Server
418D
Network
402D
412D
414D
410D
408D
406D
ECU 1
Security Functionality
ECU n
Security Functionality
Head Unit
Security Functionality
404D
Processors / Sensors
416D

FIG. 4C

400D

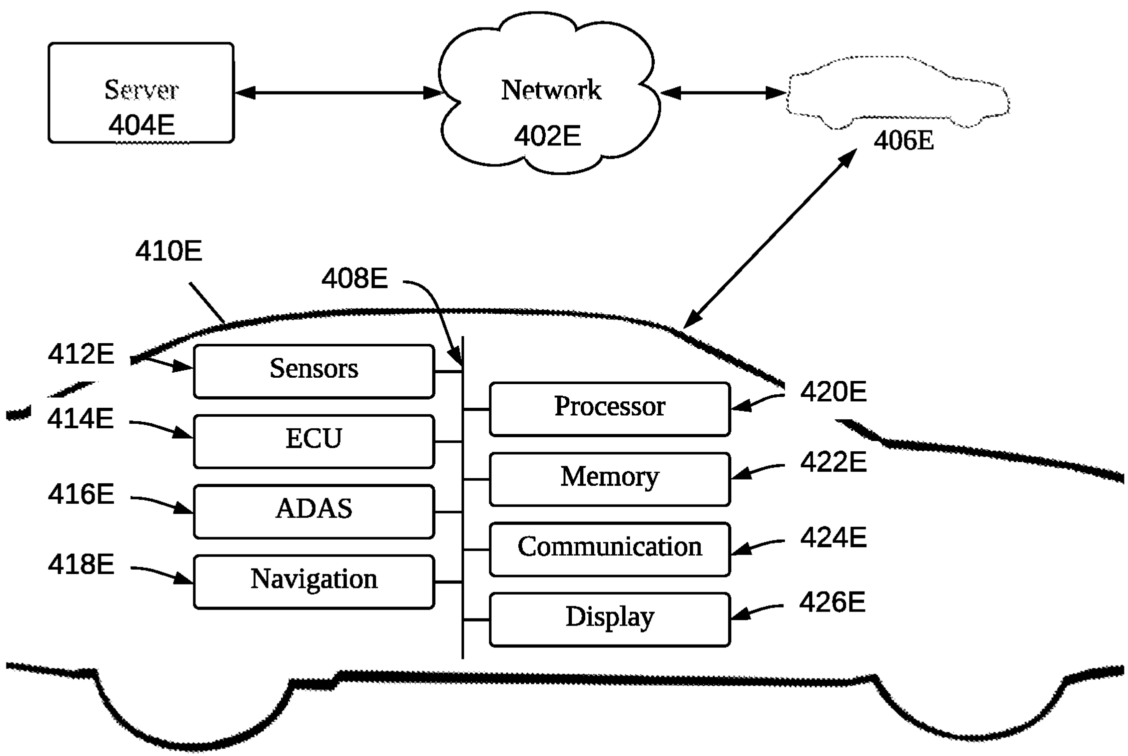

Server
404E
Network
402E
406E
410E
408E
412E
Sensors
414E
ECU
416E
ADAS
418E
Navigation
Processor
420E
Memory
422E
Communication
424E
Display
426E

FIG. 4D

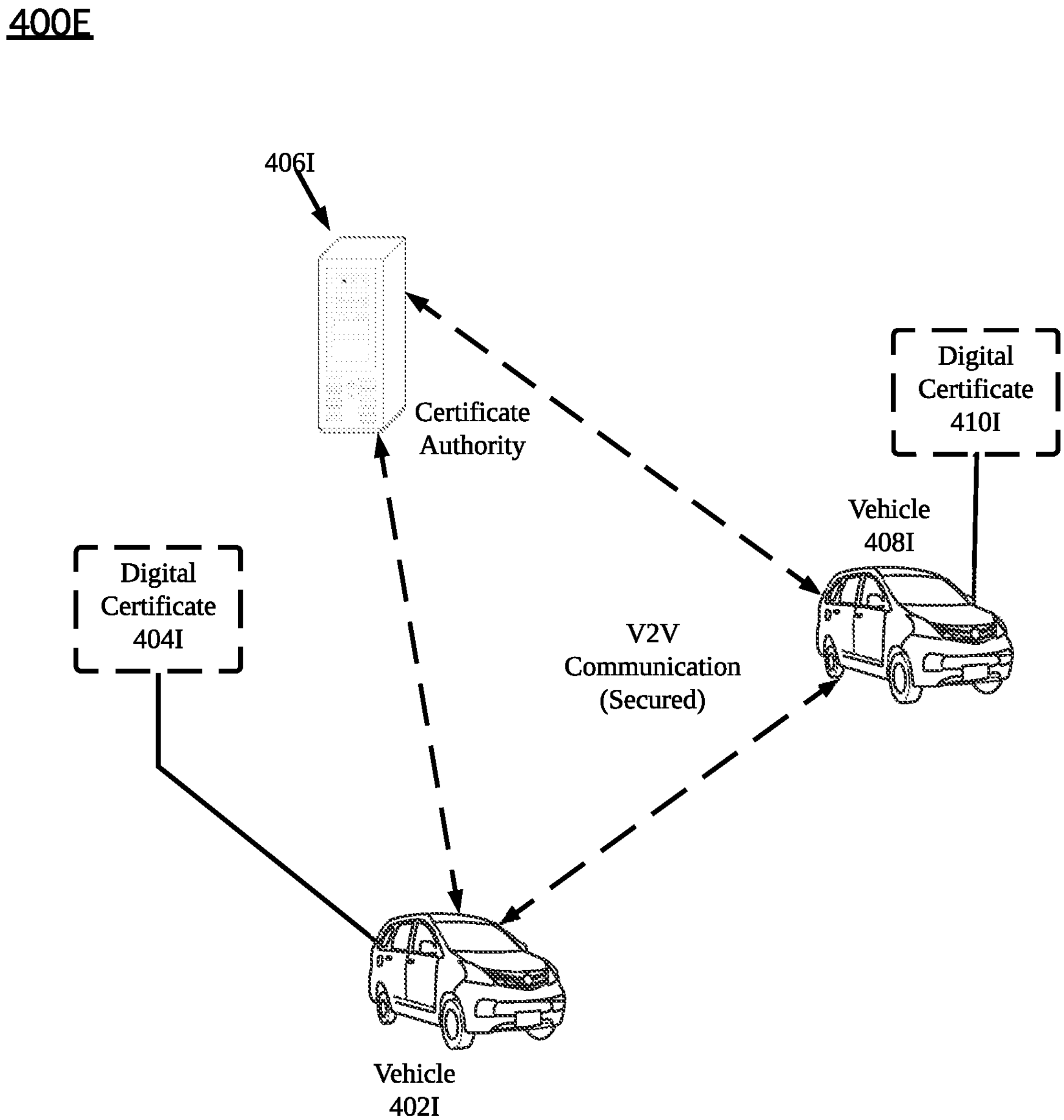

400E
406I
Certificate
Authority
Digital
Certificate
410I
Vehicle
408I
Digital
Certificate
404I
V2V
Communication
(Secured)
Vehicle
402I

FIG. 4E

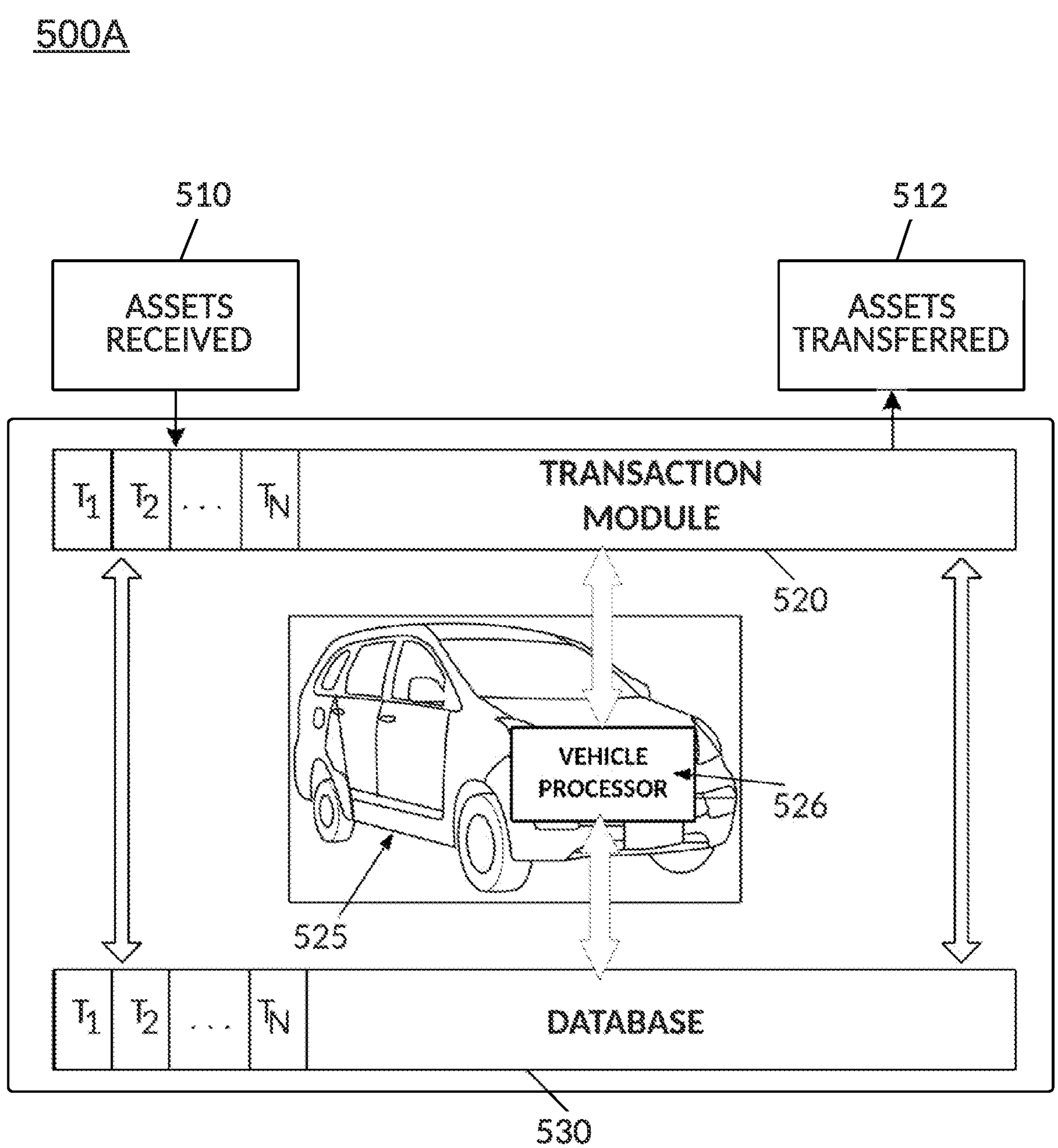

500A
510
ASSETS RECEIVED
512
ASSETS TRANSFERRED
T1
T2
...
TN
TRANSACTION MODULE
520
VEHICLE PROCESSOR
526
525
T1
T2
...
TN
DATABASE
530

FIG. 5A

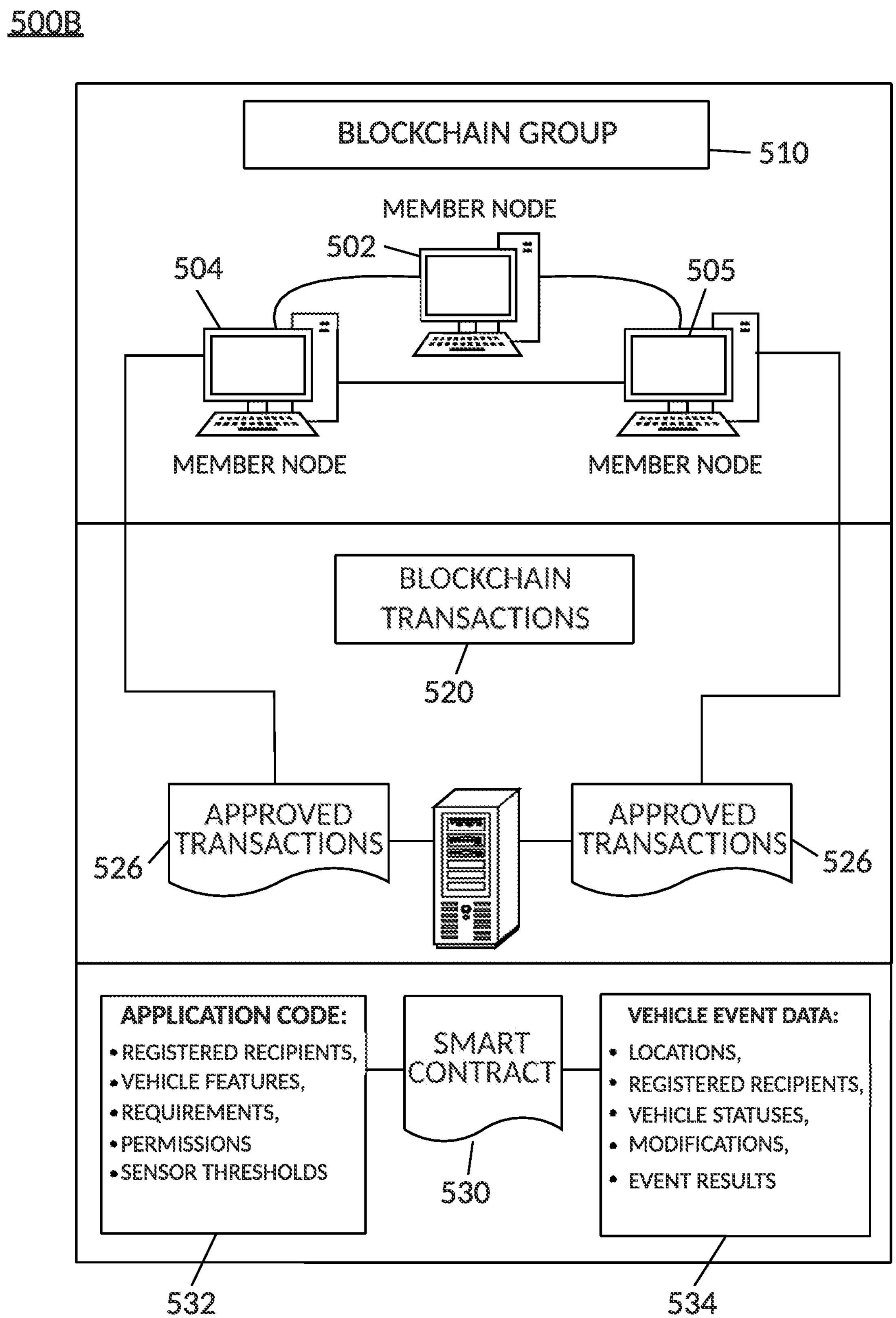

500B
BLOCKCHAIN GROUP
510
MEMBER NODE
502
504
505
MEMBER NODE
MEMBER NODE
BLOCKCHAIN TRANSACTIONS
520
APPROVED TRANSACTIONS
526
APPROVED TRANSACTIONS
526
APPLICATION CODE:
• REGISTERED RECIPIENTS,
• VEHICLE FEATURES,
• REQUIREMENTS,
• PERMISSIONS
• SENSOR THRESHOLDS
SMART CONTRACT
530
VEHICLE EVENT DATA:
• LOCATIONS,
• REGISTERED RECIPIENTS,
• VEHICLE STATUSES,
• MODIFICATIONS,
• EVENT RESULTS
532
534

FIG. 5B

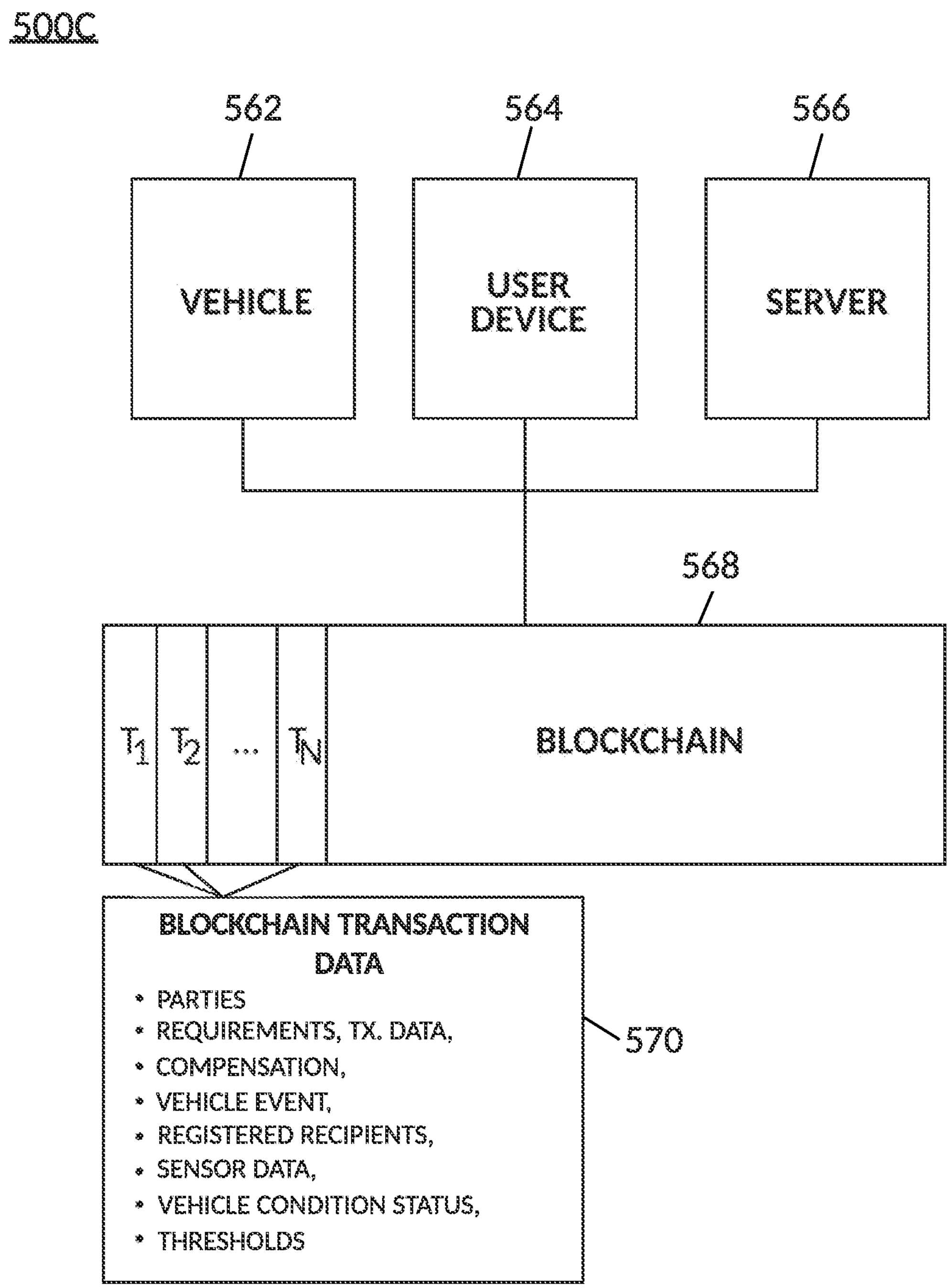

500C
562
564
566
VEHICLE
USER DEVICE
SERVER
568
T_1
T_2
...
T_N
BLOCKCHAIN
BLOCKCHAIN TRANSACTION DATA
• PARTIES
• REQUIREMENTS, TX. DATA,
• COMPENSATION,
• VEHICLE EVENT,
• REGISTERED RECIPIENTS,
• SENSOR DATA,
• VEHICLE CONDITION STATUS,
• THRESHOLDS
570

FIG. 5C

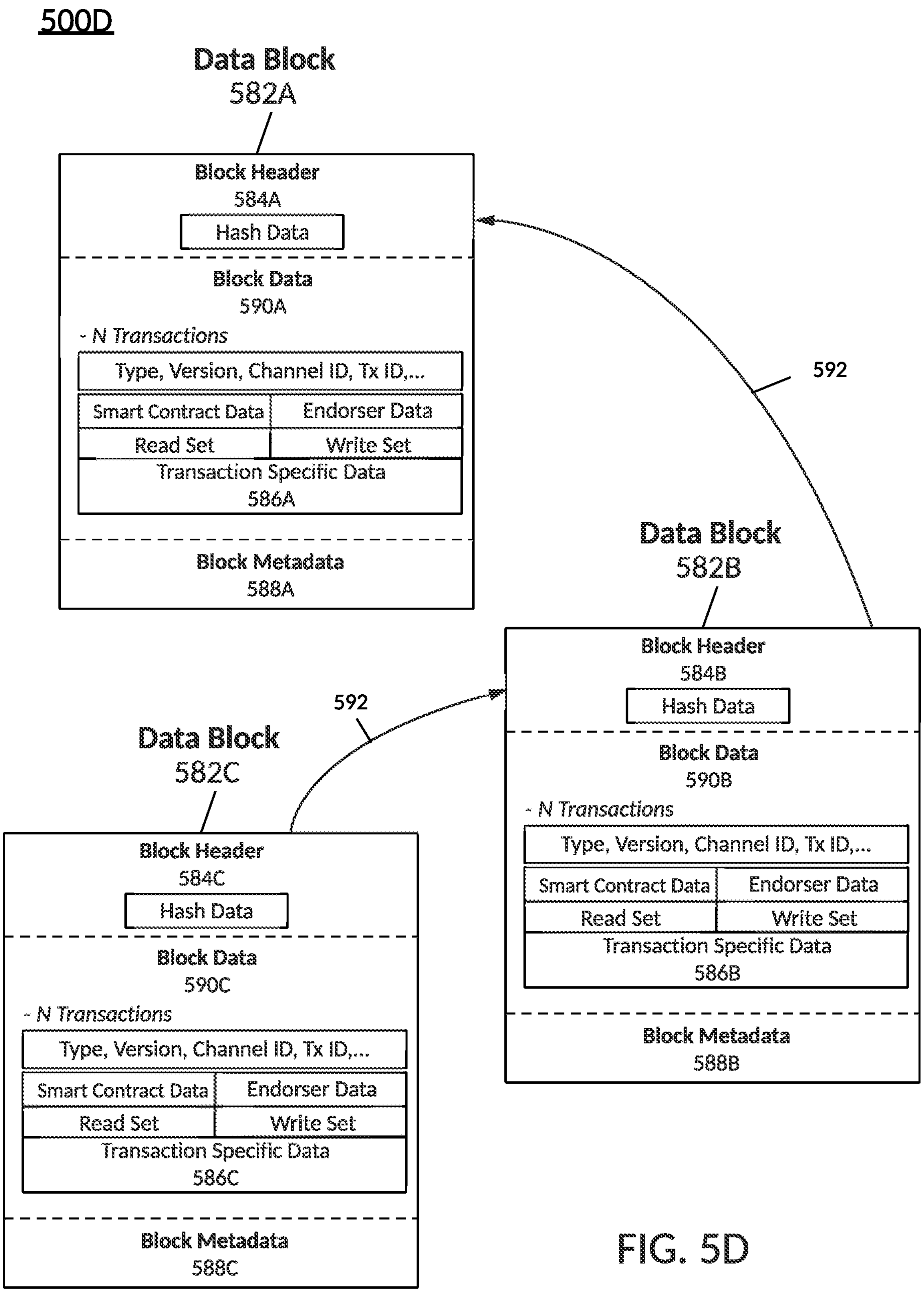

500D
Data Block
582A
Block Header
584A
Hash Data
Block Data
590A
- N Transactions
Type, Version, Channel ID, Tx ID,...
Smart Contract Data
Endorser Data
Read Set
Write Set
Transaction Specific Data
586A
Block Metadata
588A
592
Data Block
582B
Block Header
584B
Hash Data
Block Data
590B
- N Transactions
Type, Version, Channel ID, Tx ID,...
Smart Contract Data
Endorser Data
Read Set
Write Set
Transaction Specific Data
586B
Block Metadata
588B
592
Data Block
582C
Block Header
584C
Hash Data
Block Data
590C
- N Transactions
Type, Version, Channel ID, Tx ID,...
Smart Contract Data
Endorser Data
Read Set
Write Set
Transaction Specific Data
586C
Block Metadata
588C

FIG. 5D

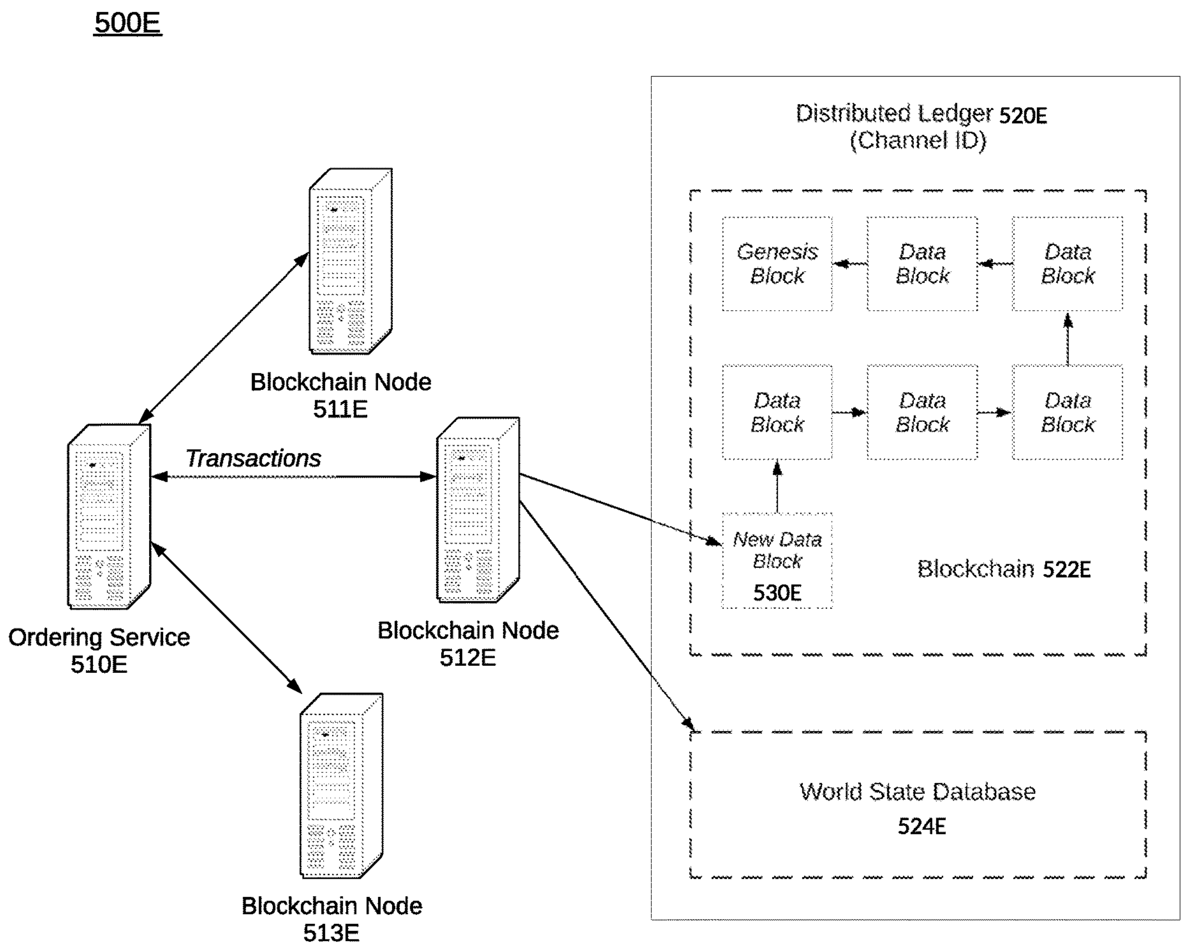

500E
Distributed Ledger 520E
(Channel ID)
Genesis Block
Data Block
Data Block
Data Block
Data Block
Data Block
New Data Block
530E
Blockchain 522E
World State Database
524E
Blockchain Node
511E
Transactions
Ordering Service
510E
Blockchain Node
512E
Blockchain Node
513E

FIG. 5E

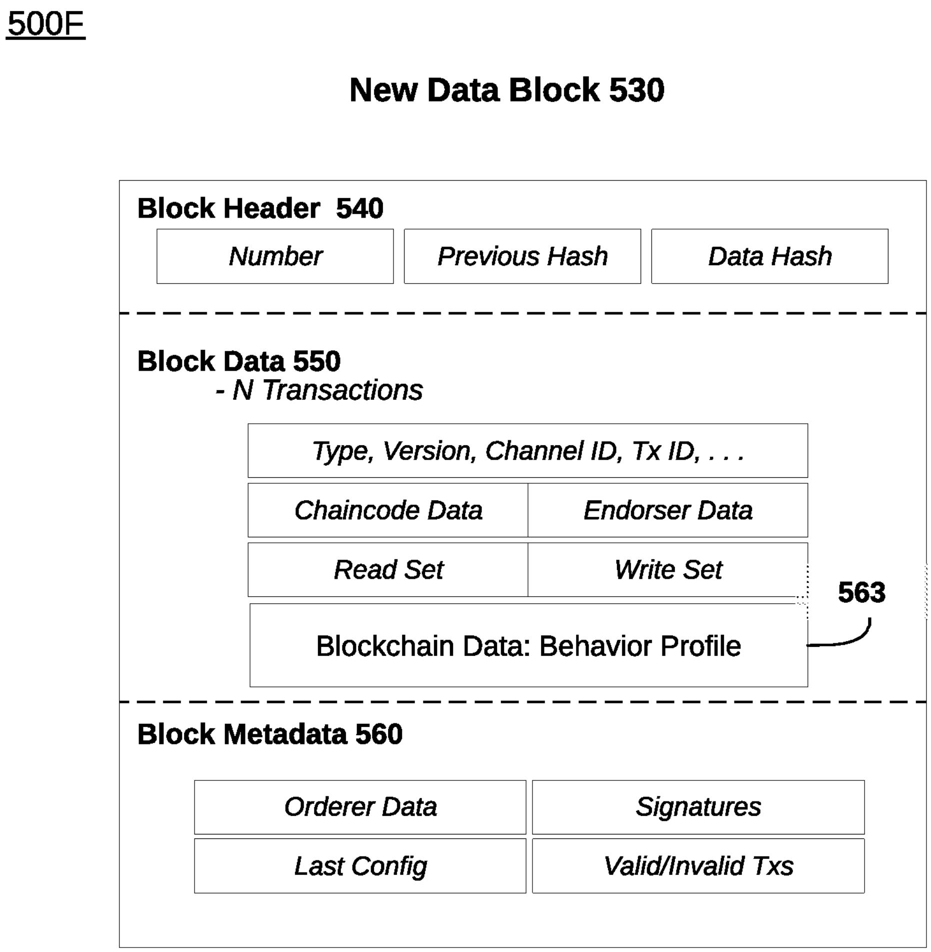

500F
New Data Block 530
Block Header 540
Number
Previous Hash
Data Hash
Block Data 550
- N Transactions
Type, Version, Channel ID, Tx ID, . . .
Chaincode Data
Endorser Data
Read Set
Write Set
563
Blockchain Data: Behavior Profile
Block Metadata 560
Orderer Data
Signatures
Last Config
Valid/Invalid Txs

FIG. 5F

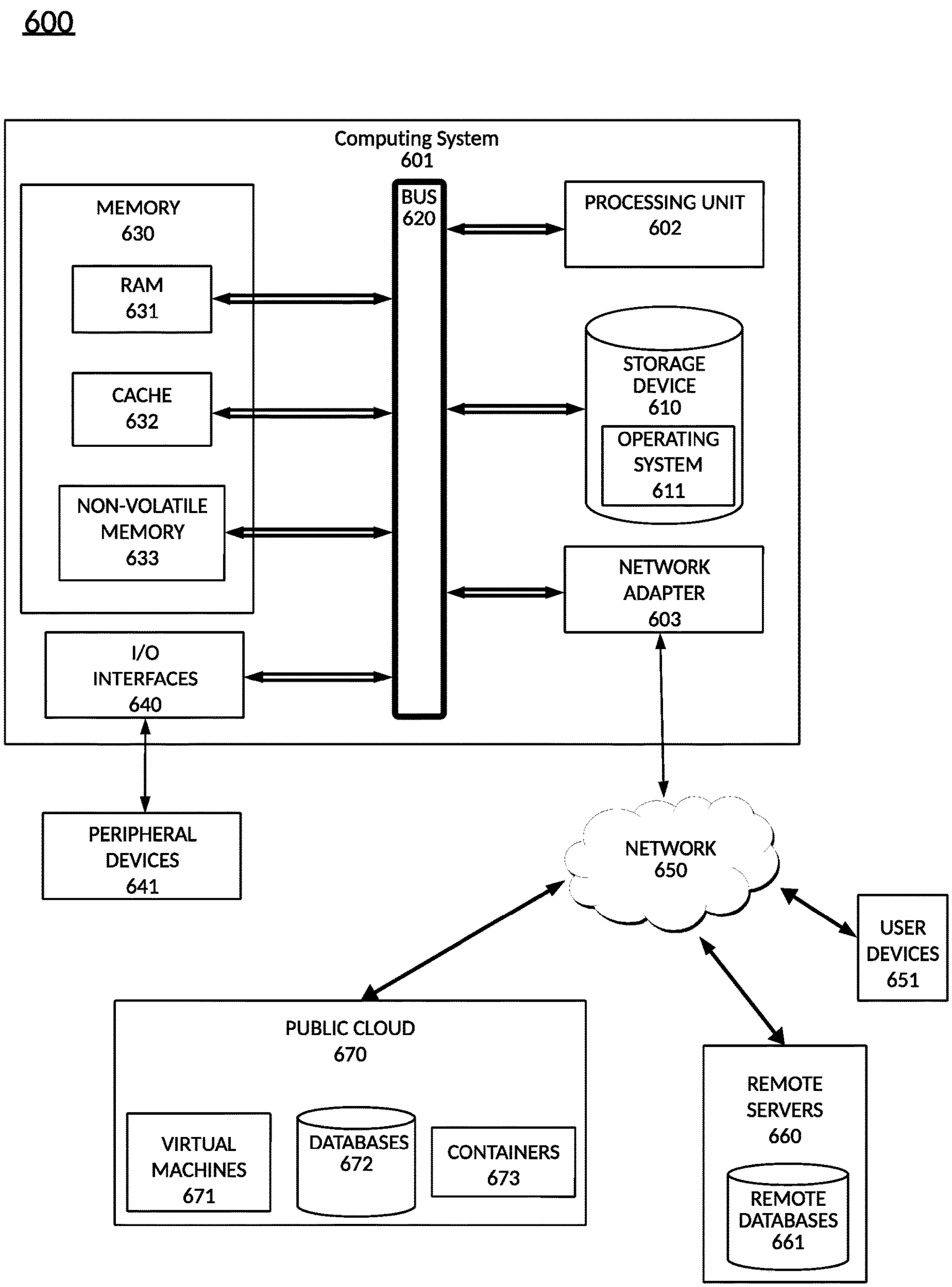

600
Computing System
601
MEMORY
630
RAM
631
CACHE
632
NON-VOLATILE
MEMORY
633
BUS
620
PROCESSING UNIT
602
STORAGE
DEVICE
610
OPERATING
SYSTEM
611
NETWORK
ADAPTER
603
I/O
INTERFACES
640
PERIPHERAL
DEVICES
641
NETWORK
650
USER
DEVICES
651
PUBLIC CLOUD
670
VIRTUAL
MACHINES
671
DATABASES
672
CONTAINERS
673
REMOTE
SERVERS
660
REMOTE
DATABASES
661

FIG. 6

PREDICTING EFFICIENCY OF PROVIDED ELECTRICITY DURING AN OUTAGE

BACKGROUND

[0001] Vehicles or transports, such as cars, motorcycles, trucks, planes, trains, etc., generally provide transportation needs to occupants and/or goods in a variety of ways. Functions related to vehicles may be identified and utilized by various computing devices, such as a smartphone or a computer located on and/or off the vehicle.

SUMMARY

[0002] One example embodiment provides a method that includes one or more of predicting, by a vehicle, a duration and a severity of an electrical-related event, determining, by the vehicle, a behavior profile of a group of devices at a location, based on a need of at least one of the group of devices during the event, and an amount of energy consumption of at least one of the group of devices during the event, predicting, by the vehicle, an energy requirement during the event at the location, based on the predicted duration, the predicted severity and the determined behavior profile, directing, by the vehicle, energy to be stored prior to the event by both the vehicle and an energy storage unit (ESU) associated with the location, and directing, by the vehicle, a flow of energy to the group of devices during the event, from the ESU and the vehicle.

[0003] Another example embodiment provides a system that includes a memory communicably coupled to a processor, wherein the processor performs one or more of predicts, at a vehicle, a duration and a severity of an electrical-related event, determines, at the vehicle, a behavior profile of a group of devices at a location, based on a need of at least one of the group of devices contemporaneously with the event, and an amount of energy consumption of at least one of the group of devices contemporaneously with the event, predicts, at the vehicle, an energy requirement contemporaneously with the event at the location, based on the predicted duration, the predicted severity and the determined behavior profile, directs energy to be stored prior to the event by both the vehicle and an energy storage unit (ESU) associated with the location, and directs a flow of energy to the group of devices contemporaneously with the event, from the ESU and the vehicle.

[0004] A further example embodiment provides a computer readable storage medium comprising instructions, that when read by a processor, cause the processor to perform one or more of predicting, by a vehicle, a duration and a severity of an electrical-related event, determining, by the vehicle, a behavior profile of a group of devices at a location, based on a need of at least one of the group of devices during the event, and an amount of energy consumption of at least one of the group of devices during the event, predicting, by the vehicle, an energy requirement during the event at the location, based on the predicted duration, the predicted severity and the determined behavior profile, directing, by the vehicle, energy to be stored prior to the event by both the vehicle and an energy storage unit (ESU) associated with the location, and directing, by the vehicle, a flow of energy to the group of devices during the event, from the ESU and the vehicle.

BRIEF DESCRIPTION OF THE DRAWINGS

[0005] FIG. **1**A illustrates an example of a system diagram, according to example embodiments.

[0006] FIG. **1**B illustrates a further example of a system diagram, according to example embodiments.

[0007] FIG. **2**A illustrates a vehicle network diagram, according to example embodiments.

[0008] FIG. **2**B illustrates another vehicle network diagram, according to example embodiments.

[0009] FIG. **2**C illustrates yet another vehicle network diagram, according to example embodiments.

[0010] FIG. **2**D illustrates a further vehicle network diagram, according to example embodiments.

[0011] FIG. **2**E illustrates a flow diagram, according to example embodiments.

[0012] FIG. **2**F illustrates another flow diagram, according to example embodiments.

[0013] FIG. **3**A illustrates an Artificial Intelligence (AI)/Machine Learning (ML) network diagram for integrating an artificial intelligence (AI) model into any decision point in the example embodiments.

[0014] FIG. **3**B illustrates a process for developing an Artificial Intelligence (AI)/Machine Learning (ML) model that supports AI-assisted vehicle or occupant decision points.

[0015] FIG. **3**C illustrates a process for utilizing an Artificial Intelligence (AI)/Machine Learning (ML) model that supports AI-assisted vehicle or occupant decision points.

[0016] FIG. **3**D illustrates a machine learning network diagram, according to example embodiments.

[0017] FIG. **3**E illustrates a machine learning network diagram, according to example embodiments.

[0018] FIG. **4**A illustrates a diagram depicting electrification of one or more elements, according to example embodiments.

[0019] FIG. **4**B illustrates a diagram depicting interconnections between different elements, according to example embodiments.

[0020] FIG. **4**C illustrates a further diagram depicting interconnections between different elements, according to example embodiments.

[0021] FIG. **4**D illustrates yet a further diagram depicting interconnections between elements, according to example embodiments.

[0022] FIG. **4**E illustrates yet a further diagram depicting an example of vehicles performing secured Vehicle-to-Vehicle (V2V) communications using security certificates, according to example embodiments.

[0023] FIG. **5**A illustrates an example vehicle configuration for managing database transactions associated with a vehicle, according to example embodiments.

[0024] FIG. **5**B illustrates an example blockchain group, according to example embodiments.

[0025] FIG. **5**C illustrates an example interaction between elements and a blockchain, according to example embodiments.

[0026] FIG. **5**D illustrates an example data block interaction, according to example embodiments.

[0027] FIG. **5**E illustrates a blockchain network diagram, according to example embodiments.

[0028] FIG. **5**F illustrates an example new data block, according to example embodiments.

[0029] FIG. **6** illustrates an example system that supports one or more of the example embodiments.

DETAILED DESCRIPTION

[0030] It will be readily understood that the instant components, as generally described and illustrated in the figures herein, may be arranged and designed in a wide variety of different configurations. Thus, the following detailed description of the embodiments of at least one of a method, apparatus, computer readable storage medium and system, as represented in the attached figures, is not intended to limit the scope of the application as claimed but is merely representative of selected embodiments. Multiple embodiments depicted herein are not intended to limit the scope of the solution. The computer-readable storage medium may be a non-transitory computer readable medium or a non-transitory computer readable storage medium.

[0031] Communications between the vehicle(s) and certain entities, such as remote servers, other vehicles and local computing devices (e.g., smartphones, personal computers, vehicle-embedded computers, etc.) may be sent and/or received and processed by one or more 'components' which may be hardware, firmware, software, or a combination thereof. The components may be part of any of these entities or computing devices or certain other computing devices. In one example, consensus decisions related to blockchain transactions may be performed by one or more computing devices or components (which may be any element described and/or depicted herein) associated with the vehicle (s) and one or more of the components outside or at a remote location from the vehicle(s).

[0032] The instant features, structures, or characteristics described in this specification may be combined in any suitable manner in one or more embodiments. For example, the usage of the phrases "example embodiments," "some embodiments,", "a first embodiment", or other similar language throughout this specification refers to the fact that a particular feature, structure, or characteristic described in connection with the one or more embodiments may be included in one or more other embodiments described or depicted herein. Thus, the one or more embodiments, described or depicted throughout this specification can all refer to the same embodiment. Thus, these embodiments may work in conjunction with any of the other embodiments, may not be functionally separate, and the described features, structures, or characteristics may be combined in any suitable manner in one or more embodiments. Although described in a particular manner, by example only, or more feature(s), element(s), and step(s) described herein may be utilized together and in various combinations, without exclusivity, unless expressly indicated otherwise herein. In the figures, any connection between elements can permit one-way and/or two-way communication, even if the depicted connection is a one-way or two-way connection, such as an arrow.

[0033] In the instant solution, a vehicle may include one or more of cars, trucks, Internal Combustion Engine (ICE) vehicles, battery electric vehicle (BEV), fuel cell vehicles, any vehicle utilizing renewable sources, hybrid vehicles, e-Palettes, buses, motorcycles, scooters, bicycles, boats, recreational vehicles, planes, drones, Unmanned Aerial Vehicle (UAV) and any object that may be used to transport people and/or goods from one location to another.

[0034] In addition, while the term "message" may have been used in the description of embodiments, other types of network data, such as, a packet, frame, datagram, etc. may also be used. Furthermore, while certain types of messages and signaling may be depicted in exemplary embodiments they are not limited to a certain type of message and signaling.

[0035] Example embodiments provide methods, systems, components, non-transitory computer readable medium, devices, and/or networks, which provide at least one of a transport (also referred to as a vehicle or car herein), a data collection system, a data monitoring system, a verification system, an authorization system, and a vehicle data distribution system. The vehicle status condition data received in the form of communication messages, such as wireless data network communications and/or wired communication messages, may be processed to identify vehicle status conditions and provide feedback on the condition and/or changes of a vehicle. In one example, a user profile may be applied to a particular vehicle to authorize a current vehicle event, service stops at service stations, to authorize subsequent vehicle rental services, and enable vehicle-to-vehicle communications.

[0036] Within the communication infrastructure, a decentralized database is a distributed storage system which includes multiple nodes that communicate with each other. A blockchain is an example of a decentralized database, which includes an append-only immutable data structure (i.e., a distributed ledger) capable of maintaining records between untrusted parties. The untrusted parties are referred to herein as peers, nodes, or peer nodes. Each peer maintains a copy of the database records, and no single peer can modify the database records without a consensus being reached among the distributed peers. For example, the peers may execute a consensus protocol to validate blockchain storage entries, group the storage entries into blocks, and build a hash chain via the blocks. This process forms the ledger by ordering the storage entries, as is necessary, for consistency. In public or permissionless blockchains, anyone can participate without a specific identity. Public blockchains can involve crypto-currencies and use consensus-based on various protocols such as proof of work (PoW). Conversely, a permissioned blockchain database can secure interactions among a group of entities, which share a common goal, but which do not or cannot fully trust one another, such as businesses that exchange funds, goods, information, and the like. The instant solution can function in a permissioned and/or a permissionless blockchain setting.

[0037] Smart contracts are trusted distributed applications which leverage tamper-proof properties of the shared or distributed ledger (which may be in the form of a blockchain) and an underlying agreement between member nodes, which is referred to as an endorsement or endorsement policy. In general, blockchain entries are "endorsed" before being committed to the blockchain while entries which are not endorsed are disregarded. A typical endorsement policy allows smart contract executable code to specify endorsers for an entry in the form of a set of peer nodes that are necessary for endorsement. When a client sends the entry to the peers specified in the endorsement policy, the entry is executed to validate the entry. After validation, the entries enter an ordering phase in which a consensus protocol produces an ordered sequence of endorsed entries grouped into blocks.

[0038] Nodes are the communication entities of the blockchain system. A "node" may perform a logical function in the sense that multiple nodes of different types can run on the same physical server. Nodes are grouped in trust domains and are associated with logical entities that control them in various ways. Nodes may include different types, such as a client or submitting-client node, which submits an entry-invocation to an endorser (e.g., peer), and broadcasts entry proposals to an ordering service (e.g., ordering node). Another type of node is a peer node, which can receive client submitted entries, commit the entries, and maintain a state and a copy of the ledger of blockchain entries. Peers can also have the role of an endorser. An ordering-service-node or orderer is a node running the communication service for all nodes and which implements a delivery guarantee, such as a broadcast to each of the peer nodes in the system when committing entries and modifying a world state of the blockchain. The world state can constitute the initial blockchain entry, which normally includes control and setup information.

[0039] A ledger is a sequenced, tamper-resistant record of all state transitions of a blockchain. State transitions may result from smart contract executable code invocations (i.e., entries) submitted by participating parties (e.g., client nodes, ordering nodes, endorser nodes, peer nodes, etc.). An entry may result in a set of asset key-value pairs being committed to the ledger as one or more operands, such as creates, updates, deletes, and the like. The ledger includes a blockchain (also referred to as a chain), which stores an immutable, sequenced record in blocks. The ledger also includes a state database, which maintains a current state of the blockchain. There is typically one ledger per channel. Each peer node maintains a copy of the ledger for each channel of which they are a member.

[0040] A chain is an entry log structured as hash-linked blocks, and each block contains a sequence of N entries where N is equal to or greater than one. The block header includes a hash of the blocks' entries, as well as a hash of the prior block's header. In this way, all entries on the ledger may be sequenced and cryptographically linked together. Accordingly, it is not possible to tamper with the ledger data without breaking the hash links. A hash of a most recently added blockchain block represents every entry on the chain that has come before it, making it possible to ensure that all peer nodes are in a consistent and trusted state. The chain may be stored on a peer node file system (i.e., local, attached storage, cloud, etc.), efficiently supporting the append-only nature of the blockchain workload.

[0041] The current state of the immutable ledger represents the latest values for all keys that are included in the chain entry log. Since the current state represents the latest key values known to a channel, it is sometimes referred to as a world state. Smart contract executable code invocations execute entries against the current state data of the ledger. To make these smart contract executable code interactions efficient, the latest values of the keys may be stored in a state database. The state database may be simply an indexed view into the chain's entry log and can therefore be regenerated from the chain at any time. The state database may automatically be recovered (or generated if needed) upon peer node startup and before entries are accepted.

[0042] A blockchain is different from a traditional database in that the blockchain is not a central storage but rather a decentralized, immutable, and secure storage, where nodes must share in changes to records in the storage. Some properties that are inherent in blockchain and which help implement the blockchain include, but are not limited to, an immutable ledger, smart contracts, security, privacy, decentralization, consensus, endorsement, accessibility, and the like.

[0043] Example embodiments provide a service to a particular vehicle and/or a user profile that is applied to the vehicle. For example, a user may be the owner of a vehicle or the operator of a vehicle owned by another party. The vehicle may require service at certain intervals, and the service needs may require authorization before permitting the services to be received. Also, service centers may offer services to vehicles in a nearby area based on the vehicle's current route plan and a relative level of service requirements (e.g., immediate, severe, intermediate, minor, etc.). The vehicle needs may be monitored via one or more vehicle and/or road sensors or cameras, which report sensed data to a central controller computer device in and/or apart from the vehicle. This data is forwarded to a management server for review and action. A sensor may be located on one or more of the interior of the vehicle, the exterior of the vehicle, on a fixed object apart from the vehicle, and on another vehicle proximate the vehicle. The sensor may also be associated with the vehicle's speed, the vehicle's braking, the vehicle's acceleration, fuel levels, service needs, the gear-shifting of the vehicle, the vehicle's steering, and the like. A sensor, as described herein, may also be a device, such as a wireless device in and/or proximate to the vehicle. Also, sensor information may be used to identify whether the vehicle is operating safely and whether an occupant has engaged in any unexpected vehicle conditions, such as during a vehicle access and/or utilization period. Vehicle information collected before, during and/or after a vehicle's operation may be identified and stored in a transaction on a shared/distributed ledger, which may be generated and committed to the immutable ledger as determined by a permission granting consortium, and thus in a "decentralized" manner, such as via a blockchain membership group.

[0044] Each interested party (i.e., owner, user, company, agency, etc.) may want to limit the exposure of private information, and therefore the blockchain and its immutability can be used to manage permissions for each particular user vehicle profile. A smart contract may be used to provide compensation, quantify a user profile score/rating/review, apply vehicle event permissions, determine when service is needed, identify a collision and/or degradation event, identify a safety concern event, identify parties to the event and provide distribution to registered entities seeking access to such vehicle event data. Also, the results may be identified, and the necessary information can be shared among the registered companies and/or individuals based on a consensus approach associated with the blockchain. Such an approach may not be implemented on a traditional centralized database.

[0045] Various driving systems of the instant solution can utilize software, an array of sensors as well as machine learning functionality, light detection and ranging (Lidar) projectors, radar, ultrasonic sensors, etc. to create a map of terrain and road that a vehicle can use for navigation and other purposes. In some embodiments, GPS, maps, cameras, sensors, and the like can also be used in autonomous vehicles in place of Lidar.

[0046] The instant solution includes, in certain embodiments, authorizing a vehicle for service via an automated and quick authentication scheme. For example, driving up to a charging station or fuel pump may be performed by a vehicle operator or an autonomous vehicle and the authorization to receive charge or fuel may be performed without any delays provided the authorization is received by the service and/or charging station. A vehicle may provide a communication signal that provides an identification of a vehicle that has a currently active profile linked to an account that is authorized to accept a service, which can be later rectified by compensation. Additional measures may be used to provide further authentication, such as another identifier may be sent from the user's device wirelessly to the service center to replace or supplement the first authorization effort between the vehicle and the service center with an additional authorization effort.

[0047] Data shared and received may be stored in a database, which maintains data in one single database (e.g., database server) and generally at one particular location. This location is often a central computer, for example, a desktop central processing unit (CPU), a server CPU, or a mainframe computer. Information stored on a centralized database is typically accessible from multiple different points. A centralized database is easy to manage, maintain, and control, especially for purposes of security because of its single location. Within a centralized database, data redundancy is minimized as a single storing place of all data also implies that a given set of data only has one primary record. A blockchain may be used for storing vehicle-related data and transactions.

[0048] Any of the actions described herein may be performed by one or more processors (such as a microprocessor, a sensor, an Electronic Control Unit (ECU), a head unit, and the like), with or without memory, which may be located on-board the vehicle and/or off-board the vehicle (such as a server, computer, mobile/wireless device, etc.). The one or more processors may communicate with other memory and/or other processors on-board or off-board other vehicles to utilize data being sent by and/or to the vehicle. The one or more processors and the other processors can send data, receive data, and utilize this data to perform one or more of the actions described or depicted herein.

[0049] FIG. **1**A illustrates an example of a system diagram **100**, according to example embodiments. In some embodiments, the instant solution fully or partially executes in a memory **105** of a server **103**, in a memory **110** of a processor **111** associated with a vehicle **102**, in a memory **125** of a processor **124** associated with an energy storage unit **107**, in a memory **138** of a home controller **139** associated with a location **108**, or in a memory of one or more other processors associated with devices and/or entities mentioned herein. In some embodiments, one or more of the server **103**, the processor **111**, the processor **124**, or the controller **139** may include a microcontroller that contains one or more central processing unit (CPU) cores, along with program memory and programmable input/output peripherals. Program memory can be provided, for example, in the form of flash memory.

[0050] In some embodiments, the processor **111** predicts a duration and a severity of an electrical-related event. For example, the processor **111** may communicate with a grid server **142** over a network **104** to obtain information about a current and/or projected outage status of an electrical grid **143**. Alternatively, or additionally, the processor **111** may communicate with a weather conditions server **145** over the network **104** to obtain information about one or more current or upcoming weather conditions that may impact the electrical grid **143**. The processor **111** may use the outage status information and/or the weather condition information to predict the duration and the severity of the electrical-related event.

[0051] In some embodiments, the processor **111** determines a behavior profile of a group of devices **109** at a location **108**, based on a need of at least one of the group of devices **109** during the event, and an amount of energy consumption of at least one of the group of devices **109** during the event. For example, the group of devices **109** may include a first device **131** that comprises a heating, ventilation and air conditioning system. The group of devices **109** may also include a second device **132** that comprises a light fixture. During the event, the location **108** may have a need for air conditioning and/or heat to be supplied by the first device **131**. Likewise, during the event, the location **108** may have a need for light to be supplied by the second device **132**. The microcontroller **116** of the smart panel **106** may monitor a current consumption of the first device **131** and a current consumption of the second device **132**, to determine how much energy the first and second devices **132** will consume during the predicted event. The microcontroller **116** may transmit the current consumption of the first and second devices **131**, **132** and the need for the first and second devices **131**, **132** during the predicted event, to the processor **111** over the network **104**. The processor **111** may use the current consumption of the first and second devices **131**, **132** and the need for the first and second devices **131**, **132** to create the behavior profile for the group of devices **109**, and to store the behavior profile in memory **110**.

[0052] In some embodiments, the processor **111** predicts an energy requirement during the event at the location **108**, based on the predicted duration of the event, the predicted severity of the event, and the determined behavior profile. For example, the predicted duration of the event may be six hours, the predicted severity of the event may be a total electrical power outage from downed power lines due to an ice storm, and the determined behavior profile may indicate that heat and light may be needed at the location **108** throughout the event, with heat being supplied by the first device **131** and light being supplied by the second device **132**. Based on the predicted duration, the predicted severity, and the determined behavior profile, the processor **111** may predict that the first device **131** and the second device **132** will consume a certain number of kilowatt-hours of electrical energy during the event.

[0053] In some embodiments, the processor **111** directs energy to be stored prior to the event by both the vehicle **102** and the energy storage unit (ESU) **107** associated with the location **108**. For example, the ESU **107** may include a battery bank **123** operatively coupled to a battery management system **122**. The battery management system **122** may be operatively coupled to the smart panel **106** that receives electrical energy from the electrical grid **143**. Likewise, the vehicle **102** may include a battery **113** coupled to a battery management system **112**. The battery management system **112** is connected to a charging connector **127**. The charging connector **127** is configured for connecting to a charging station **128**, wherein the charging station **128** is connected to the electrical grid **143**. The processor **111** may direct the vehicle **102** to store a first amount of energy in the battery **113**, and the ESU **107** to store a second amount of energy in the battery bank **123**, wherein the sum of the first and second amounts is sufficient for supplying the location **108** with at least the number of kilowatt-hours of electrical energy that is needed during the event.

[0054] In some embodiments, the processor **111** directs a flow of energy to the group of devices **109**, from the battery bank **123** of the ESU **107** and from the battery **113** of the vehicle **102**. The battery **113** may be connected to an inverter **114** for converting a direct current (DC) from the battery **113** to an alternating current (AC). The AC from the inverter **114** can be supplied to a Vehicle-to-Load (V2L) connector **115** at the vehicle **102**. The V2L connector **115** may be connected to the smart panel **106** for supplying electrical power to the group of devices **109**. Likewise, the battery bank **123** of the ESU **107** may be connected to an inverter **129** for converting DC from the battery bank **123** to AC for supplying the smart panel **106**, with the smart panel **106** supplying electrical power to the group of devices **109**.

[0055] In some embodiments, the processor **111** directs the flow of energy to the group of devices **109** during the event by first directing energy from the battery **113** of the vehicle **102** to the group of devices **109** through the inverter **114**, the V2L connector **115**, and the smart panel **106**. Subsequently, the processor **111** may direct the flow of energy to the group of devices **109** during the event by directing energy from the battery bank **123** of the ESU **107** to the group of devices **109** through the inverter **129** and the smart panel **106**.

[0056] In some embodiments, the processor **111** determines a first amount of time that the ESU **107** provides energy to the group of devices **109**, based on the determined behavior profile. For example, when the ESU **107** provides energy to the group of devices **109** for the first amount of time, a state-of-charge of the battery bank **123** will have been depleted by a certain amount. In some embodiments, the processor **111** determines an additional charge to be added to the battery **113** of the vehicle **102**, based on the first amount of time. For example, the additional charge to be added to the battery **113** may be equal to or greater than the amount of the depletion of the state-of-charge of the battery bank **123**. In a further embodiment, the processor **111** determines a second amount of time for the vehicle **102** to leave the location to receive the determined additional charge, return to the location **108**, and connect to the group of devices **109**. The processor **111** may switch the directed energy from the battery **113** of the vehicle **102** to the battery bank **123** of the ESU **107** when the first amount of time is less than the second amount of time.

[0057] In some embodiments, the processor **111** gathers information specifying an amount of energy consumption for the group of devices **109** over a period of time. For example, the processor **111** may gather the amount of energy consumption over the network **104** from the microcontroller **116** of the smart panel. Alternatively, or additionally, the processor **111** may gather the amount of energy consumption over the network **104** from the home controller **139**. The gathered amount of energy consumption may be stored in the behavior profile.

[0058] In some embodiments, the processor **111** directs the flow of energy to the group of devices **109** during the event by directing energy from the battery **113** of the vehicle **102** to the group of devices **109**. The processor **111** may monitor a state-of-charge of the battery **113**. In response to the state-of-charge of the battery **113** dropping below a threshold, the processor **111** may direct energy from the battery bank **123** of the ESU **107** to the group of devices **109**.

[0059] In some embodiments, the processor **111** determines that at least one device of the group of devices **109** is a medical device associated with an occupant of the location **108**. For example, the first device **131** may be a dialysis machine, an insulin pump, a heart-lung machine, a continuous positive airway pressor (CPAP) machine, or another type of medical device. The processor **111** determines an amount of energy required to power the medical device based on the predicted duration and the predicted severity of the event. For example, the microcontroller **116** of the smart panel **106** may monitor the first device **131** over a period of time to determine an amount of current consumption for the first device **131**, and/or an amount of kilowatt-hours of power consumed by the first device **131**. The microcontroller **116** may transmit the amount of current consumption and/or the amount of kilowatt-hours over the network **104** to the processor **111**. Based on the amount of current consumption and/or the amount of kilowatt-hours, the processor **111** may determine an amount of energy needed by the first device **131** for the expected duration and severity of the event. The processor **111** may reserve at least the amount of energy needed by the first device **131** on at least one of the ESU **107** or the battery **113** of the vehicle **102**.

[0060] FIG. **1B** illustrates an example of a system diagram **150**, according to example embodiments. In some embodiments, the instant solution fully or partially executes in the memory **105** of the server **103**, in the memory **110** of the processor **111** associated with the vehicle **102**, in the memory **125** of the processor **124** associated with the energy storage unit **107**, or in a memory of one or more other processors associated with devices and/or entities mentioned herein. In some embodiments, one or more of the server **103**, the processor **111**, or the processor **124** may include a microcontroller that contains one or more central processing unit (CPU) cores, along with program memory and programmable input/output peripherals. Program memory can be provided, for example, in the form of flash memory.

[0061] In some embodiments, the processor **111** of the vehicle **102** directs a flow of energy to the group of devices **109** during the event by directing energy from the battery **113** of the vehicle **102** to the group of devices **109**. The processor **111** may determine a minimum state-of-charge for the battery **113** that is required for the vehicle **102** to transport an occupant from the location **108** to another location **157** during the event. For example, the another location **157** may be a medical facility, a hospital, a residence, a charging station, a police station, a fire station, a bus station, a train station, an airport, a boat launch, a military base, a school, a church, a media outlet, or another type of location. In a further embodiment, the another location **157** includes a charging station **158**. For example, the charging station **158** may be connected to the electrical grid **143**. The processor **111** may determine the minimum state-of-charge for the battery **113** by determining a driving distance between the location **108** and the another location **157**, and calculating an amount of charge for the battery **113** that is sufficient for the vehicle **102** to reach the another location **157**. For example, the driving distance may be determined using a Global Positioning System (GPS) device operatively coupled to the processor **111**. In a further embodiment, the processor **111** may determine the minimum state-of-charge for the battery **113** based on traffic conditions between the location **108** and the another location **157**. For example, the processor **111** may communicate with a traffic conditions server **163** over the network **104** to obtain information about traffic congestion between the location **108** and the another location **157**.

[0062] In some embodiments, the processor **111** monitors a state-of-charge of the battery **113** while the battery **113** is supplying energy to the group of devices **109**. For example, the processor **111** may obtain the state-of-charge from the battery management system **112**. In response to the state-of-charge of the battery **113** approaching a minimum state-of-charge threshold, the processor **111** directs energy from the ESU **107** to the group of devices **109**, and directs the vehicle **102** to cease providing power to the group of devices **109** from the battery **113**.

[0063] Flow diagrams depicted herein, such as FIG. **2**C, FIG. **2**D, FIG. **2**E, and FIG. **2**F are separate examples but may be the same or different embodiments. Any of the operations in one flow diagram may be adopted and shared with another flow diagram. No example operation is intended to limit the subject matter of any embodiment or corresponding claim.

[0064] It is important to note that all the flow diagrams and corresponding processes derived from FIG. **2**C, FIG. **2**D, FIG. **2**E, and FIG. **2**F may be part of a same process or may share sub-processes with one another thus making the diagrams combinable into a single preferred embodiment that does not require any one specific operation but which performs certain operations from one example process and from one or more additional processes. All the example processes are related to the same physical system and can be used separately or interchangeably.

[0065] The instant solution can be used in conjunction with one or more types of vehicles: battery electric vehicles, hybrid vehicles, fuel cell vehicles, internal combustion engine vehicles and/or vehicles utilizing renewable sources.

[0066] FIG. **2**A illustrates a vehicle network diagram **200**, according to example embodiments. The network comprises elements including a vehicle **202** including a processor **204**, as well as a vehicle **202'** including a processor **204'**. The vehicles **202**, **202'** communicate with one another via the processors **204**, **204'**, as well as other elements (not shown) including transceivers, transmitters, receivers, storage, sensors, and other elements capable of providing communication. The communication between the vehicles **202**, and **202'** can occur directly, via a private and/or a public network (not shown), or via other vehicles and elements comprising one or more of a processor, memory, and software. Although depicted as single vehicles and processors, a plurality of vehicles and processors may be present. One or more of the applications, features, steps, solutions, etc., described and/or depicted herein may be utilized and/or provided by the instant elements.

[0067] FIG. **2**B illustrates another vehicle network diagram **210**, according to example embodiments. The network comprises elements including a vehicle **202** including a processor **204**, as well as a vehicle **202'** including a processor **204'**. The vehicles **202**, **202'** communicate with one another via the processors **204**, **204'**, as well as other elements (not shown), including transceivers, transmitters, receivers, storage, sensors, and other elements capable of providing communication. The communication between the vehicles **202**, and **202'** can occur directly, via a private and/or a public network (not shown), or via other vehicles and elements comprising one or more of a processor, memory, and software. The processors **204**, **204'** can further communicate with one or more elements **230** including sensor **212**, wired device **214**, wireless device **216**, database **218**, mobile phone **220**, vehicle **222**, computer **224**, input/output (I/O) device **226**, and voice application **228**. The processors **204**, **204'** can further communicate with elements comprising one or more of a processor, memory, and software.

[0068] Although depicted as single vehicles, processors and elements, a plurality of vehicles, processors and elements may be present. Information or communication can occur to and/or from any of the processors **204**, **204'** and elements **230**. For example, the mobile phone **220** may provide information to the processor **204**, which may initiate the vehicle **202** to take an action, may further provide the information or additional information to the processor **204'**, which may initiate the vehicle **202'** to take an action, may further provide the information or additional information to the mobile phone **220**, the vehicle **222**, and/or the computer **224**. One or more of the applications, features, steps, solutions, etc., described and/or depicted herein may be utilized and/or provided by the instant elements.

[0069] FIG. **2**C illustrates yet another vehicle network diagram **240**, according to example embodiments. The network comprises elements including a vehicle **202**, a processor **204**, and a non-transitory computer readable medium **242**C. The processor **204** is communicably coupled to the computer readable medium **242**C and elements **230** (which were depicted in FIG. **2**B). The vehicle **202** may be a vehicle, server, or any device with a processor and memory.

[0070] The processor **204** performs one or more of predicting, by a vehicle, a duration and a severity of an electrical-related event **244**C; determining, by the vehicle, a behavior profile of a group of devices at a location, based on a need of at least one of the group of devices during the event, and an amount of energy consumption of at least one of the group of devices during the event **246**C; predicting, by the vehicle, an energy requirement during the event at the location, based on the predicted duration, the predicted severity and the determined behavior profile **248**C; directing, by the vehicle, energy to be stored prior to the event by both the vehicle and an energy storage unit (ESU) associated with the location **250**C; and directing, by the vehicle, a flow of energy to the group of devices during the event, from the ESU and the vehicle **252**C.

[0071] FIG. **2**D illustrates a further vehicle network diagram **250**, according to example embodiments. The network comprises elements including a vehicle **202** a processor **204**, and a non-transitory computer readable medium **242**D. The processor **204** is communicably coupled to the computer readable medium **242**D and elements **230** (which were depicted in FIG. **2**B). The vehicle **202** may be a vehicle, server or any device with a processor and memory.

[0072] The processor **204** performs one or more of directing, by the vehicle, the flow of energy to the group of devices during the event by first directing energy from a battery of the vehicle to the group of devices **244**D; determining, by the vehicle, a first amount of time that the ESU provides energy to the group of devices, based on the determined behavior profile; and determining, by the vehicle, an additional charge to be added to a battery of the vehicle, based on the first amount of time **245**D; gathering the amount of energy consumption for the group of devices over a period of time; and storing the gathered amount of energy consumption in the behavior profile **246**D; wherein the directing, by the vehicle, the flow of energy to the group of devices during the event is performed by directing energy from a battery of the vehicle to the group of devices; monitoring a state-of-charge of the battery; and, in response to the state-of-charge dropping below a threshold, directing energy from the ESU to the group of devices **247**D; wherein the directing, by the vehicle, the flow of energy to the group of devices during the event is performed by directing energy from a battery of the vehicle to the group of devices; determining a minimum state-of-charge of the battery required for the vehicle to transport an occupant from the location to another location during the event; monitoring a state-of-charge of the battery; and, in response to the state-of-charge approaching the minimum state-of-charge, directing energy from the ESU to the group of devices **248**D; and determining that at least one device of the group of devices is a medical device associated with an occupant of the location; determining an amount of energy required to power the medical device based on the predicted duration and the predicted severity of the event; and reserving at least the amount of energy on at least one of the ESU or the vehicle for powering the medical device **249**D.

[0073] While this example describes in detail only one vehicle **202**, multiple such nodes may be connected to the blockchain. It should be understood that the vehicle **202** may include additional components and that some of the components described herein may be removed and/or modified without departing from a scope of the instant application. The vehicle **202** may have a computing device or a server computer, or the like, and may include a processor **204**, which may be a semiconductor-based microprocessor, a central processing unit (CPU), an application-specific integrated circuit (ASIC), a field-programmable gate array (FPGA), and/or another hardware device. Although a single processor **204** is depicted, it should be understood that the vehicle **202** may include multiple processors, multiple cores, or the like without departing from the scope of the instant application. The vehicle **202** may be a vehicle, server or any device with a processor and memory.

[0074] The processor **204** performs one or more of receiving a confirmation of an event from one or more elements described or depicted herein, wherein the confirmation comprises a blockchain consensus between peers represented by any of the elements and executing a smart contract to record the confirmation on the blockchain consensus. Consensus is formed between one or more of any element **230** and/or any element described or depicted herein, including a vehicle, a server, a wireless device, etc. In another example, the vehicle **202** can be one or more of any element **230** and/or any element described or depicted herein, including a server, a wireless device, etc.

[0075] The processors and/or computer readable medium may fully or partially reside in the interior or exterior of the vehicles. The steps or features stored in the computer readable medium may be fully or partially performed by any of the processors and/or elements in any order. Additionally, one or more steps or features may be added, omitted, combined, performed at a later time, etc.

[0076] FIG. **2**E illustrates a flow diagram **260**, according to example embodiments. Referring to FIG. **2**E, the instant solution includes one or more of predicting, by a vehicle, a duration and a severity of an electrical-related event **244**E; determining, by the vehicle, a behavior profile of a group of devices at a location, based on a need of at least one of the group of devices during the event, and an amount of energy consumption of at least one of the group of devices during the event **246**E; predicting, by the vehicle, an energy requirement during the event at the location, based on the predicted duration, the predicted severity and the determined behavior profile **248**E; directing, by the vehicle, energy to be stored prior to the event by both the vehicle and an energy storage unit (ESU) associated with the location **250**E; and directing, by the vehicle, a flow of energy to the group of devices during the event, from the ESU and the vehicle **252**E.

[0077] FIG. **2**F illustrates another flow diagram **270**, according to example embodiments. Referring to FIG. **2**F, the instant solution includes one or more of directing, by the vehicle, the flow of energy to the group of devices during the event by first directing energy from a battery of the vehicle to the group of devices **244**F; determining, by the vehicle, a first amount of time that the ESU provides energy to the group of devices, based on the determined behavior profile; and determining, by the vehicle, an additional charge to be added to a battery of the vehicle, based on the first amount of time **245**F; gathering the amount of energy consumption for the group of devices over a period of time; and storing the gathered amount of energy consumption in the behavior profile **246**F; wherein the directing, by the vehicle, the flow of energy to the group of devices during the event is performed by directing energy from a battery of the vehicle to the group of devices; monitoring a state-of-charge of the battery; and, in response to the state-of-charge dropping below a threshold, directing energy from the ESU to the group of devices **247**F; wherein the directing, by the vehicle, the flow of energy to the group of devices during the event is performed by directing energy from a battery of the vehicle to the group of devices; determining a minimum state-of-charge of the battery required for the vehicle to transport an occupant from the location to another location during the event; monitoring a state-of-charge of the battery; and, in response to the state-of-charge approaching the minimum state-of-charge, directing energy from the ESU to the group of devices **248**F; and determining that at least one device of the group of devices is a medical device associated with an occupant of the location; determining an amount of energy required to power the medical device based on the predicted duration and the predicted severity of the event; and reserving at least the amount of energy on at least one of the ESU or the vehicle for powering the medical device **249**F.

[0078] Technological advancements typically build upon the fundamentals of predecessor technologies; such is the case with Artificial Intelligence (AI) models. An AI classification system describes the stages of AI progression. The first classification is known as "Reactive Machines," followed by present-day AI classification "Limited Memory Machines" (also known as "Artificial Narrow Intelligence"), then progressing to "Theory of Mind" (also known as "Artificial General Intelligence"), and reaching the AI classification "Self-Aware" (also known as "Artificial Superintelligence"). Present-day Limited Memory Machines are a growing group of AI models built upon the foundation of its predecessor, Reactive Machines. Reactive Machines emulate human responses to stimuli; however, they are limited in their capabilities as they cannot typically learn from prior experience. Once the AI model's learning abilities emerged, its classification was promoted to Limited Memory Machines. In this present-day classification, AI models learn from large volumes of data, detect patterns, solve problems, generate and predict data, and the like, while inheriting all of the capabilities of Reactive Machines. Examples of AI models classified as Limited Memory Machines include, but are not limited to, Chatbots, Virtual Assistants, Machine Learning (ML), Deep Learning (DL), Natural Language Processing (NLP), Generative AI (GenAI) models, and any future AI models that are yet to be developed possessing characteristics of Limited Memory Machines. Generative AI models combine Limited Memory Machine technologies, incorporating ML and DL, forming the foundational building blocks of future AI models. For example, Theory of Mind is the next progression of AI that may be able to perceive, connect, and react by generating appropriate reactions in response to an entity with which the AI model is interacting; all of these capabilities rely on the fundamentals of Generative AI. Furthermore, in an evolution into the Self-Aware classification, AI models will be able to understand and evoke emotions in the entities they interact with, as well as possessing their own emotions, beliefs, and needs, all of which rely on Generative AI fundamentals of learning from experiences to generate and draw conclusions about itself and its surroundings. Generative AI models are integral and core to future artificial intelligence models. As described herein, Generative AI refers to present-day Generative AI models and future AI models.

[0079] FIG. **3**A illustrates an AI/ML network diagram **300**A that supports AI-assisted vehicle or occupant decision points. Other branches of AI, such as, but not limited to, computer vision, fuzzy logic, expert systems, neural networks/deep learning, generative AI, and natural language processing, may all be employed in developing the AI model shown in these embodiments. Further, the AI model included in these embodiments is not limited to particular AI algorithms. Any algorithm or combination of algorithms related to supervised, unsupervised, and reinforcement learning algorithms may be employed.

[0080] In one embodiment, Generative AI (GenAI) may be used by the instant solution in the transformation of data. Vehicles are equipped with diverse sensors, cameras, radars, and LIDARs, which collect a vast array of data, such as images, speed readings, GPS data, and acceleration metrics. However, raw data, once acquired, undergoes preprocessing that may involve normalization, anonymization, missing value imputation, or noise reduction to allow the data to be further used effectively.

[0081] The GenAI executes data augmentation following the preprocessing of the data. Due to the limitation of datasets in capturing the vast complexity of real-world vehicle scenarios, augmentation tools are employed to expand the dataset. This might involve image-specific transformations like rotations, translations, or brightness adjustments. For non-image data, techniques like jittering can be used to introduce synthetic noise, simulating a broader set of conditions.

[0082] In the instant solution, data generation is then performed on the data. Tools like Generative Adversarial Networks (GANs) and Variational Autoencoders (VAEs) are trained on existing datasets to generate new, plausible data samples. For example, GANs might be tasked with crafting images showcasing vehicles in uncharted conditions or from unique perspectives. As another example, the synthesis of sensor data may be performed to model and create synthetic readings for such scenarios, enabling thorough system testing without actual physical encounters. A critical step in the use of GenAI, given the safety-critical nature of vehicles, is validation. This validation might include the output data being compared with real-world datasets or using specialized tools like a GAN discriminator to gauge the realism of the crafted samples.

[0083] Vehicle node **310** may include a plurality of sensors **312** that may include but are not limited to, light sensors, weight sensors, cameras, lidar, and radar. In some embodiments, these sensors **312** send data to a database **320** that stores data about the vehicle and occupants of the vehicle. In some embodiments, these sensors **312** send data to one or more decision subsystems **316** in vehicle node **310** to assist in decision-making.

[0084] Vehicle node **310** may include one or more user interfaces (UIs) **314**, such as a steering wheel, navigation controls, audio/video controls, temperature controls, etc. In some embodiments, these UIs **314** send data to a database **320** that stores event data about the UIs **314** that includes but is not limited to selection, state, and display data. In some embodiments, these UIs **314** send data to one or more decision subsystems **316** in vehicle node **310** to assist decision-making.

[0085] Vehicle node **310** may include one or more decision subsystems **316** that drive a decision-making process around, but are not limited to, vehicle control, temperature control, charging control, etc. In some embodiments, the decision subsystems **316** gather data from one or more sensors **312** to aid in the decision-making process. In some embodiments, a decision subsystem **316** may gather data from one or more UIs **314** to aid in the decision-making process. In some embodiments, a decision subsystem **316** may provide feedback to a UI **314**.

[0086] An AI/ML production system **330** may be used by a decision subsystem **316** in a vehicle node **310** to assist in its decision-making process. The AI/ML production system **330** includes one or more AI/ML models **332** that are executed to retrieve the needed data, such as, but not limited to, a prediction, a categorization, a UI prompt, etc. In some embodiments, an AI/ML production system **330** is hosted on a server. In some embodiments, the AI/ML production system **330** is cloud-hosted. In some embodiments, the AI/ML production system **330** is deployed in a distributed multi-node architecture. In some embodiments, the AI production system resides in vehicle node **310**.

[0087] An AI/ML development system **340** creates one or more AI/ML models **332**. In some embodiments, the AI/ML development system **340** utilizes data in the database **320** to develop and train one or more AI models **332**. In some embodiments, the AI/ML development system **340** utilizes feedback data from one or more AI/ML production systems **330** for new model development and/or existing model re-training. In an embodiment, the AI/ML development system **340** resides and executes on a server. In another embodiment the AI/ML development system **340** is cloud hosted. In a further embodiment, the AI/ML development system **340** utilizes a distributed data pipeline/analytics engine.

[0088] Once an AI/ML model **332** has been trained and validated in the AI/ML development system **340**, it may be stored in an AI/ML model registry **360** for retrieval by either the AI/ML development system **340** or by one or more AI/ML production systems **330**. The AI/ML model registry **360** resides in a dedicated server in one embodiment. In some embodiments, the AI/ML model registry **360** is cloud-hosted. The AI/ML model registry **360** is a distributed database in other embodiments. In further embodiments, the AI/ML model registry **360** resides in the AI/ML production system **330**.

[0089] FIG. **3**B illustrates a process **300**B for developing one or more AI/ML models that support AI-assisted vehicle or occupant decision points. An AI/ML development system **340** executes steps to develop an AI/ML model **332** that begins with data extraction **342**, in which data is loaded and ingested from one or more data sources. In some embodiments, vehicle and user data is extracted from a database **320**. In some embodiments, model feedback data is extracted from one or more AI/ML production systems **330**.

[0090] Once the required data has been extracted **342**, it must be prepared **344** for model training. In some embodiments, this step involves statistical testing of the data to see how well it reflects real-world events, its distribution, the variety of data in the dataset, etc. In some embodiments, the results of this statistical testing may lead to one or more data transformations being employed to normalize one or more values in the dataset. In some embodiments, this step includes cleaning data deemed to be noisy. A noisy dataset includes values that do not contribute to the training, such as but are not limited to, null and long string values. Data preparation **344** may be a manual process or an automated process using one or more of the elements, functions described or depicted herein.

[0091] Features of the data are identified and extracted **346**. In some embodiments, a feature of the data is internal to the prepared data from step **344**. In other embodiments, a feature of the data requires a piece of prepared data from step **344** to be enriched by data from another data source to be useful in developing an AI/ML model **332**. In some embodiments, identifying features is a manual process or an automated process using one or more of the elements, functions described or depicted herein. Once the features have been identified, the values of the features are collected into a dataset that will be used to develop the AI/ML model **332**.

[0092] The dataset output from feature extraction step **346** is split **348** into a training and validation data set. The training data set is used to train the AI/ML model **332**, and the validation data set is used to evaluate the performance of the AI/ML model **332** on unseen data.

[0093] The AI/ML model **332** is trained and tuned **350** using the training data set from the data splitting step **348**. In this step, the training data set is fed into an AI/ML algorithm and an initial set of algorithm parameters. The performance of the AI/ML model **332** is then tested within the AI/ML development system **340** utilizing the validation data set from step **348**. These steps may be repeated with adjustments to one or more algorithm parameters until the model's performance is acceptable based on various goals and/or results.

[0094] The AI/ML model **332** is evaluated **352** in a staging environment (not shown) that resembles the ultimate AI/ML production system **330**. This evaluation uses a validation dataset to ensure the performance in an AI/ML production system **330** matches or exceeds expectations. In some embodiments, the validation dataset from step **348** is used. In other embodiments, one or more unseen validation datasets are used. In some embodiments, the staging environment is part of the AI/ML development system **340**. In other embodiments, the staging environment is managed separately from the AI/ML development system **340**. Once the AI/ML model **332** has been validated, it is stored in an AI/ML model registry **360**, which can be retrieved for deployment and future updates. As before, in some embodiments, the model evaluation step **352** is a manual process or an automated process using one or more of the elements, functions described or depicted herein.

[0095] Once an AI/ML model **332** has been validated and published to an AI/ML model registry **360**, it may be deployed **354** to one or more AI/ML production systems **330**. In some embodiments, the performance of deployed AI/ML models **332** is monitored **356** by the AI/ML development system **340**. In some embodiments, AI/ML model **332** feedback data is provided by the AI/ML production system **330** to enable model performance monitoring **356**. In some embodiments, the AI/ML development system **340** periodically requests feedback data for model performance monitoring **356**. In some embodiments, model performance monitoring includes one or more triggers that result in the AI/ML model **332** being updated by repeating steps **342**-**354** with updated data from one or more data sources.

[0096] FIG. **3**C illustrates a process **300**C for utilizing an AI/ML model that supports AI-assisted vehicle or occupant decision points. As stated previously, the AI model utilization process depicted herein reflects ML, which is a particular branch of AI, but the instant solution is not limited to ML and is not limited to any AI algorithm or combination of algorithms.

[0097] Referring to FIG. **3**C, an AI/ML production system **330** may be used by a decision subsystem **316** in vehicle node **310** to assist in its decision-making process. The AI/ML production system **330** provides an application programming interface (API) **334**, executed by an AI/ML server process **336** through which requests can be made. In some embodiments, a request may include an AI/ML model **332** identifier to be executed. In some embodiments, the AI/ML model **332** to be executed is implicit based on the type of request. In some embodiments, a data payload (e.g., to be input to the model during execution) is included in the request. In some embodiments, the data payload includes sensor **312** data from vehicle node **310**. In some embodiments, the data payload includes UI **314** data from vehicle node **310**. In some embodiments, the data payload includes data from other vehicle node **310** subsystems (not shown), including but not limited to, occupant data subsystems. In an embodiment, one or more elements or nodes **320**, **330**, **340**, or **360** may be located in the vehicle **310**.

[0098] Upon receiving the API **334** request, the AI/ML server process **336** may need to transform the data payload or portions of the data payload to be valid feature values into an AI/ML model **332**. Data transformation may include but is not limited to combining data values, normalizing data values, and enriching the incoming data with data from other data sources. Once any required data transformation occurs, the AI/ML server process **336** executes the appropriate AI/ML model **332** using the transformed input data. Upon receiving the execution result, the AI/ML server process **336** responds to the API caller, which is a decision subsystem **316** of vehicle node **310**. In some embodiments, the response may result in an update to a UI **314** in vehicle node **310**. In some embodiments, the response includes a request identifier that can be used later by the decision subsystem **316** to provide feedback on the AI/ML model **332** performance. Further, in some embodiments, immediate performance feedback may be recorded into a model feedback log **338** by the AI/ML server process **336**. In some embodiments, execution model failure is a reason for immediate feedback.

[0099] In some embodiments, the API **334** includes an interface to provide AI/ML model **332** feedback after an AI/ML model **332** execution response has been processed. This mechanism may be used to evaluate the performance of the AI/ML model **332** by enabling the API caller to provide feedback on the accuracy of the model results. For example, if the AI/ML model **332** provided an estimated time of arrival of 20 minutes, but the actual travel time was 24 minutes, that may be indicated. In some embodiments, the feedback interface includes the identifier of the initial request so that it can be used to associate the feedback with the request. Upon receiving a call into the feedback interface of API **334**, the AI/ML server process **336** records the feedback in the model feedback log **338**. In some embodiments, the data in this model feedback log **338** is provided to model performance monitoring **356** in the AI/ML development system **340**. This log data is streamed to the AI/ML development system **340** in one embodiment. In some embodiments, the log data is provided upon request.

[0100] A number of the steps/features that may utilize the AI/ML process described herein include one or more of predicting, by a vehicle, a duration and a severity of an electrical-related event; determining, by the vehicle, a behavior profile of a group of devices at a location, based on a need of at least one of the group of devices during the event, and an amount of energy consumption of at least one of the group of devices during the event; predicting, by the vehicle, an energy requirement during the event at the location, based on the predicted duration, the predicted severity and the determined behavior profile; directing, by the vehicle, energy to be stored prior to the event by both the vehicle and an energy storage unit (ESU) associated with the location; and directing, by the vehicle, a flow of energy to the group of devices during the event, from the ESU and the vehicle; directing, by the vehicle, the flow of energy to the group of devices during the event by first directing energy from a battery of the vehicle to the group of devices; determining, by the vehicle, a first amount of time that the ESU provides energy to the group of devices, based on the determined behavior profile; and determining, by the vehicle, an additional charge to be added to a battery of the vehicle, based on the first amount of time; gathering the amount of energy consumption for the group of devices over a period of time; and storing the gathered amount of energy consumption in the behavior profile; wherein the directing, by the vehicle, the flow of energy to the group of devices during the event is performed by directing energy from a battery of the vehicle to the group of devices; monitoring a state-of-charge of the battery; and, in response to the state-of-charge dropping below a threshold, directing energy from the ESU to the group of devices; wherein the directing, by the vehicle, the flow of energy to the group of devices during the event is performed by directing energy from a battery of the vehicle to the group of devices; determining a minimum state-of-charge of the battery required for the vehicle to transport an occupant from the location to another location during the event; monitoring a state-of-charge of the battery; and, in response to the state-of-charge approaching the minimum state-of-charge, directing energy from the ESU to the group of devices; and determining that at least one device of the group of devices is a medical device associated with an occupant of the location; determining an amount of energy required to power the medical device based on the predicted duration and the predicted severity of the event; and reserving at least the amount of energy on at least one of the ESU or the vehicle for powering the medical device.

[0101] Data associated with any of these steps/features, as well as any other features or functionality described or depicted herein, the AI/ML production system **330**, as well as one or more of the other elements depicted in FIG. **3**C may be used to process this data in a pre-transformation and/or post-transformation process. Data related to this process can be used by the vehicle node **310**. In one embodiment, data related to this process may be used with a charging station/charging point, a server, a wireless device, and/or any of the processors described or depicted herein.

[0102] FIG. **3**D illustrates a process **300**D of designing a new machine learning model via a user interface **370** of the system according to example embodiments. As an example, a model may be output as part of the AI/ML Development System **340**. Referring to FIG. **3**D, a user can use an input mechanism from a menu **372** of a user interface **370** to add pieces/components to a model being developed within a workspace **374** of the user interface **370**.

[0103] The menu **372** includes a plurality of graphical user interface (GUI) menu options which can be selected to reveal additional components that can be added to the model design shown in the workspace **374**. The GUI menu includes options for adding elements to the workspace, such as features which may include neural networks, machine learning models, AI models, data sources, conversion processes (e.g., vectorization, encoding, etc.), analytics, etc. The user can continue to add features to the model and connect them using edges or other means to create a flow within the workspace **374**. For example, the user may add a node **376** to a flow of a new model within the workspace **374**. For example, the user may connect the node **376** to another node in the diagram via an edge **378**, creating a dependency within the diagram. When the user is done, the user can save the model for subsequent training/testing.

[0104] In another example, the name of the object can be identified from a web page or a user interface **370** where the object is visible within a browser or the workspace **374** on the user device. A pop-up within the browser or the workspace **374** can be overlayed where the object is visible, which includes an option to navigate to the identified web page corresponding to the alternative object via a rule set.

[0105] FIG. **3**E illustrates a process **300**E of accessing an object **392** from an object storage **390** of the host platform **380** according to example embodiments. For example, the object storage **390** may store data that is used by the AI models and machine learning (ML) models, training data, expected outputs for testing, training results, and the like. The object storage **390** may also store any other kind of data. Each object may include a unique identifier, a data section **394**, and a metadata section **396**, which provide a descriptive context associated with the data, including data that can later be extracted for purposes of machine learning. The unique identifier may uniquely identify an object with respect to all other objects in the object storage **390**. The data section **394** may include unstructured data such as web pages, digital content, images, audio, text, and the like.

[0106] Instead of breaking files into blocks stored on disks in a file system, the object storage **390** handles objects as discrete units of data stored in a structurally flat data environment. Here, the object storage may not use folders, directories, or complex hierarchies. Instead, each object may be a simple, self-contained repository that includes the data, the metadata, and the unique identifier that a client application can use to locate and access it. In this case, the metadata is more descriptive than a file-based approach. The metadata can be customized with additional context that can later be extracted and leveraged for other purposes, such as data analytics.

[0107] The objects that are stored in the object storage **390** may be accessed via an API **384**. The API **384** may be a Hypertext Transfer Protocol (HTTP)-based RESTful API (also known as a RESTful Web service). The API **384** can be used by the client application to query an object's metadata to locate the desired object (data) via the Internet from anywhere on any device. The API **384** may use HTTP commands such as "PUT" or "POST" to upload an object, "GET" to retrieve an object, "DELETE" to remove an object, and the like.

[0108] The object storage **390** may provide a directory **398** that uses the metadata of the objects to locate appropriate data files. The directory **398** may contain descriptive information about each object stored in the object storage **390**, such as a name, a unique identifier, a creation timestamp, a collection name, etc. To query the object within the object storage **390**, the client application may submit a command, such as an HTTP command, with an identifier of the object **392**, a payload, etc. The object storage **390** can store the actions and results described herein, including associating two or more lists of ranked assets with one another based on variables used by the two or more lists of ranked assets that have a correlation above a predetermined threshold.

[0109] FIG. **4**A illustrates a diagram **400**A depicting the electrification of one or more elements. In one example, a vehicle **402**B may provide power stored in its batteries to one or more elements, including other vehicle(s) **408**B, charging station(s) **406**B, and electric grid(s) **404**B. The electric grid(s) **404**B is/are coupled to one or more of the charging stations **406**B, which may be coupled to one or more of the vehicles **408**B. This configuration allows the distribution of electricity/power received from the vehicle **402**B. The vehicle **402**B may also interact with the other vehicle(s) **408**B, such as via V2V technology, communication over cellular, Wi-Fi, and the like. The vehicle **402**B may also interact wirelessly and/or wired with other vehicles **408**B, the charging station(s) **406**B and/or with the electric grid(s) **404**B. In one example, the vehicle **402**B is routed (or routes itself) in a safe and efficient manner to the electric grid(s) **404**B, the charging station(s) **406**B, or the other vehicle(s) **408**B. Using one or more embodiments of the instant solution, the vehicle **402**B can provide energy to one or more of the elements depicted herein in various advantageous ways as described and/or depicted herein. Further, the safety and efficiency of the vehicle may be increased, and the environment may be positively affected as described and/or depicted herein.

[0110] The term 'energy', 'electricity', 'power', and the like may be used to denote any form of energy received, stored, used, shared, and/or lost by the vehicles(s). The energy may be referred to in conjunction with a voltage source and/or a current supply of charge provided from an entity to the vehicle(s) during a charge/use operation. Energy may also be in the form of fossil fuels (for example, for use with a hybrid vehicle) or via alternative power sources, including but not limited to lithium-based, nickel-based, hydrogen fuel cells, atomic/nuclear energy, fusion-based energy sources, and energy generated during an energy sharing and/or usage operation for increasing or decreasing one or more vehicles energy levels at a given time.

[0111] In one example, the charging station **406**B manages the amount of energy transferred from the vehicle **402**B such that there is sufficient charge remaining in the vehicle **402**B to arrive at a destination. In one example, a wireless connection is used to wirelessly direct an amount of energy transfer between vehicles **408**B, wherein the vehicles may both be in motion. In one embodiment, wireless charging may occur via a fixed charger and batteries of the vehicle in alignment with one another (such as a charging mat in a garage or parking space). In one example, an idle vehicle, such as a vehicle **402**B (which may be autonomous) is directed to provide an amount of energy to a charging station **406**B and return to the original location (for example, its original location or a different destination). In one example, a mobile energy storage unit (not shown) is used to collect surplus energy from at least one other vehicle **408**B and transfer the stored surplus energy at a charging station **406**B. In one example, factors determine an amount of energy to transfer to a charging station **406**B, such as distance, time, as well as traffic conditions, road conditions, environmental/weather conditions, the vehicle's condition (weight, etc.), an occupant(s) schedule while utilizing the vehicle, a prospective occupant(s) schedule waiting for the vehicle, etc. In one example, the vehicle(s) **408**B, the charging station(s) **406**B and/or the electric grid(s) **404**B can provide energy to the vehicle **402**B.

[0112] In one embodiment, a location such as a building, a residence, or the like (not depicted), communicably coupled to one or more of the electric grid **404**B, the vehicle **402**B, and/or the charging station(s) **406**B. The rate of electric flow to one or more of the location, the vehicle **402**B, the other vehicle(s) **408**B is modified, depending on external conditions, such as weather. For example, when the external temperature is extremely hot or extremely cold, raising the chance for an outage of electricity, the flow of electricity to a connected vehicle **402**B/**408**B is slowed to help minimize the chance for an outage.

[0113] In one embodiment, vehicles **402**B and **408**B may be utilized as bidirectional vehicles. Bidirectional vehicles are those that may serve as mobile microgrids that can assist in the supplying of electrical power to the grid **404**B and/or reduce the power consumption when the grid is stressed. Bidirectional vehicles incorporate bidirectional charging, which in addition to receiving a charge to the vehicle, the vehicle can transfer energy from the vehicle to the grid **404**B, otherwise referred to as "V2G". In bidirectional charging, the electricity flows both ways; to the vehicle and from the vehicle. When a vehicle is charged, alternating current (AC) electricity from the grid **404**B is converted to direct current (DC). This may be performed by one or more of the vehicle's own converter or a converter on the charging station **406**B. The energy stored in the vehicle's batteries may be sent in an opposite direction back to the grid. The energy is converted from DC to AC through a converter usually located in the charging station **406**B, otherwise referred to as a bidirectional charger. Further, the instant solution as described and depicted with respect to FIG. 4B can be utilized in this and other networks and/or systems.

[0114] FIG. 4B is a diagram showing interconnections between different elements 400B. The instant solution may be stored and/or executed entirely or partially on and/or by one or more computing devices 414C, 418C, 424C, 428C, 432C, 436C, 406C, 442C and 410C associated with various entities, all communicably coupled and in communication with a network 402C. A database 438C is communicably coupled to the network and allows for the storage and retrieval of data. In one example, the database is an immutable ledger. One or more of the various entities may be a vehicle 404C, one or more service provider 416C, one or more public buildings 422C, one or more traffic infrastructure 426C, one or more residential dwellings 430C, an electric grid/charging station 434C, a microphone 440C, and/or another vehicle 408C. Other entities and/or devices, such as one or more private users using a smartphone 412C, a laptop 420C, an augmented reality (AR) device, a virtual reality (VR) device, and/or any wearable device may also interwork with the instant solution. The smartphone 412C, laptop 420C, the microphone 440C, and other devices may be connected to one or more of the connected computing devices 414C, 418C, 424C, 428C, 432C, 436C, 406C, 442C, and 410C. The one or more public buildings 422C may include various agencies. The one or more public buildings 422C may utilize a computing device 424C. The one or more service provider 416C may include a dealership, a tow truck service, a collision center, or other repair shop. The one or more service provider 416C may utilize a computing apparatus 418C. These various computer devices may be directly and/or communicably coupled to one another, such as via wired networks, wireless networks, blockchain networks, and the like. The microphone 440C may be utilized as a virtual assistant, in one example. In one example, the one or more traffic infrastructure 426C may include one or more traffic signals, one or more sensors including one or more cameras, vehicle speed sensors or traffic sensors, and/or other traffic infrastructure. The one or more traffic infrastructure 426C may utilize a computing device 428C.

[0115] In one embodiment, anytime an electrical charge is given or received to/from a charging station and/or an electrical grid, the entities that allow that to occur are one or more of a vehicle, a charging station, a server, and a network communicably coupled to the vehicle, the charging station, and the electrical grid.

[0116] In one example, a vehicle 408C/404C can transport a person, an object, a permanently or temporarily affixed apparatus, and the like. In one example, the vehicle 408C may communicate with vehicle 404C via V2V communication through the computers associated with each vehicle 406C and 410C and may be referred to as a car, vehicle, automobile, and the like. The vehicle 404C/408C may be a self-propelled wheeled conveyance, such as a car, a sports utility vehicle, a truck, a bus, a van, or other motor or battery-driven or fuel cell-driven vehicle. For example, vehicle 404C/408C may be an electric vehicle, a hybrid vehicle, a hydrogen fuel cell vehicle, a plug-in hybrid vehicle, or any other type of vehicle with a fuel cell stack, a motor, and/or a generator. Other examples of vehicles include bicycles, scooters, trains, planes, boats, and any other form of conveyance that is capable of transportation. The vehicle 404C/408C may be semi-autonomous or autonomous. For example, vehicle 404C/408C may be self-maneuvering and navigate without human input. An autonomous vehicle may have and use one or more sensors and/or a navigation unit to drive autonomously. All of the data described or depicted herein can be stored, analyzed, processed and/or forwarded by one or more of the elements in FIG. 4B.

[0117] FIG. 4C is another block diagram showing interconnections between different elements in one example 400C. A vehicle 412D is presented and includes ECUs 410D, 408D, and a Head Unit (otherwise known as an Infotainment System) 406D. An ECU is an embedded system in automotive electronics controlling one or more of the electrical systems or subsystems in a vehicle. ECUs may include but are not limited to the management of a vehicle's engine, brake system, gearbox system, door locks, dashboard, airbag system, infotainment system, electronic differential, and active suspension. ECUs are connected to the vehicle's Controller Area Network (CAN) bus 416D. The ECUs may also communicate with a vehicle computer 404D via the CAN bus 416D. The vehicle's processors/sensors (such as the vehicle computer) 404D can communicate with external elements, such as a server 418D via a network 402D (such as the Internet). Each ECU 410D, 408D, and Head Unit 406D may contain its own security policy. The security policy defines permissible processes that can be executed in the proper context. In one example, the security policy may be partially or entirely provided in the vehicle computer 404D.

[0118] ECUs 410D, 408D, and Head Unit 406D may each include a custom security functionality element 414D defining authorized processes and contexts within which those processes are permitted to run. Context-based authorization to determine validity if a process can be executed allows ECUs to maintain secure operation and prevent unauthorized access from elements such as the vehicle's CAN Bus. When an ECU encounters a process that is unauthorized, that ECU can block the process from operating. Automotive ECUs can use different contexts to determine whether a process is operating within its permitted bounds, such as proximity contexts, nearby objects, distance to approaching objects, speed, and trajectory relative to other moving objects, and operational contexts such as an indication of whether the vehicle is moving or parked, the vehicle's current speed, the transmission state, user-related contexts such as devices connected to the transport via wireless protocols, use of the infotainment, cruise control, parking assist, driving assist, location-based contexts, and/or other contexts.

[0119] Referring to FIG. 4D, an operating environment 400D for a connected vehicle, is illustrated according to some embodiments. As depicted, the vehicle 410E includes a CAN bus 408E connecting elements 412E-426E of the vehicle. Other elements may be connected to the CAN bus and are not depicted herein. The depicted elements connected to the CAN bus include a sensor set 412E, Electronic Control Units 414E, autonomous features or Advanced Driver Assistance Systems (ADAS) 416E, and the navigation system 418E. In some embodiments, the vehicle 410E includes a processor 420E, a memory 422E, a communication unit 424E, and an electronic display 426E.

[0120] The processor 420E includes an arithmetic logic unit, a microprocessor, a general-purpose controller, and/or a similar processor array to perform computations and provide electronic display signals to a display unit **426**E. The processor **420**E processes data signals and may include various computing architectures, including a complex instruction set computer (CISC) architecture, a reduced instruction set computer (RISC) architecture, or an architecture implementing a combination of instruction sets. The vehicle **410**E may include one or more processors **420**E. Other processors, operating systems, sensors, displays, and physical configurations that are communicably coupled to one another (not depicted) may be used with the instant solution.

[0121] Memory **422**E is a non-transitory memory storing instructions or data that may be accessed and executed by the processor **420**E. The instructions and/or data may include code to perform the techniques described herein. The memory **422**E may be a dynamic random-access memory (DRAM) device, a static random-access memory (SRAM) device, flash memory, or another memory device. In some embodiments, the memory **422**E also may include non-volatile memory or a similar permanent storage device and media, which may include a hard disk drive, a floppy disk drive, a compact disc read only memory (CD-ROM) device, a digital versatile disk read only memory (DVD-ROM) device, a digital versatile disk random access memory (DVD-RAM) device, a digital versatile disk rewritable (DVD-RW) device, a flash memory device, or some other mass storage device for storing information on a permanent basis. A portion of the memory **422**E may be reserved for use as a buffer or virtual random-access memory (virtual RAM). The vehicle **410**E may include one or more memories **422**E without deviating from the current solution.

[0122] The memory **422**E of the vehicle **410**E may store one or more of the following types of data: navigation route data **418**E, and autonomous features data **416**E. In some embodiments, the memory **422**E stores data that may be necessary for the navigation application **418**E to provide the functions.

[0123] The navigation system **418**E may describe at least one navigation route including a start point and an endpoint. In some embodiments, the navigation system **418**E of the vehicle **410**E receives a request from a user for navigation routes wherein the request includes a starting point and an ending point. The navigation system **418**E may query a real-time data server **404**E (via a network **402**E), such as a server that provides driving directions, for navigation route data corresponding to navigation routes, including the start point and the endpoint. The real-time data server **404**E transmits the navigation route data to the vehicle **410**E via a wireless network **402**E, and the communication system **424**E stores the navigation data **418**E in the memory **422**E of the vehicle **410**E.

[0124] The ECU **414**E controls the operation of many of the systems of the vehicle **410**E, including the ADAS systems **416**E. The ECU **414**E may, responsive to instructions received from the navigation system **418**E, deactivate any unsafe and/or unselected autonomous features for the duration of a journey controlled by the ADAS systems **416**E. In this way, the navigation system **418**E may control whether ADAS systems **416**E are activated or enabled so that they may be activated for a given navigation route.

[0125] The sensor set **412**E may include any sensors in the vehicle **410**E generating sensor data. For example, the sensor set **412**E may include short-range sensors and long-range sensors. In some embodiments, the sensor set **412**E of the vehicle **410**E may include one or more of the following vehicle sensors: a camera, a Light Detection and Ranging (Lidar) sensor, an ultrasonic sensor, an automobile engine sensor, a radar sensor, a laser altimeter, a manifold absolute pressure sensor, an infrared detector, a motion detector, a thermostat, a sound detector, a carbon monoxide sensor, a carbon dioxide sensor, an oxygen sensor, a mass airflow sensor, an engine coolant temperature sensor, a throttle position sensor, a crankshaft position sensor, a valve timer, an air-fuel ratio meter, a blind spot meter, a curb feeler, a defect detector, a Hall effect sensor, a parking sensor, a radar gun, a speedometer, a speed sensor, a tire-pressure monitoring sensor, a torque sensor, a transmission fluid temperature sensor, a turbine speed sensor (TSS), a variable reluctance sensor, a vehicle speed sensor (VSS), a water sensor, a wheel speed sensor, a global positioning system (GPS) sensor, a mapping functionality, and any other type of automotive sensor. The navigation system **418**E may store the sensor data in the memory **422**E.

[0126] The communication unit **424**E transmits and receives data to and from the network **402**E or to another communication channel. In some embodiments, the communication unit **424**E may include a dedicated short-range communication (DSRC) transceiver, a DSRC receiver, and other hardware or software necessary to make the vehicle **410**E a DSRC-equipped device.

[0127] The vehicle **410**E may interact with other vehicles **406**E via V2V technology. V2V communication includes sensing radar information corresponding to relative distances to external objects, receiving GPS information of the vehicles, setting areas where the other vehicles **406**E are located based on the sensed radar information, calculating probabilities that the GPS information of the object vehicles will be located at the set areas, and identifying vehicles and/or objects corresponding to the radar information and the GPS information of the object vehicles based on the calculated probabilities, in one example.

[0128] For a vehicle to be adequately secured, the vehicle must be protected from unauthorized physical access as well as unauthorized remote access (e.g., cyber-threats). To prevent unauthorized physical access, a vehicle is equipped with a secure access system such as a keyless entry in one example. Meanwhile, security protocols are added to a vehicle's computers and computer networks to facilitate secure remote communications to and from the vehicle in one example.

[0129] ECUs are nodes within a vehicle that control tasks such as activating the windshield wipers to tasks such as an anti-lock brake system. ECUs are often connected to one another through the vehicle's central network, which may be referred to as a controller area network (CAN). State-of-the-art features such as autonomous driving are strongly reliant on implementing new, complex ECUs such as ADAS, sensors, and the like. While these new technologies have helped improve the safety and driving experience of a vehicle, they have also increased the number of externally-communicating units inside of the vehicle, making them more vulnerable to attack. Below are some examples of protecting the vehicle from physical intrusion and remote intrusion.

[0130] In one embodiment, a CAN includes a CAN bus with a high and low terminal and a plurality of ECUs, which are connected to the CAN bus via wired connections. The

CAN bus is designed to allow microcontrollers and devices to communicate with each other in an application without a host computer. The CAN bus implements a message-based protocol (i.e., ISO 11898 standards) that allows ECUs to send commands to one another at a root level. Meanwhile, the ECUs represent controllers for controlling electrical systems or subsystems within the vehicle. Examples of the electrical systems include power steering, anti-lock brakes, air-conditioning, tire pressure monitoring, cruise control, and many other features.

[0131] In this example, the ECU includes a transceiver and a microcontroller. The transceiver may be used to transmit and receive messages to and from the CAN bus. For example, the transceiver may convert the data from the microcontroller into a format of the CAN bus and also convert data from the CAN bus into a format for the microcontroller. Meanwhile, the microcontroller interprets the messages and also decides what messages to send using ECU software installed therein in one example.

[0132] To protect the CAN from cyber threats, various security protocols may be implemented. For example, sub-networks (e.g., sub-networks A and B, etc.) may be used to divide the CAN into smaller sub-CANs and limit an attacker's capabilities to access the vehicle remotely. In one embodiment, a firewall (or gateway, etc.) may be added to block messages from crossing the CAN bus across sub-networks. If an attacker gains access to one sub-network, the attacker will not have access to the entire network. To make sub-networks even more secure, the most critical ECUs are not placed on the same sub-network, in one example.

[0133] In addition to protecting a vehicle's internal network, vehicles may also be protected when communicating with external networks such as the Internet. One of the benefits of having a vehicle connection to a data source such as the Internet is that information from the vehicle can be sent through a network to remote locations for analysis. Examples of vehicle information include GPS, onboard diagnostics, tire pressure, and the like. These communication systems are often referred to as telematics because they involve the combination of telecommunications and informatics. Further, the instant solution as described and depicted can be utilized in this and other networks and/or systems, including those that are described and depicted herein.

[0134] FIG. **4**E illustrates an example **400**E of vehicles **402**I and **408**I performing secured V2V communications using security certificates, according to example embodiments. Referring to FIG. **4**E, the vehicles **402**I and **408**I may communicate via V2V communications over a short-range network, a cellular network, or the like. Before sending messages, the vehicles **402**I and **408**I may sign the messages using a respective public key certificate. For example, the vehicle **402**I may sign a V2V message using a public key certificate **404**I. Likewise, the vehicle **408**I may sign a V2V message using a public key certificate **410**I. The public key certificates **404**I and **410**I are associated with the vehicles **402**I and **408**I, respectively, in one example.

[0135] Upon receiving the communications from each other, the vehicles may verify the signatures with a certificate authority **406**I or the like. For example, the vehicle **408**I may verify with the certificate authority **406**I that the public key certificate **404**I used by vehicle **402**I to sign a V2V communication is authentic. If the vehicle **408**I successfully verifies the public key certificate **404**I, the vehicle knows that the data is from a legitimate source. Likewise, the vehicle **402**I may verify with the certificate authority **406**I that the public key certificate **410**I used by the vehicle **408**I to sign a V2V communication is authentic. Further, the instant solution as described and depicted with respect to FIG. **4**E can be utilized in this and other networks and/or systems including those that are described and depicted herein.

[0136] In some embodiments, a computer may include a security processor. In particular, the security processor may perform authorization, authentication, cryptography (e.g., encryption), and the like, for data transmissions that are sent between ECUs and other devices on a CAN bus of a vehicle, and also data messages that are transmitted between different vehicles. The security processor may include an authorization module, an authentication module, and a cryptography module. The security processor may be implemented within the vehicle's computer and may communicate with other vehicle elements, for example, the ECUs/CAN network, wired and wireless devices such as wireless network interfaces, input ports, and the like. The security processor may ensure that data frames (e.g., CAN frames, etc.) that are transmitted internally within a vehicle (e.g., via the ECUs/CAN network) are secure. Likewise, the security processor can ensure that messages transmitted between different vehicles and devices attached or connected via a wire to the vehicle's computer are also secured.

[0137] For example, the authorization module may store passwords, usernames, PIN codes, biometric scans, and the like for different vehicle users. The authorization module may determine whether a user (or technician) has permission to access certain settings such as a vehicle's computer. In some embodiments, the authorization module may communicate with a network interface to download any necessary authorization information from an external server. When a user desires to make changes to the vehicle settings or modify technical details of the vehicle via a console or GUI within the vehicle or via an attached/connected device, the authorization module may require the user to verify themselves in some way before such settings are changed. For example, the authorization module may require a username, a password, a PIN code, a biometric scan, a predefined line drawing or gesture, and the like. In response, the authorization module may determine whether the user has the necessary permissions (access, etc.) being requested.

[0138] The authentication module may be used to authenticate internal communications between ECUs on the CAN network of the vehicle. As an example, the authentication module may provide information for authenticating communications between the ECUs. As an example, the authentication module may transmit a bit signature algorithm to the ECUs of the CAN network. The ECUs may use the bit signature algorithm to insert authentication bits into the CAN fields of the CAN frame. All ECUs on the CAN network typically receive each CAN frame. The bit signature algorithm may dynamically change the position, amount, etc., of authentication bits each time a new CAN frame is generated by one of the ECUs. The authentication module may also provide a list of ECUs that are exempt (safe list) and that do not need to use the authentication bits. The authentication module may communicate with a remote server to retrieve updates to the bit signature algorithm and the like.

[0139] The encryption module may store asymmetric key pairs to be used by the vehicle to communicate with other external user devices and vehicles. For example, the encryption module may provide a private key to be used by the vehicle to encrypt/decrypt communications, while the corresponding public key may be provided to other user devices and vehicles to enable the other devices to decrypt/encrypt the communications. The encryption module may communicate with a remote server to receive new keys, updates to keys, keys of new vehicles, users, etc., and the like. The encryption module may also transmit any updates to a local private/public key pair to the remote server.

[0140] FIG. **5**A illustrates an example vehicle configuration **500**A for managing database transactions associated with a vehicle, according to example embodiments. Referring to FIG. **5**A, as a particular vehicle **525** is engaged in transactions (e.g., vehicle service, dealer transactions, delivery/pickup, transportation services, etc.), the vehicle may receive assets **510** and/or expel/transfer assets **512** according to a transaction(s). A vehicle processor **526** resides in the vehicle **525** and communication exists between the vehicle processor **526**, a database **530**, and the transaction module **520**. The transaction module **520** may record information, such as assets, parties, credits, service descriptions, date, time, location, results, notifications, unexpected events, etc. Those transactions in the transaction module **520** may be replicated into a database **530**. The database **530** can be one of a SQL database, a relational database management system (RDBMS), a relational database, a non-relational database, a blockchain, a distributed ledger, and may be on board the vehicle, may be off-board the vehicle, may be accessed directly and/or through a network, or be accessible to the vehicle.

[0141] In one embodiment, a vehicle may engage with another vehicle to perform various actions such as to share, transfer, acquire service calls, etc. when the vehicle has reached a status where the services need to be shared with another vehicle. For example, the vehicle may be due for a battery charge and/or may have an issue with a tire and may be in route to pick up a package for delivery. A vehicle processor resides in the vehicle and communication exists between the vehicle processor, a first database, and a transaction module. The vehicle may notify another vehicle, which is in its network and which operates on its blockchain member service. A vehicle processor resides in another vehicle and communication exists between the vehicle processor, a second database, the vehicle processor, and a transaction module. The another vehicle may then receive the information via a wireless communication request to perform the package pickup from the vehicle and/or from a server (not shown). The transactions are logged in the transaction modules and of both vehicles. The credits are transferred from the vehicle to the other vehicle and the record of the transferred service is logged in the first database, assuming that the blockchains are different from one another, or are logged in the same blockchain used by all members. The first database can be one of a SQL database, an RDBMS, a relational database, a non-relational database, a blockchain, a distributed ledger, and may be on board the vehicle, may be off-board the vehicle, may be accessible directly and/or through a network.

[0142] FIG. **5**B illustrates a blockchain architecture configuration **500**B, according to example embodiments. Referring to FIG. **5**B, the blockchain architecture **500**B may include certain blockchain elements, for example, a group of blockchain member nodes **502**-**505** as part of a blockchain group **510**. In one example embodiment, a permissioned blockchain is not accessible to all parties but only to those members with permissioned access to the blockchain data. The blockchain nodes participate in a number of activities, such as blockchain entry addition and validation process (consensus). One or more of the blockchain nodes may endorse entries based on an endorsement policy and may provide an ordering service for all blockchain nodes. A blockchain node may initiate a blockchain action (such as an authentication) and seek to write to a blockchain immutable ledger stored in the blockchain, a copy of which may also be stored on the underpinning physical infrastructure.

[0143] The blockchain transactions **520** are stored in memory of computers as the transactions are received and approved by the consensus model dictated by the members' nodes. Approved transactions **526** are stored in current blocks of the blockchain and committed to the blockchain via a committal procedure, which includes performing a hash of the data contents of the transactions in a current block and referencing a previous hash of a previous block. Within the blockchain, one or more smart contracts **530** may exist that define the terms of transaction agreements and actions included in smart contract executable application code **532**, such as registered recipients, vehicle features, requirements, permissions, sensor thresholds, etc. The code may be configured to identify whether requesting entities are registered to receive vehicle services, what service features they are entitled/required to receive given their profile statuses and whether to monitor their actions in subsequent events. For example, when a service event occurs and a user is riding in the vehicle, the sensor data monitoring may be triggered, and a certain parameter, such as a vehicle charge level, may be identified as being above/below a particular threshold for a particular period of time, then the result may be a change to a current status, which requires an alert to be sent to the managing party (i.e., vehicle owner, vehicle operator, server, etc.) so the service can be identified and stored for reference. The vehicle sensor data collected may be based on types of sensor data used to collect information about vehicle's status. The sensor data may also be the basis for the vehicle event data **534**, such as a location(s) to be traveled, an average speed, a top speed, acceleration rates, whether there were any collisions, was the expected route taken, what is the next destination, whether safety measures are in place, whether the vehicle has enough charge/fuel, etc. All such information may be the basis of smart contract terms **530**, which are then stored in a blockchain. For example, sensor thresholds stored in the smart contract can be used as the basis for whether a detected service is necessary and when and where the service should be performed.

[0144] In one embodiment, a blockchain logic example includes a blockchain application interface as an API or plug-in application that links to the computing device and execution platform for a particular transaction. The blockchain configuration may include one or more applications, which are linked to application programming interfaces (APIs) to access and execute stored program/application code (e.g., smart contract executable code, smart contracts, etc.), which can be created according to a customized configuration sought by participants and can maintain their own state, control their own assets, and receive external information. This can be deployed as an entry and installed, via appending to the distributed ledger, on all blockchain nodes.

[0145] The smart contract application code provides a basis for the blockchain transactions by establishing application code, which when executed causes the transaction terms and conditions to become active. The smart contract, when executed, causes certain approved transactions to be generated, which are then forwarded to the blockchain platform. The platform includes a security/authorization, computing devices, which execute the transaction management and a storage portion as a memory that stores transactions and smart contracts in the blockchain.

[0146] The blockchain platform may include various layers of blockchain data, services (e.g., cryptographic trust services, virtual execution environment, etc.), and underpinning physical computer infrastructure that may be used to receive and store new entries and provide access to auditors, which are seeking to access data entries. The blockchain may expose an interface that provides access to the virtual execution environment necessary to process the program code and engage the physical infrastructure. Cryptographic trust services may be used to verify entries such as asset exchange entries and keep information private.

[0147] The blockchain architecture configuration of FIGS. **5**A and **5**B may process and execute program/application code via one or more interfaces exposed, and services provided, by the blockchain platform. As a non-limiting example, smart contracts may be created to execute reminders, updates, and/or other notifications subject to the changes, updates, etc. The smart contracts can themselves be used to identify rules associated with authorization and access requirements and usage of the ledger. For example, the information may include a new entry, which may be processed by one or more processing entities (e.g., processors, virtual machines, etc.) included in the blockchain layer. The result may include a decision to reject or approve the new entry based on the criteria defined in the smart contract and/or a consensus of the peers. The physical infrastructure may be utilized to retrieve any of the data or information described herein.

[0148] Within smart contract executable code, a smart contract may be created via a high-level application and programming language, and then written to a block in the blockchain. The smart contract may include executable code that is registered, stored, and/or replicated with a blockchain (e.g., distributed network of blockchain peers). An entry is an execution of the smart contract code, which can be performed in response to conditions associated with the smart contract being satisfied. The executing of the smart contract may trigger a trusted modification(s) to a state of a digital blockchain ledger. The modification(s) to the blockchain ledger caused by the smart contract execution may be automatically replicated throughout the distributed network of blockchain peers through one or more consensus protocols.

[0149] The smart contract may write data to the blockchain in the format of key-value pairs. Furthermore, the smart contract code can read the values stored in a blockchain and use them in application operations. The smart contract code can write the output of various logic operations into the blockchain. The code may be used to create a temporary data structure in a virtual machine or other computing platform. Data written to the blockchain can be public and/or can be encrypted and maintained as private. The temporary data that is used/generated by the smart contract is held in memory by the supplied execution environment, then deleted once the data needed for the blockchain is identified.

[0150] A smart contract executable code may include the code interpretation of a smart contract, with additional features. As described herein, the smart contract executable code may be program code deployed on a computing network, where it is executed and validated by chain validators together during a consensus process. The smart contract executable code receives a hash and retrieves from the blockchain a hash associated with the data template created by use of a previously stored feature extractor. If the hashes of the hash identifier and the hash created from the stored identifier template data match, then the smart contract executable code sends an authorization key to the requested service. The smart contract executable code may write to the blockchain data associated with the cryptographic details.

[0151] FIG. **5**C illustrates a blockchain configuration for storing blockchain transaction data, according to example embodiments. Referring to FIG. **5**C, the example configuration **500**C provides for the vehicle **562**, the user device **564** and a server **566** sharing information with a distributed ledger (i.e., blockchain) **568**. The server may represent a service provider entity inquiring with a vehicle service provider to share user profile rating information in the event that a known and established user profile is attempting to rent a vehicle with an established rated profile. The server **566** may be receiving and processing data related to a vehicle's service requirements. As the service events occur, such as the vehicle sensor data indicates a need for fuel/charge, a maintenance service, etc., a smart contract may be used to invoke rules, thresholds, sensor information gathering, etc., which may be used to invoke the vehicle service event. The blockchain transaction data **570** is saved for each transaction, such as the access event, the subsequent updates to a vehicle's service status, event updates, etc. The transactions may include the parties, the requirements (e.g., 18 years of age, service eligible candidate, valid driver's license, etc.), compensation levels, the distance traveled during the event, the registered recipients permitted to access the event and host a vehicle service, rights/permissions, sensor data retrieved during the vehicle event operation to log details of the next service event and identify a vehicle's condition status, and thresholds used to make determinations about whether the service event was completed and whether the vehicle's condition status has changed.

[0152] FIG. **5**D illustrates blockchain blocks that can be added to a distributed ledger, according to example embodiments, and contents of block structures **582**A to **582***n*. Referring to FIG. **5**D, clients (not shown) may submit entries to blockchain nodes to enact activity on the blockchain. As an example, clients may be applications that act on behalf of a requester, such as a device, person, or entity to propose entries for the blockchain. The plurality of blockchain peers (e.g., blockchain nodes) may maintain a state of the blockchain network and a copy of the distributed ledger. Different types of blockchain nodes/peers may be present in the blockchain network including endorsing peers, which simulate and endorse entries proposed by clients and committing peers which verify endorsements, validate entries, and commit entries to the distributed ledger. In this example, the blockchain nodes may perform the role of endorser node, committer node, or both.

[0153] The instant system includes a blockchain that stores immutable, sequenced records in blocks, and a state database (current world state) maintaining a current state of the blockchain. One distributed ledger may exist per channel and each peer maintains its own copy of the distributed ledger for each channel of which they are a member. The instant blockchain is an entry log, structured as hash-linked blocks where each block contains a sequence of N entries. Blocks may include various components such as those shown in FIG. **5**D. The linking of the blocks may be generated by adding a hash of a prior block's header within a block header of a current block. In this way, all entries on the blockchain are sequenced and cryptographically linked together preventing tampering with blockchain data without breaking the hash links. Furthermore, because of the links, the latest block in the blockchain represents every entry that has come before it. The instant blockchain may be stored on a peer file system (local or attached storage), which supports an append-only blockchain workload.

[0154] The current state of the blockchain and the distributed ledger may be stored in the state database. Here, the current state data represents the latest values for all keys ever included in the chain entry log of the blockchain. Smart contract executable code invocations execute entries against the current state in the state database. To make these smart contract executable code interactions extremely efficient, the latest values of all keys are stored in the state database. The state database may include an indexed view into the entry log of the blockchain, it can therefore be regenerated from the chain at any time. The state database may automatically get recovered (or generated if needed) upon peer startup, before entries are accepted.

[0155] Endorsing nodes receive entries from clients and endorse the entry based on simulated results. Endorsing nodes hold smart contracts, which simulate the entry proposals. When an endorsing node endorses an entry, the endorsing nodes creates an entry endorsement, which is a signed response from the endorsing node to the client application indicating the endorsement of the simulated entry. The method of endorsing an entry depends on an endorsement policy that may be specified within smart contract executable code. An example of an endorsement policy is "the majority of endorsing peers must endorse the entry." Different channels may have different endorsement policies. Endorsed entries are forwarded by the client application to an ordering service.

[0156] The ordering service accepts endorsed entries, orders them into a block, and delivers the blocks to the committing peers. For example, the ordering service may initiate a new block when a threshold of entries has been reached, a timer times out, or another condition. In this example, blockchain node is a committing peer that has received a data block **582**A for storage on the blockchain. The ordering service may be made up of a cluster of orderers. The ordering service does not process entries, smart contracts, or maintain the shared ledger. Rather, the ordering service may accept the endorsed entries and specifies the order in which those entries are committed to the distributed ledger. The architecture of the blockchain network may be designed such that the specific implementation of 'ordering' becomes a pluggable component.

[0157] Entries are written to the distributed ledger in a consistent order. The order of entries is established to ensure that the updates to the state database are valid when they are committed to the network. Unlike a cryptocurrency blockchain system where ordering occurs through the solving of a cryptographic puzzle, or mining, in this example the parties of the distributed ledger may choose the ordering mechanism that best suits that network.

[0158] Referring to FIG. **5**D, a block **582**A (also referred to as a data block) that is stored on the blockchain and/or the distributed ledger may include multiple data segments such as a block header **584**A to **584***n*, transaction-specific data **586**A to **586***n*, and block metadata **588**A to **588***n*. It should be appreciated that the various depicted blocks and their contents, such as block **582**A and its contents are merely for purposes of an example and are not meant to limit the scope of the example embodiments. In some cases, both the block header **584**A and the block metadata **588**A may be smaller than the transaction-specific data **586**A, which stores entry data; however, this is not a requirement. The block **582**A may store transactional information of N entries (e.g., 100, 500, 1000, 2000, 3000, etc.) within the block data **590**A to **590***n*. The block **582**A may also include a link to a previous block (e.g., on the blockchain) within the block header **584**A. In particular, the block header **584**A may include a hash of a previous block's header. The block header **584**A may also include a unique block number, a hash of the block data **590**A of the current block **582**A, and the like. The block number of the block **582**A may be unique and assigned in an incremental/sequential order starting from zero. The first block in the blockchain may be referred to as a genesis block, which includes information about the blockchain, its members, the data stored therein, etc.

[0159] The block data **590**A may store entry information of each entry that is recorded within the block. For example, the entry data may include one or more of a type of the entry, a version, a timestamp, a channel ID of the distributed ledger, an entry ID, an epoch, a payload visibility, a smart contract executable code path (deploy tx), a smart contract executable code name, a smart contract executable code version, input (smart contract executable code and functions), a client (creator) identify such as a public key and certificate, a signature of the client, identities of endorsers, endorser signatures, a proposal hash, smart contract executable code events, response status, namespace, a read set (list of key and version read by the entry, etc.), a write set (list of key and value, etc.), a start key, an end key, a list of keys, a Merkel tree query summary, and the like. The entry data may be stored for each of the N entries.

[0160] In some embodiments, the block data **590**A may also store transaction-specific data **586**A, which adds additional information to the hash-linked chain of blocks in the blockchain. Accordingly, the data **586**A can be stored in an immutable log of blocks on the distributed ledger. Some of the benefits of storing such data **586**A are reflected in the various embodiments disclosed and depicted herein. The block metadata **588**A may store multiple fields of metadata (e.g., as a byte array, etc.). Metadata fields may include signature on block creation, a reference to a last configuration block, an entry filter identifying valid and invalid entries within the block, last offset persisted of an ordering service that ordered the block, and the like. The signature, the last configuration block, and the orderer metadata may be added by the ordering service. Meanwhile, a committer of the block (such as a blockchain node) may add validity/invalidity information based on an endorsement policy, verification of read/write sets, and the like. The entry filter may include a byte array of a size equal to the number of entries in the block data and a validation code identifying whether an entry was valid/invalid.

[0161] The other blocks **582**B to **582***n* in the blockchain also have headers, files, and values. However, unlike the first block **582**A, each of the headers **584**A to **584***n* in the other blocks includes the hash value of an immediately preceding block. The hash value of the immediately preceding block may be just the hash of the header of the previous block or may be the hash value of the entire previous block. By including the hash value of a preceding block in each of the remaining blocks, a trace can be performed from the Nth block back to the genesis block (and the associated original file) on a block-by-block basis, as indicated by arrows **592**, to establish an auditable and immutable chain-of-custody.

[0162] FIG. **5**E illustrates a process **500**E of a new block being added to a distributed ledger **520**E, according to example embodiments, and FIG. **5**D illustrates the contents of FIG. **5**E's new data block structure **530**E for blockchain, according to example embodiments. Referring to FIG. **5**E, clients (not shown) may submit transactions to blockchain nodes **511**E, **512**E, and/or **513**E. Clients may be instructions received from any source to enact activity on the blockchain **522**E. As an example, clients may be applications that act on behalf of a requester, such as a device, person, or entity to propose transactions for the blockchain. The plurality of blockchain peers (e.g., blockchain nodes **511**E, **512**E, and **513**E) may maintain a state of the blockchain network and a copy of the distributed ledger **520**E. Different types of blockchain nodes/peers may be present in the blockchain network including endorsing peers which simulate and endorse transactions proposed by clients and committing peers which verify endorsements, validate transactions, and commit transactions to the distributed ledger **520**E. In this example, the blockchain nodes **511**E, **512**E, and **513**E may perform the role of endorser node, committer node, or both.

[0163] The distributed ledger **520**E includes a blockchain which stores immutable, sequenced records in blocks, and a state database **524**E (current world state) maintaining a current state of the blockchain **522**E. One distributed ledger **520**E may exist per channel and each peer maintains its own copy of the distributed ledger **520**E for each channel of which they are a member. The blockchain **522**E is a transaction log, structured as hash-linked blocks where each block contains a sequence of N transactions. The linking of the blocks (shown by arrows in FIG. **5**E) may be generated by adding a hash of a prior block's header within a block header of a current block. In this way, all transactions on the blockchain **522**E are sequenced and cryptographically linked together preventing tampering with blockchain data without breaking the hash links. Furthermore, because of the links, the latest block in the blockchain **522**E represents every transaction that has come before it. The blockchain **522**E may be stored on a peer file system (local or attached storage), which supports an append-only blockchain workload.

[0164] The current state of the blockchain **522**E and the distributed ledger **520**E may be stored in the state database **524**E. Here, the current state data represents the latest values for all keys ever included in the chain transaction log of the blockchain **522**E. Chaincode invocations execute transactions against the current state in the state database **524**E. To make these chaincode interactions extremely efficient, the latest values of all keys are stored in the state database **524**E. The state database **524**E may include an indexed view into the transaction log of the blockchain **522**E, and it can therefore be regenerated from the chain at any time. The state database **524**E may automatically get recovered (or generated if needed) upon peer startup, before transactions are accepted.

[0165] Endorsing nodes receive transactions from clients and endorse the transaction based on simulated results. Endorsing nodes hold smart contracts which simulate the transaction proposals. When an endorsing node endorses a transaction, the endorsing node creates a transaction endorsement which is a signed response from the endorsing node to the client application indicating the endorsement of the simulated transaction. The method of endorsing a transaction depends on an endorsement policy which may be specified within chaincode. An example of an endorsement policy is "the majority of endorsing peers must endorse the transaction." Different channels may have different endorsement policies. Endorsed transactions are forwarded by the client application to the ordering service **510**E.

[0166] The ordering service **510**E accepts endorsed transactions, orders them into a block, and delivers the blocks to the committing peers. For example, the ordering service **510**E may initiate a new block when a threshold of transactions has been reached, a timer times out, or another condition. In the example of FIG. **5**E, blockchain node **512**E is a committing peer that has received a new data block **530**E for storage on blockchain **522**E. The first block in the blockchain may be referred to as a genesis block which includes information about the blockchain, its members, the data stored therein, etc.

[0167] The ordering service **510**E may be made up of a cluster of orderers. The ordering service **510**E does not process transactions, smart contracts, or maintain the shared ledger. Rather, the ordering service **510**E may accept the endorsed transactions and specifies the order in which those transactions are committed to the distributed ledger **522**E. The architecture of the blockchain network may be designed such that the specific implementation of 'ordering' becomes a pluggable component.

[0168] Transactions are written to the distributed ledger **520**E in a consistent order. The order of transactions is established to ensure that the updates to the state database **524**E are valid when they are committed to the network. Unlike a cryptocurrency blockchain system where ordering occurs through the solving of a cryptographic puzzle, or mining, in this example the parties of the distributed ledger **520**E may choose the ordering mechanism that best suits the network.

[0169] When the ordering service **510**E initializes a new data block **530**E, the new data block **530**E may be broadcast to committing peers (e.g., blockchain nodes **511**E, **512**E, and **513**E). In response, each committing peer validates the transaction within the new data block **530**E by checking to make sure that the read set and the write set still match the current world state in the state database **524**E. Specifically, the committing peer can determine whether the read data that existed when the endorsers simulated the transaction is identical to the current world state in the state database **524**E. When the committing peer validates the transaction, the transaction is written to the blockchain **522**E on the distributed ledger **520**E, and the state database **524**E is updated with the write data from the read-write set. If a transaction fails, that is, if the committing peer finds that the read-write set does not match the current world state in the state database **524**E, the transaction ordered into a block will still be included in that block, but it will be marked as invalid, and the state database **524**E will not be updated.

[0170] Referring to FIG. **5**F **500**F, a new data block **530** (also referred to as a data block) that is stored on the blockchain **522**E of the distributed ledger **520**E may include multiple data segments such as a block header **540**, block data **550**, and block metadata **560**. It should be appreciated that the various depicted blocks and their contents, such as new data block **530** and its contents shown in FIG. **5**F, are merely examples and are not meant to limit the scope of the example embodiments. The new data block **530** may store transactional information of N transaction(s) (e.g., 1, 10, 100, 500, 1000, 2000, 3000, etc.) within the block data **550**. The new data block **530** may also include a link to a previous block (e.g., on the blockchain **522**E in FIG. **5**E) within the block header **540**. In particular, the block header **540** may include a hash of a previous block's header. The block header **540** may also include a unique block number, a hash of the block data **550** of the new data block **530**, and the like. The block number of the new data block **530** may be unique and assigned in various orders, such as an incremental/sequential order starting from zero.

[0171] The block data **550** may store transactional information of each transaction that is recorded within the new data block **530**. For example, the transaction data may include one or more of a type of the transaction, a version, a timestamp, a channel ID of the distributed ledger **520**E (shown in FIG. **5**E), a transaction ID, an epoch, a payload visibility, a chaincode path (deploy tx), a chaincode name, a chaincode version, input (chaincode and functions), a client (creator) identify such as a public key and certificate, a signature of the client, identities of endorsers, endorser signatures, a proposal hash, chaincode events, response status, namespace, a read set (list of key and version read by the transaction, etc.), a write set (list of key and value, etc.), a start key, an end key, a list of keys, a Merkel tree query summary, and the like. The transaction data may be stored for each of the N transactions.

[0172] In one embodiment of the instant solution, blockchain data **563** may include data for performing one or more of predicting, by a vehicle, a duration and a severity of an electrical-related event; determining, by the vehicle, a behavior profile of a group of devices at a location, based on a need of at least one of the group of devices during the event, and an amount of energy consumption of at least one of the group of devices during the event; predicting, by the vehicle, an energy requirement during the event at the location, based on the predicted duration, the predicted severity and the determined behavior profile; directing, by the vehicle, energy to be stored prior to the event by both the vehicle and an energy storage unit (ESU) associated with the location; and directing, by the vehicle, a flow of energy to the group of devices during the event, from the ESU and the vehicle.

[0173] Although in FIG. **5**F the blockchain data **563** is depicted in the block data **550** but may also be located in the block header **540** or the block metadata **560**.

[0174] The block metadata **560** may store multiple fields of metadata (e.g., as a byte array, etc.). Metadata fields may include signature on block creation, a reference to a last configuration block, a transaction filter identifying valid and invalid transactions within the block, last offset persisted of an ordering service that ordered the block, and the like. The signature, the last configuration block, and the orderer metadata may be added by the ordering service **510**E in FIG. **5**E. Meanwhile, a committer of the block (such as blockchain node **512**E in FIG. **5**E) may add validity/invalidity information based on an endorsement policy, verification of read/write sets, and the like. The transaction filter may include a byte array of a size equal to the number of transactions in the block data and a validation code identifying whether a transaction was valid/invalid.

[0175] The above embodiments may be implemented in hardware, in a computer program executed by a processor, in firmware, or in a combination of the above. A computer program may be embodied on a computer readable medium, such as a storage medium. For example, a computer program may reside in random access memory ("RAM"), flash memory, read-only memory ("ROM"), erasable programmable read-only memory ("EPROM"), electrically erasable programmable read-only memory ("EEPROM"), registers, hard disk, a removable disk, a compact disk read-only memory ("CD-ROM"), or any other form of storage medium known in the art.

[0176] An exemplary storage medium may be coupled to the processor such that the processor may read information from, and write information to, the storage medium. In the alternative, the storage medium may be integral to the processor. The processor and the storage medium may reside in an application-specific integrated circuit ("ASIC"). In the alternative, the processor and the storage medium may reside as discrete components. For example, FIG. **6** illustrates an example computer system architecture **600**, which may represent or be integrated in any of the above-described components, etc.

[0177] FIG. **6** illustrates a computing environment according to example embodiments. FIG. **6** is not intended to suggest any limitation as to the scope of use or functionality of embodiments of the application described herein. Regardless, the computing environment **600** can be implemented to perform any of the functionalities described herein. In computer environment **600**, computer system **601** is operational within numerous other general-purpose or special-purpose computing system environments or configurations.

[0178] Computer system **601** may take the form of a desktop computer, laptop computer, tablet computer, smartphone, smartwatch or other wearable computer, server computer system, thin client, thick client, network PC, minicomputer system, mainframe computer, quantum computer, and distributed cloud computing environment that include any of the described systems or devices, and the like or any other form of computer or mobile device now known or to be developed in the future that is capable of running a program, accessing a network **650** or querying a database. Depending upon the technology, the performance of a computer-implemented method may be distributed among multiple computers and between multiple locations. However, in this presentation of the computing environment **600**, a detailed discussion is focused on a single computer, specifically computer system **601**, to keep the presentation as simple as possible.

[0179] Computer system **601** may be located in a cloud, even though it is not shown in a cloud in FIG. **6**. On the other hand, computer system **601** is not required to be in a cloud except to any extent as may be affirmatively indicated. Computer system **601** may be described in the general context of computer system-executable instructions, such as program modules, executed by a computer system **601**. Generally, program modules may include routines, programs, objects, components, logic, data structures, and so on that perform tasks or implement certain abstract data types. As shown in FIG. **6**, computer system **601** in computing environment **600** is shown in the form of a general-purpose computing device. The components of computer system **601** may include but are not limited to, one or more processors or processing units **602**, a system memory **630**, and a bus **620** that couples various system components, including system memory **630** to processor **602**.

[0180] Processing unit **602** includes one or more computer processors of any type now known or to be developed. The processing unit **602** may contain circuitry distributed over multiple integrated circuit chips. The processing unit **602** may also implement multiple processor threads and multiple processor cores. Cache **632** is a memory that may be in the processor chip package(s) or located "off-chip," as depicted in FIG. **6**. Cache **632** is typically used for data or code that the threads or cores running on the processing unit **602** should be available for rapid access. In some computing environments, processing unit **602** may be designed to work with qubits and perform quantum computing.

[0181] Network adapter **603** enables the computer system **601** to connect and communicate with one or more networks **650**, such as a local area network (LAN), a wide area network (WAN), and/or a public network (e.g., the Internet). It bridges the computer's internal bus **620** and the external network, exchanging data efficiently and reliably. The network adapter **603** may include hardware, such as modems or Wi-Fi signal transceivers, and software for packetizing and/or de-packetizing data for communication network transmission. Network adapter **603** supports various communication protocols to ensure compatibility with network standards. For Ethernet connections, it adheres to protocols such as IEEE 802.3, while for wireless communications, it might support IEEE 802.11 standards, Bluetooth, near-field communication (NFC), or other network wireless radio standards.

[0182] Computer system **601** may include a removable/non-removable, volatile/non-volatile computer storage device **610**. By way of example only, storage device **610** can be a non-removable, non-volatile magnetic media (not shown and typically called a "hard drive"). One or more data interfaces can connect it to the bus **620**. In embodiments where computer system **601** is required to have a large amount of storage (for example, where computer system **601** locally stores and manages a large database), then this storage may be provided by peripheral storage devices **610** designed for storing very large amounts of data, such as a storage area network (SAN) that is shared by multiple, geographically distributed computers.

[0183] The operating system **611** is software that manages computer system **601** hardware resources and provides common services for computer programs. Operating system **611** may take several forms, such as various known proprietary operating systems or open-source Portable Operating System Interface type operating systems that employ a kernel.

[0184] The Bus **620** represents one or more of several types of bus structures, including a memory bus or memory controller, a peripheral bus, an accelerated graphics port, and a processor or local bus using various bus architectures. By way of example, and not limitation, such architectures include Industry Standard Architecture (ISA) buses, Micro Channel Architecture (MCA) buses, Enhanced ISA (EISA) buses, Video Electronics Standards Association (VESA) local buses, and Peripheral Component Interconnects (PCI) bus. The bus **620** is the signal conduction paths that allow the various components of computer system **601** to communicate with each other.

[0185] Memory **630** is any volatile memory now known or to be developed in the future. Examples include dynamic random-access memory (RAM **631**) or static type RAM **631**. Typically, the volatile memory is characterized by random access, but this is not required unless affirmatively indicated. In computer system **601**, memory **630** is in a single package and is internal to computer system **601**, but alternatively or additionally, the volatile memory may be distributed over multiple packages and/or located externally with respect to computer system **601**. By way of example only, memory **630** can be provided for reading from and writing to a non-removable, non-volatile magnetic media (shown as storage device **610**, and typically called a "hard drive"). Memory **630** may include at least one program product having a set (e.g., at least one) of program modules configured to carry out various functions. A typical computer system **601** may include cache **632**, a specialized volatile memory generally faster than RAM **631** and generally located closer to the processing unit **602**. Cache **632** stores frequently accessed data and instructions accessed by the processing unit **602** to speed up processing time. The computer system **601** may include non-volatile memory **633** in ROM, PROM, EEPROM, and flash memory. Non-volatile memory **633** often contains programming instructions for starting the computer, including the BIOS and information required to start the operating system **611**.

[0186] Computer system **601** may also communicate with one or more peripheral devices **641** via an I/O interface **640**. Such devices may include a keyboard, a pointing device, a display, etc.; one or more devices that enable a user to interact with computer system **601**; and/or any devices (e.g., network card, modem, etc.) that enable computer system **601** to communicate with one or more other computing devices. Such communication can occur via input/output (I/O) interfaces **640**. As depicted, IO interface **640** communicates with the other components of computer system **601** via bus **620**.

[0187] Network **650** is any computer network that can receive and/or transmit data. Network **650** can include a WAN, LAN, private cloud, or public Internet, capable of communicating computer data over non-local distances by any technology that is now known or to be developed in the future. Any connection depicted can be wired and/or wireless and may traverse other components that are not shown. In some embodiments, a network **650** may be replaced and/or supplemented by LANs designed to communicate data between devices located in a local area, such as a Wi-Fi network. The network **650** typically includes computer hardware such as copper transmission cables, optical transmission fibers, wireless transmission, routers, firewalls, switches, gateway computers, and edge servers. Computer system **601** connects to network **650** via network adapter **603** and bus **620**.

[0188] User devices **651** are any computer systems used and controlled by an end user in connection with computer system **601**. For example, in a hypothetical case where computer system **601** is designed to provide a recommendation to an end user, this recommendation may typically be communicated from network adapter **603** of computer system **601** through network **650** to a user device **651**, allowing user device **651** to display, or otherwise present, the recommendation to an end user. User devices can be a wide array of devices, including PCs, laptops, tablet, hand-held, mobile phones, etc.

[0189] Remote Servers **660** are any computers that serve at least some data and/or functionality over a network **650**, for example, WAN, a virtual private network (VPN), a private cloud, or via the Internet to computer system **601**. These networks **650** may communicate with a LAN to reach users. The user interface may include a web browser or an application that facilitates communication between the user and remote data. Such applications have been called "thin" desktops or "thin clients." Thin clients typically incorporate software programs to emulate desktop sessions, such as Microsoft RDP (Remote Desktop Protocol) or Citrix ICA (Independent Computing Architecture). Mobile applications can also be used. Remote servers **660** can also host remote databases **661**, with the database located on one remote server **660** or distributed across multiple remote servers **660**. Remote databases **661** are accessible from database client applications installed locally on the remote server **660**, other remote servers **660**, user devices **651**, or computer system **601** across a network **650**.

[0190] A Public Cloud **670** is an on-demand availability of computer system resources, including data storage and computing power, without direct active management by the user. Public clouds **670** are often distributed, with data centers in multiple locations for availability and performance. Computing resources on public clouds (**670**) are shared across multiple tenants through virtual computing environments comprising virtual machines **671**, databases **672**, containers **673**, and other resources. A Container **673** is an isolated, lightweight software for running an application on the host operating system **611**. Containers **673** are built on top of the host operating system's kernel and contain only apps and some lightweight operating system APIs and services. In contrast, virtual machine **671** is a software layer that includes a complete operating system **611** and kernel. Virtual machines **671** are built on top of a hypervisor emulation layer designed to abstract a host computer's hardware from the operating software environment. Public clouds **670** generally offer hosted databases **672** abstracting high-level database management activities. It should be further understood that one or more of the elements described or depicted in FIG. **6** can perform one or more of the actions, functionalities, or features described or depicted herein.

[0191] Although an exemplary embodiment of at least one of a system, method, and non-transitory computer readable medium has been illustrated in the accompanied drawings and described in the foregoing detailed description, it will be understood that the application is not limited to the embodiments disclosed, but is capable of numerous rearrangements, modifications, and substitutions as set forth and defined by the following claims. For example, the capabilities of the system of the various figures can be performed by one or more of the modules or components described herein or in a distributed architecture and may include a transmitter, receiver, or pair of both. For example, all or part of the functionality performed by the individual modules, may be performed by one or more of these modules. Further, the functionality described herein may be performed at various times and in relation to various events, internal or external to the modules or components. Also, the information sent between various modules can be sent between the modules via at least one of: a data network, the Internet, a voice network, an Internet Protocol network, a wireless device, a wired device and/or via plurality of protocols. Also, the messages sent or received by any of the modules may be sent or received directly and/or via one or more of the other modules.

[0192] One skilled in the art will appreciate that a "system" may be embodied as a personal computer, a server, a console, a personal digital assistant (PDA), a cell phone, a tablet computing device, a smartphone or any other suitable computing device, or combination of devices. Presenting the above-described functions as being performed by a "system" is not intended to limit the scope of the present application in any way but is intended to provide one example of many embodiments. Indeed, methods, systems and apparatuses disclosed herein may be implemented in localized and distributed forms consistent with computing technology.

[0193] It should be noted that some of the system features described in this specification have been presented as modules to more particularly emphasize their implementation independence. For example, a module may be implemented as a hardware circuit comprising custom very-large-scale integration (VLSI) circuits or gate arrays, off-the-shelf semiconductors such as logic chips, transistors, or other discrete components. A module may also be implemented in programmable hardware devices such as field-programmable gate arrays, programmable array logic, programmable logic devices, graphics processing units, or the like.

[0194] A module may also be at least partially implemented in software for execution by various types of processors. An identified unit of executable code may, for instance, comprise one or more physical or logical blocks of computer instructions that may, for instance, be organized as an object, procedure, or function. Nevertheless, the executables of an identified module need not be physically located together but may comprise disparate instructions stored in different locations that, when joined logically together, comprise the module and achieve the stated purpose for the module. Further, modules may be stored on a computer-readable medium, which may be, for instance, a hard disk drive, flash device, random access memory (RAM), tape, or any other such medium used to store data.

[0195] Indeed, a module of executable code may be a single instruction, or many instructions, and may even be distributed over several different code segments, among different programs, and across several memory devices. Similarly, operational data may be identified and illustrated herein within modules and may be embodied in any suitable form and organized within any suitable type of data structure. The operational data may be collected as a single data set or may be distributed over different locations, including over different storage devices, and may exist, at least partially, merely as electronic signals on a system or network.

[0196] It will be readily understood that the components of the application, as generally described and illustrated in the figures herein, may be arranged and designed in a wide variety of different configurations. Thus, the detailed description of the embodiments is not intended to limit the scope of the application as claimed but is merely representative of selected embodiments of the application.

[0197] One having ordinary skill in the art will readily understand that the above may be practiced with steps in a different order and/or with hardware elements in configurations that are different from those which are disclosed. Therefore, although the application has been described based upon these preferred embodiments, it would be apparent to those of skill in the art that certain modifications, variations, and alternative constructions would be apparent.

[0198] While preferred embodiments of the present application have been described, it is to be understood that the embodiments described are illustrative only and the scope of the application is to be defined solely by the appended claims when considered with a full range of equivalents and modifications (e.g., protocols, hardware devices, software platforms etc.) thereto.

What is claimed is:

1. A method, comprising:

predicting, by a vehicle, a duration and a severity of an electrical-related event;

determining, by the vehicle, a behavior profile of a group of devices at a location, based on a need of at least one of the group of devices during the event, and an amount of energy consumption of at least one of the group of devices during the event;

predicting, by the vehicle, an energy requirement during the event at the location, based on the predicted duration, the predicted severity and the determined behavior profile;

directing, by the vehicle, energy to be stored prior to the event by both the vehicle and an energy storage unit (ESU) associated with the location; and

directing, by the vehicle, a flow of energy to the group of devices during the event, from the ESU and the vehicle.

2. The method of claim **1**, wherein the directing, by the vehicle, the flow of energy to the group of devices during the event is performed by first directing energy from a battery of the vehicle to the group of devices.

3. The method of claim **1**, comprising:

determining, by the vehicle, a first amount of time that the ESU provides energy to the group of devices, based on the determined behavior profile; and

determining, by the vehicle, an additional charge to be added to a battery of the vehicle, based on the first amount of time.

4. The method of claim **1**, comprising:

gathering the amount of energy consumption for the group of devices over a period of time; and

storing the gathered amount of energy consumption in the behavior profile.

5. The method of claim **1**, wherein the directing, by the vehicle, the flow of energy to the group of devices during the event is performed by:

directing energy from a battery of the vehicle to the group of devices;

monitoring a state-of-charge of the battery; and

in response to the state-of-charge dropping below a threshold, directing energy from the ESU to the group of devices.

6. The method of claim **1**, wherein the directing, by the vehicle, the flow of energy to the group of devices during the event is performed by:

directing energy from a battery of the vehicle to the group of devices;

determining a minimum state-of-charge of the battery required for the vehicle to transport an occupant from the location to another location during the event;

monitoring a state-of-charge of the battery; and

in response to the state-of-charge approaching the minimum state-of-charge, directing energy from the ESU to the group of devices.

7. The method of claim **1**, comprising:

determining that at least one device of the group of devices is a medical device associated with an occupant of the location;

determining an amount of energy required to power the medical device based on the predicted duration and the predicted severity of the event; and

reserving at least the amount of energy on at least one of the ESU or the vehicle for powering the medical device.

8. A system, comprising:

a processor; and

a memory, wherein the processor and the memory are communicably coupled, wherein the processor:

predicts, at a vehicle, a duration and a severity of an electrical-related event;

determines, at the vehicle, a behavior profile of a group of devices at a location, based on a need of at least one of the group of devices contemporaneously with the event, and an amount of energy consumption of at least one of the group of devices contemporaneously with the event;

predicts, at the vehicle, an energy requirement contemporaneously with the event at the location, based on the predicted duration, the predicted severity and the determined behavior profile;

directs energy to be stored prior to the event by both the vehicle and an energy storage unit (ESU) associated with the location; and

directs a flow of energy to the group of devices contemporaneously with the event, from the ESU and the vehicle.

9. The system of claim **8**, wherein the directs the flow of energy to the group of devices contemporaneously with the event, first directs energy from a battery of the vehicle to the group of devices.

10. The system of claim **8**, wherein the processor:

determines, at the vehicle, a first amount of time that the ESU provides energy to the group of devices, based on the determined behavior profile; and

determines, at the vehicle, an additional charge to be added to a battery of the vehicle, based on the first amount of time.

11. The system of claim **8**, wherein the processor:

gathers the amount of energy consumption for the group of devices over a period of time; and

stores the gathered amount of energy consumption in the behavior profile.

12. The system of claim **8**, wherein the directs the flow of energy to the group of devices contemporaneously with the event comprises:

directs energy from a battery of the vehicle to the group of devices;

monitors a state-of-charge of the battery; and

in response to the state-of-charge drops below a threshold, directs energy from the ESU to the group of devices.

13. The system of claim **8**, wherein the processor:

directs energy from a battery of the vehicle to the group of devices;

determines a minimum state-of-charge of the battery required for the vehicle to transport an occupant from the location to another location during the event;

monitors a state-of-charge of the battery; and

in response to the state-of-charge approaches the minimum state-of-charge, directs energy from the ESU to the group of devices.

14. The system of claim **8**, wherein the processor:

determines that at least one device of the group of devices is a medical device associated with an occupant of the location;

determines an amount of energy required to power the medical device based on the predicted duration and the predicted severity of the event; and

reserves at least the amount of energy on at least one of the ESU or the vehicle for powering the medical device.

15. A computer-readable storage medium comprising instructions that, when read by a processor, cause the processor to perform:

predicting, by a vehicle, a duration and a severity of an electrical-related event;

determining, by the vehicle, a behavior profile of a group of devices at a location, based on a need of at least one of the group of devices during the event, and an amount of energy consumption of at least one of the group of devices during the event;

predicting, by the vehicle, an energy requirement during the event at the location, based on the predicted duration, the predicted severity and the determined behavior profile;

directing, by the vehicle, energy to be stored prior to the event by both the vehicle and an energy storage unit (ESU) associated with the location; and

directing, by the vehicle, a flow of energy to the group of devices during the event, from the ESU and the vehicle.

16. The computer-readable storage medium of claim **15**, wherein the directing, by the vehicle, the flow of energy to the group of devices during the event is performed by first directing energy from a battery of the vehicle to the group of devices.

17. The computer-readable storage medium of claim **15**, further comprising instructions for:

determining, by the vehicle, a first amount of time that the ESU provides energy to the group of devices, based on the determined behavior profile; and

determining, by the vehicle, an additional charge to be added to a battery of the vehicle, based on the first amount of time.

18. The computer-readable storage medium of claim **15**, further comprising instructions for:

gathering the amount of energy consumption for the group of devices over a period of time; and

storing the gathered amount of energy consumption in the behavior profile.

19. The computer-readable storage medium of claim **15**, wherein the directing, by the vehicle, the flow of energy to the group of devices during the event is performed by:

directing energy from a battery of the vehicle to the group of devices;

monitoring a state-of-charge of the battery; and

in response to the state-of-charge dropping below a threshold, directing energy from the ESU to the group of devices.

20. The computer-readable storage medium of claim **15**, wherein the directing, by the vehicle, the flow of energy to the group of devices during the event is performed by:

directing energy from a battery of the vehicle to the group of devices;

determining a minimum state-of-charge of the battery required for the vehicle to transport an occupant from the location to another location during the event;

monitoring a state-of-charge of the battery; and

in response to the state-of-charge approaching the minimum state-of-charge, directing energy from the ESU to the group of devices.

* * * * *