United States Patent Application Publication 20250256604
Kind Code A1
Publication Date August 14, 2025
Inventor(s) Wilder; James D. et al.

DELIVERING A GOOD OR SERVICE TO A CHARGING VEHICLE

Abstract

An example operation includes one or more of determining a charging window to charge an electric vehicle (EV) battery of a vehicle from a first charge level to a second charge level, determining a delivery window during which an item will be delivered to the vehicle based on the charging window, and dispatching a delivery of the item to the vehicle.

Inventors: **Wilder; James D. (Dallas, TX), Swartz; James G. (Prosper, TX)**

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

Family ID: **96661637**

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

Appl. No.: **18/439762**

Filed: **February 12, 2024**

Publication Classification

Int. Cl.: **B60L53/62** (20190101); **B60L53/66** (20190101)

U.S. Cl.:

CPC **B60L53/62** (20190201); **B60L53/66** (20190201);

Background/Summary

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 determining a charging window to charge an electric vehicle (EV) battery of a vehicle from a first charge level to a second charge level, determining a delivery window during which an item will be delivered to the vehicle based on the charging window, and dispatching a delivery of the item to 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 determine a charging window to charge an electric vehicle (EV) battery of a vehicle from a first charge level to a second charge level, determine a delivery window during which an item will be delivered to the vehicle based on the charging window, and dispatch a delivery of the item to the vehicle.

[0004] Another example embodiment includes a non-transitory computer readable storage medium configured to store instructions that when executed cause a processor to perform determining a charging window to charge an electric vehicle (EV) battery of a vehicle from a first charge level to a second charge level, determining a delivery window during which an item will be delivered to the vehicle based on the charging window, and dispatching a delivery of the item to the vehicle.

Description

BRIEF DESCRIPTION OF THE DRAWINGS

[0005] FIG. **1**A illustrates an example system diagram of a vehicle charging at a charging station and the process of providing a delivery being managed by a server, according to example embodiments.

[0006] FIG. **1**B illustrates a network diagram of a vehicle using a charging station and a delivery vehicle being scheduled to perform a delivery to the vehicle, 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.

[0044] 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.

[0045] 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.

[0046] 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.

[0047] 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.

[0048] 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.

[0049] 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.

[0050] FIG. **1**A illustrates an example system diagram of a vehicle charging at a charging station and the process of providing a delivery being managed by a server, according to example embodiments. Referring to FIG. **1**A, the system **100** includes a vehicle **130** being identified as a candidate to initiate charging at a charging station **152** or charging point, hereinafter used interchangeably. A server **160** may be a remote computer in a cloud network or a local computer operating near the vehicle **130**, such as a mobile device or other computing device associated with the vehicle **130**. Once a vehicle **130** is identified as a candidate for power charging (e.g., electrical charging), the server **160** may assign the vehicle to a particular charging station **152** and the vehicle **130** may begin receiving charge **112**. The vehicle **130** may then communicate with the charging station **152** to initiate the charge process **114**. An initial vehicle charge level may quickly change to an elevated/higher/greater amount of charge which is continuously monitored by the server **160**. The amount of time, the updated charge level at any particular time, and other charge attributes may be monitored by the server **160** in an ongoing basis to determine a status of the vehicle **130**.

[0051] A prediction or estimation may be made by the server **160** based on known data as to how long the vehicle should charge or will need to charge to reach an optimal charge level (e.g., 60 percent, 70 percent, 80 percent or higher). The estimated charge time may be determined **116** based on a type of vehicle, the type of charging station, previous charge durations on a per vehicle basis, etc. All such information may be used as the basis for predicting how long the vehicle **130** should remain charging at the charge station **152**.

[0052] In the event that another vehicle or vehicles are registered and identified as being in a queue maintained by the server **160**, certain considerations are initiated regarding the status of the vehicle **130** that is actively charging, such as what charge level did the vehicle have before charging, what charge level does the vehicle currently have, how long has the vehicle been charging, what is the destination of the vehicle, etc. The vehicle charging station **152** and/or the vehicle **130** may determine one or more charging thresholds have been reached and incentives should be offered at those threshold maturity times to persuade vehicle operators to leave their current charging stations so queued vehicles can begin charging. The incentive items may be sent to the vehicle to offer the vehicle an opportunity to accept the items and leave the charging station. The incentives may be sent to the display of the vehicle, a user's mobile device or other notification platforms.

[0053] While a vehicle **130** is charging, certain affiliations **118** may be identified from a user and/or vehicle profile associated with the vehicle **130**. Examples may include subscription services, a current status of orders which require delivery, all of which may be identified to route those orders to the vehicle during charging. In one example, the charging station **152** may be in a busier part of town than the user's home address, and it may be prudent to consider this temporary location as a delivery location to reduce the number of miles the delivery vehicle may normally have to drive to reach the user's home. The affiliations **118** identified by the server **160** may yield a list of locations and/or services which are included in a user's profile as pending services, such as pending deliveries, pending subscriptions, etc. Any one of the affiliations with a pending status may be dispatched to include an order or service during the estimated time window for charging the vehicle. Once an affiliation is identified, an item or service may be selected to provide to the vehicle. This may include one, two, or more items, which can include physical items, such as meal delivery, parcel delivery and non-tangible items, such as streaming movies, music, etc. One or more items may be selected **122** to be delivered to the vehicle **130** while it is charging at the charging station **152** based on the user's affiliation profile accessible by the user's mobile device and/or the transport computer. A review of the charge time remaining at that point in time may be performed **124** to determine whether the delivery time of a selected item is less than the charge time. When enough time is available to perform the delivery, a dispatch operation may occur 126 to provide the item to the vehicle **130**. A user profile may have a preference setup so that only computer-based content is sent to the vehicle instead of deliveries, or that all possible deliveries are sent to a charging station at any given time for that particular vehicle.

[0054] FIG. **1**B illustrates a network diagram of a vehicle using a charging station and a delivery vehicle being scheduled to perform a delivery to the vehicle, according to example embodiments.

Referring to FIG. 1B, the network **150** includes a vehicle management server **160** which may maintain a status of the vehicle **130** which is having a charge cycle at a charging point or charging station **152**. The currently charging vehicle **130** may have an initial charge level, such as 10 percent or a low amount of charge respective to a full charge level of 100 percent. Any vehicle that is attempting to charge below an initial minimum charge level, such as 30 percent may be charged more per unit of charge received since their vehicle may not be considered charge deficient at that particular time for being over the minimum charge level. The vehicle **130** may receive a consistent charge rate to reach a saturation point, such as 80 percent, where the charge rate over time begins to slow and the other vehicles should be offered charge at a sooner time to avoid overuse of the charging station **152**. If the charge is taking place during a peak use period, such as 7 am to 10 am or 3 μm to 6 μm, the saturation point of 80 percent may be adjusted to include a preferred exit point, such as 60 percent to encourage those on the charging station to leave in favor of other vehicles due to heavy usage in the area. After the peak use time is over, the preferred exit point of 60 percent may be removed from consideration and the target may resume to 80 percent.

[0055] In this example, the vehicle **130** may begin charging at the charging station **152** and an estimated amount of time may be determined **162** by the vehicle management server **160** or another computing device to identify whether the time remaining until the vehicle **130** exits the charging station **152** is a long enough period of time to receive a delivery of an item from a delivery vehicle **132** which will take time to deliver. The moment the vehicle is linked with the charging station **152**, a timer may begin to measure a total time estimated for the vehicle **130** to complete the charge. Another approach may be to schedule a package delivery during a time window that coincides with when the vehicle is registered to be present at the charging station **152**. That way, the planning and delivery vehicle **132** has more knowledge of when to deliver the item at an earlier time which permits more efficient time for planning deliveries.

[0056] The scheduling of deliveries is based on the estimated charging time for a vehicle to charge to reach a determined charge level. The downtime experienced by a driver or passengers of the vehicle when charging, provides an opportunity for items and service delivery to that particular vehicle. The charging may cause a location snapshot to be taken that coincides with the vehicle. The time estimate can then provide a location and a period of time estimated for the vehicle to be at that location. The system application may use predictive analytics to assess the vehicle's various needs and available time window. This may include receiving a package addressed to one or more occupants, facilitating maintenance, repairs, and shopping order fulfillment.

[0057] In one example, the system application may utilize the charging time for delivering content to the occupant(s). Recognizing that vehicle owners may charge 80% or more of their battery capacity which can take a significant amount of time, the system application capitalizes on this available time duration by providing the occupant(s) with entertainment and/or informational data. Examples may include streaming media, news updates, and/or interactive content tailored to the estimated duration of the charging session. The latest news may also be delivered via video, audio, and/or text, depending on the user's preferences and the content may be premium and licensed to the charge service. Also, games, quizzes, educational content, or even guided meditation sessions may be offered to engage the users interactively. The system application may provide a selection of the content that users can consume during the time it takes to charge their vehicle. For instance, if a charging session is expected to take 30 minutes, the system application may suggest content that can be completed in that identified time duration, so the user has a satisfying start-to-finish experience at the station. The content delivery may be automated and personalized as soon as the charging session begins. Over time, the system application may learn the user's content preferences and suggest personalized options for each charging session which are different from a previous session. The content offerings may also be based on the time of day, location, and/or charging speed, and may be dynamically adjusted so the options are different if the charging time is shorter or longer than initially expected.

[0058] In one example, the delivery of goods can be coordinated to the user while their vehicle is being charged. The location of a charging point may be stored as a temporary delivery address for the owner of the vehicle during the charging period to offer a way to manage deliveries more efficiently. Various delivery services may have an accessible database of deliveries intended for a particular user profile that is the same or similar to the one user profile of the vehicle, which may include postal services, courier companies, food delivery services, and even local businesses that offer delivery of their products. The application may review the available data once the charging is initiated to attempt the scheduling of packages to coincide with the user's charging time at the charging point. The trigger to review user planned activities (e.g., product delivery, etc.) may initiate once the user has scheduled a charging time and/or location, the delivery time will then be used for the future delivery of goods that match a time frame known for the user. The delivery status may include a real-time communication interface with the vehicle computer and/or a user's mobile device to receive delivery services. The status information may provide updates on delivery timing, allowing for dynamic adjustment, for example, when the charging is completed sooner or is taking longer than expected.
[0059] In one example, the system application may coordinate certain services, such as vehicle washes and repairs while parked at the charging station. The scheduling of these services may be based on the expected duration of the vehicle's stay, communicated by the system application's ability to predict how long the vehicle will be charging (e.g., 38 minutes from 20% to 80% battery capacity). The prediction may assist with scheduling of certain ancillary services received within that window. Using a combination of predictive analytics to estimate charging times and a scheduling application for creating and confirming service appointments, the vehicle owner may be able to optimize the use of their time while charging.
[0060] In one example, the system application uses predictive capabilities for determining the vehicle's accessibility window for various services or goods being delivered. The system application operating on the server **160** can predict the available time by assessing factors such as the vehicle's current state, required maintenance, such as fluid levels, tire pressure, or headlight functionality, etc., and the user's consumption of goods, such as videos watched recently, recent food purchases, etc. The application can then coordinate with service providers to secure appointments for those items within the timeframe allotted, and by presenting users with options and associated costs for services.
[0061] In one example, the system application can offer additional services or goods by extending an estimated time frame by adding more time and reducing a particular charge rate that may otherwise increase after the vehicle has reached a saturation point of 80 percent charge or a peak charging time has started. Those charging concerns can be suspended for an extended time, such as 5 or 10 minutes to ensure the service is delivered prior to departure time.
[0062] While in a charging status, the user may be able to leave the vehicle and participate in a nearby activity, such as entertainment or dining. An autonomous vehicle, such as a taxi car, scooter or other transportation item may be dispatched to visit the vehicle **130** and collect the user and take them on a tour of the area or to a nearby restaurant during the charging period. The vehicle may deliver food or other items prior to having the user enter the autonomous vehicle. The schedule of time and activities may be performed autonomously based on the predicted charging time window of the vehicle.
[0063] One example may include determining a charging window to charge an electric vehicle (EV) battery of a vehicle from a first charge level to a second charge level and determining a delivery window during which an item will be delivered to the vehicle based on the charging window and dispatching a delivery of the item to the vehicle. The initial charge level may be used as a basis for determining whether the dispatch should be performed. If the charge level is closer to the second charge level (i.e., the predicted final charge threshold), the dispatch may not be performed. If the initial charge level is closer to the first charge level, such as the starting charge

level, then the dispatch of the item may be performed. The process may also include determining a type of a charging point used to charge the vehicle and a type of the vehicle, and the charging window may be based on the type of the charging point and the type of the vehicle, and the delivery window can then be predicted based on the type of the charging point and the type of the vehicle. Certain vehicles require certain time periods to charge while others require less time.

[0064] The process may also include determining whether the item is consumable by a vehicle occupant of the vehicle during the charging window, and the item is one or more of content and a consumable product. When the item is food or other perishables or consumables, the application may identify whether the user has a preference to receive food at the charging station at a particular time frame and then dispatch the order and delivery to accommodate the user's preference stored in the user's profile to receive such products while charging. The process may also include subtracting an amount of time for the delivery (i.e., estimated) from the charging window to identify a remaining time window, determining the item may be consumed during the remaining time window, and determining the dispatching of the delivery is acceptable when the time is sufficient.

[0065] The process may also include determining a location of the item, determining a status of a delivery vehicle associated with the item, and determining whether to perform the dispatching of the delivery of the item to the vehicle based on the location, the status and the delivery window being shorter than the charging window. When a delivery vehicle has too many deliveries or is too far away, the delivery to the charging station may be cancelled or not confirmed. The process may also include determining to perform the dispatching of the item when the delivery window will be completed prior to the vehicle achieving the second charge level. Alternatively, when determining a first charge level of the vehicle is above a threshold first charge level, then the delivery may be cancelled in this case where the time duration will not be long enough to include the delivery since the vehicle does not need much more charge.

[0066] In one embodiment, the instant solution presents a way for electric vehicle (EV) owners to combine their charging sessions with their grocery shopping needs. The platform can be accessed via a mobile application or an in-vehicle infotainment system. When an EV owner schedules a charging session through the application, they also gain access to a list of participating grocery stores within proximity to the selected charging station. After choosing a grocery store, the user can then place an order for their groceries through the same platform. The system calculates the estimated charging time based on factors like the vehicle's current battery level, the charger's capacity, and the desired charge level. This calculated time window is communicated to the grocery store as the ideal timeframe for order preparation. The grocery store prepares the order, ensuring it's ready for pickup within the EV's charging window. The user receives real-time updates on their order status and the charging progress. If there are any delays or changes in the charging time (for instance, if the user decides to charge more or less than initially planned), the system dynamically adjusts the pickup time, communicating these changes to both the user and the grocery store. As the charging session nears completion, the user is notified that their grocery order is ready. The system can provide directions to the specific pickup area within the grocery store, optimized to be conveniently located near the charging station. The user can either choose a drive-through pickup, where groceries are loaded into their vehicle, or a quick walk-in pickup if the station is within walking distance. The entire transaction, including the EV charging fee and the grocery bill, can be handled seamlessly through the integrated platform. Users have the option to save their payment details for quick and easy processing in future transactions.

[0067] In one embodiment, the instant solution integrates EV charging stations with local marketplaces. The system creates a platform connected to charging stations where local vendors can list their products. While the vehicle is charging, the EV owner can browse and purchase items from these local vendors. The system predicts the required charging time and suggests products that can be prepared and delivered to the charging station within this window. The platform integrates with EV charging station networks. Local businesses, artisans, and vendors can register on the

platform to showcase their products and services. This can range from handmade crafts and locally produced foods to services like haircuts or quick repair jobs. The platform is accessible via a mobile application or an in-vehicle infotainment system, allowing EV owners to browse offerings while their vehicle charges. When an EV owner plugs in their vehicle at a charging station, they are automatically given access to the local marketplace through the charging station's interface. The list of local vendors, services, and products is tailored to the specific location of the charging station. The platform can offer a variety of filtering options, such as type of product, price range, and estimated preparation/delivery time. Users can place orders directly from their vehicle or mobile device. The system calculates the estimated charging time, which is then used to inform the user about how long they have to shop or receive services. For products, vendors receive notification of orders and can prepare items for pickup or delivery to the charging station within the customer's charging window. For services, appointments can be scheduled to align with the charging time. The platform facilitates real-time communication between the EV owner, the charging station, and the local vendors. If there are any changes in the charging time (due to a change in the user's plan or unforeseen circumstances at the charging station), the system can automatically notify the vendors to adjust preparation or delivery times accordingly. Depending on the nature of the product or service, customers can choose to have items delivered to them at the charging station or may visit the vendor if nearby. The system provides directions, estimated walking times, and even coordinates with local transport options if needed. The integrated platform can handle all aspects of the transaction, from ordering to payment. Users can pay for their goods, services, and charging fees all in one transaction through the application or in-vehicle system.
[0068] In one embodiment, the system offers healthcare and wellness services to EV owners during their charging sessions. Services, including mobile health check-ups, quick dental cleanings, and short wellness treatments like massages and acupuncture, can be scheduled to coincide with the vehicle's charging time. The system assesses the expected duration of the charging session and matches it with the duration of these services, offering a convenient way for busy individuals to take care of their health and well-being. The system utilizes a network that connects EV charging stations with local healthcare and wellness service providers. The network includes a diverse range of services, such as mobile medical units, wellness practitioners, and mental health professionals. The service network is accessible to EV users through a dedicated application or an in-vehicle infotainment system, enabling them to browse and book services during their charging session. Once an EV owner initiates a charging session, the system calculates the estimated duration based on the vehicle's battery level and the charging station's capabilities. The application then presents a list of available healthcare and wellness services that fit within the charging timeframe. For example, if the charging session is expected to last 30 minutes, the application may suggest a 20-minute massage, a quick dental check-up, or a brief mental health counseling session. Users book the desired service through the app, which in turn notifies the respective service provider. The providers, equipped with mobile setups, can either offer services at dedicated spaces within or near the charging station or provide in-car services where feasible. The system ensures real-time communication between the user, the service provider, and the charging station to coordinate service delivery within the charging window. The platform stores user preferences and health data (with consent), allowing it to offer personalized service recommendations in future sessions. For instance, if a user regularly opts for physiotherapy or prefers certain types of relaxation techniques, the system can prioritize these options. Additionally, regular health checks or follow-up sessions can be conveniently scheduled to align with the user's EV charging routine. In cases where the charging time is shorter or longer than expected, the system dynamically adjusts the service timing. If a charging session ends earlier, the service provider is notified to expedite the session, or alternatively, if the user decides to extend the charging time, they can opt for extended or additional services. The system handles payments for both the charging and the health/wellness services, streamlining the transaction process. Additionally, the system integrates with health insurance

providers to manage billing or claim processes directly, adding an extra layer of convenience.

[0069] In one embodiment, the instant solution allows EV charging stations to partner with educational providers to offer short, informative sessions or workshops that align with the charging window. These can be language lessons, tech workshops, and even art classes offered in a nearby facility or via an online platform accessible from the vehicle. The system schedules the sessions based on the anticipated charging duration, offering a productive way for individuals to spend their time while waiting for their EV to charge. The system's digital platform links EV charging stations with educational content providers. The providers range from local community colleges and universities to online course platforms, language institutes, and skill development centers. The platform is accessible through an application or the vehicle's infotainment system, allowing EV drivers to easily access a wide array of educational content. When an EV owner plugs in their vehicle at a charging station, the system calculates the estimated charging time. This information is used to tailor a selection of educational sessions or courses that can be completed within that time frame. For instance, if the charging session is expected to last 45 minutes, the system might suggest a 30-minute language lesson, a short seminar on a current topic, or a module from a longer course. The educational content can be delivered in various formats to suit different learning preferences and scenarios, including video lectures, interactive tutorials, podcasts, or even virtual reality (VR) and augmented reality (AR) experiences for more immersive learning. The system can also facilitate live sessions, such as webinars or virtual workshops, scheduled to coincide with popular charging times. Users browse and select sessions directly from their vehicle. If there are changes in the charging time (due to a decision to charge more or less), the system adapts by recommending shorter or longer sessions or by saving the user's progress in a session to be continued during the next charge. The platform tracks users' learning progress and preferences, offering personalized recommendations for future sessions. For courses that are part of a larger certification or skill development program, the system integrates with official certification bodies or educational institutions, allowing users to earn credits or certifications over time, making use of their regular EV charging schedule. The platform offers various payment models, from pay-per-session to subscription plans, providing flexibility and access to a wider user base. Charging stations can host live events and discussion groups, encouraging interaction and knowledge sharing among EV users.

[0070] In one embodiment, the system integrates vehicle maintenance and repair services with the EV charging process. The core of this system is a sophisticated platform that connects EV charging stations with a network of vehicle maintenance and repair service providers. The network can include various services such as tire rotation, oil changes, battery health checks, and minor repairs. The platform is accessible via an in-vehicle infotainment system or a dedicated mobile application, enabling EV owners to access these services conveniently. Upon connecting to a charging station, the vehicle's onboard diagnostics system automatically performs a series of checks to determine maintenance needs, including checking fluid levels, tire pressure, brake health, battery condition, and other standard vehicle maintenance aspects. The diagnostic results are then transmitted to the platform. Based on the diagnostics, the system suggests relevant maintenance services that can be completed within the estimated charging time. For example, if a vehicle will be charging for an hour, the system might suggest tire rotation and wiper blade replacement. The user can review the suggestions and approve the desired services. Once services are selected, the system coordinates with nearby service providers who are equipped to perform the tasks. The providers can either be stationed at or near the charging station or have mobile units capable of reaching the vehicle at its charging location. Service providers receive notifications and specifics about the required services and the estimated time available to complete them. They then perform the maintenance tasks while the vehicle is charging, ensuring efficient use of time. The system provides real-time updates to the vehicle owner on the progress of the maintenance work. When a service task is taking longer than the charging time, the system notifies the owner and offers the option to extend the charging session or to pause and reschedule the remaining service for another time. The system handles payments for both the EV charging and the maintenance services, simplifying the transaction process for the user. The system also supports subscription models offering regular maintenance packages, which include periodic check-ups and services aligned with the owner's charging schedule. After the completion of services, the platform facilitates a quality check and feedback mechanism, allowing vehicle owners to rate the service received, ensuring high service standards and continual improvement of the system based on user feedback.

[0071] In one embodiment, the system coordinates the charging of an electric vehicle (EV) battery and the delivery of an item to the vehicle. The system comprises a charging management module, a delivery management module, and a communication interface for facilitating interactions between these components. The charging management module determines the charging window to charge the EV battery from its current charge level to a specified second charge level. The module utilizes sensors and data sources to assess the battery's state of charge, energy consumption patterns, and charging infrastructure availability. It communicates with the EV's onboard charging system and external charging stations to establish the optimal charging schedule. The communication interface ensures seamless data exchange between the charging management module and relevant stakeholders, such as the EV owner and charging service providers. Simultaneously, the delivery management module determines a delivery window during which an item will be delivered to the EV based on the calculated charging window. The module uses factors like the item's size, delivery distance, and traffic conditions to optimize the delivery timing. Through the communication interface, the delivery management module communicates with delivery personnel, tracking systems, and the item sender to coordinate the logistics of the delivery. Furthermore, the system dispatches the delivery of the item to the vehicle. This involves a coordinated effort between the charging and delivery management modules. Once the delivery window aligns with the charging schedule, the item is dispatched to ensure it arrives at the vehicle during the predetermined window.

[0072] Flow diagrams depicted herein, such as FIG. **1**A, 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.

[0073] It is important to note that all the flow diagrams and corresponding processes derived from FIG. **1**A, 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.

[0074] 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.

[0075] 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.

[0076] 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.

[0077] 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.

[0078] 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 non-transitory 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.

[0079] The processor **204** performs one or more of determining a charging window to charge an electric vehicle (EV) battery of a vehicle from a first charge level to a second charge level **244**C, determining a delivery window during which an item will be delivered to the vehicle based on the charging window **246**C, and dispatching a delivery of the item to the vehicle **248**C.

[0080] 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 non-transitory 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.

[0081] The processor **204** performs one or more of determining a type of a charging point used to charge the vehicle and a type of the vehicle, and wherein the charging window is based on the type of the charging point and the type of the vehicle, and predicting the delivery window based on the type of the charging point and the type of the vehicle **244**D, determining whether the item is consumable by a vehicle occupant of the vehicle during the charging window, and wherein the item is one or more of content and a consumable product **245**D, subtracting an amount of time for the delivery from the charging window to identify a remaining time window, determining the item may be consumed during the remaining time window, and determining the dispatching of the delivery is acceptable **246**D. The process may also include determining a location of the item, determining a status of a delivery vehicle associated with the item, and determining whether to perform the dispatching of the delivery of the item to the vehicle based on the location, the status and the delivery window being shorter than the charging window **247**D, and determining to perform the dispatching when the delivery window will be completed prior to the vehicle achieving the second charge level **248**D. The process may also include determining the first charge level of the vehicle is above a threshold first charge level, and cancelling the delivery **249**D.

[0082] 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.

[0083] 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.

[0084] 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.

[0085] 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 determining a charging window to charge an electric vehicle (EV) battery of a vehicle from a first charge level to a second charge level **244**E, determining a delivery window during which an item will be delivered to the vehicle based on the charging window **246**E, and dispatching a delivery of the item to the vehicle **248**E.

[0086] 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 determining a type of a charging point used to charge the vehicle and a type of the vehicle, and wherein the charging window is based on the type of the charging point and the type of the vehicle, and predicting the delivery window based on the type of the charging point and the type of the vehicle **244**F. The process may also include determining whether the item is consumable by a vehicle occupant of the vehicle during the charging window, and wherein the item is one or more of content and a consumable product **245**F, subtracting an amount of time for the delivery from the charging window to identify a remaining time window, determining the item may be consumed during the remaining time window, and determining the dispatching of the delivery is acceptable **246**F, determining a location of the item, determining a status of a delivery vehicle associated with the item, and determining whether to perform the dispatching of the delivery of the item to the vehicle based on the location, the status and the delivery window being shorter than the charging window **247**F. The process may also include determining to perform the dispatching when the delivery window will be completed prior to the vehicle achieving the second charge level **248**F and determining the first charge level of the vehicle is above a threshold first charge level, and cancelling the delivery **249**F.

[0087] 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, Al 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.

[0088] 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.

[0089] 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.

[0090] 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.

[0091] 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.

[0092] 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.

[0093] 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.

[0094] 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**.

[0095] 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**.

[0096] 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.

[0097] 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**.

[0098] 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**.

[0099] 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.

[0100] 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**.

[0101] 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.

[0102] 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.

[0103] 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.

[0104] 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.

[0105] 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.

[0106] 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 node **310**.

[0107] 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.

[0108] 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.

[0109] A number of the steps/features that may utilize the AI/ML process described herein include one or more of: determining a charging window to charge an electric vehicle (EV) battery of a vehicle from a first charge level to a second charge level, determining a delivery window during which an item will be delivered to the vehicle based on the charging window, and dispatching a delivery of the item to the vehicle, determining a type of a charging point used to charge the vehicle and a type of the vehicle, and wherein the charging window is based on the type of the charging point and the type of the vehicle, and predicting the delivery window based on the type of the charging point and the type of the vehicle. The process may also include determining whether the item is consumable by a vehicle occupant of the vehicle during the charging window, and wherein the item is one or more of content and a consumable product, subtracting an amount of time for the delivery from the charging window to identify a remaining time window, determining the item may be consumed during the remaining time window, and determining the dispatching of the delivery is acceptable, determining a location of the item, determining a status of a delivery vehicle associated with the item, and determining whether to perform the dispatching of the delivery of the item to the vehicle based on the location, the status and the delivery window being shorter than the charging window. The process may also include determining to perform the dispatching when the delivery window will be completed prior to the vehicle achieving the second charge level and determining the first charge level of the vehicle is above a threshold first charge level, and cancelling the delivery when the delivery cannot be made in time.

[0110] 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.

[0111] 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**.

[0112] 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 elements 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.

[0113] 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.

[0114] 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.

[0115] 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.

[0116] 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.

[0117] 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.

[0118] 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.

[0119] 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.

[0120] 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.

[0121] 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.

[0122] 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. **4**B can be utilized in this and other networks and/or systems.

[0123] FIG. **4**B is a diagram showing interconnections between different elements **400**B. The instant solution may be stored and/or executed entirely or partially on and/or by one or more computing devices **414**C, **418**C, **424**C, **428**C, **432**C, **436**C, **406**C, **442**C and **410**C associated with various entities, all communicably coupled and in communication with a network **402**C. A database **438**C 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 **404**C, one or more service provider **416**C, one or more public buildings **422**C, one or more traffic infrastructure **426**C, one or more residential dwellings **430**C, an electric grid/charging station **434**C, a microphone **440**C, and/or another vehicle **408**C. Other entities and/or devices, such as one or more private users using a smartphone **412**C, a laptop **420**C, an augmented reality (AR) device, a virtual reality (VR) device, and/or any wearable device may also interwork with the instant solution. The smartphone **412**C, laptop **420**C, the microphone **440**C, and other devices may be connected to one or more of the connected computing devices **414**C, **418**C, **424**C, **428**C, **432**C, **436**C, **406**C, **442**C, and **410**C. The one or more public buildings **422**C may include various agencies. The one or more public buildings **422**C may utilize a computing device **424**C. The one or more service provider **416**C may include a dealership, a tow truck service, a collision center, or other repair shop. The one or more service provider **416**C may utilize a computing apparatus **418**C.

[0124] 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 **440**C may be utilized as a virtual assistant, in one example. In one example, the one or more traffic infrastructure **426**C 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 **426**C may utilize a computing device **428**C.

[0125] 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.

[0126] In one example, a vehicle **408**C/**404**C can transport a person, an object, a permanently or temporarily affixed apparatus, and the like. In one example, the vehicle **408**C may communicate with vehicle **404**C via V2V communication through the computers associated with each vehicle **406**C and **410**C and may be referred to as a car, vehicle, automobile, and the like. The vehicle **404**C/**408**C 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 **404**C/**408**C 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 **404**C/**408**C may be semi-autonomous or autonomous. For example, vehicle **404**C/**408**C 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. **4**B.

[0127] FIG. **4**C is another block diagram showing interconnections between different elements in one example **400**C. A vehicle **412**D is presented and includes ECUs **410**D, **408**D, and a Head Unit (otherwise known as an Infotainment System) **406**D. 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 **416**D. The ECUs may also communicate with a vehicle computer **404**D via the CAN bus **416**D. The vehicle's processors/sensors (such as the vehicle computer) **404**D can communicate with external elements, such as a server **418**D via a network **402**D (such as the Internet). Each ECU **410**D, **408**D, and Head Unit **406**D 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 **404**D.

[0128] ECUs **410**D, **408**D, and Head Unit **406**D may each include a custom security functionality element **414**D 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.

[0129] Referring to FIG. **4**D, an operating environment **400**D for a connected vehicle, is illustrated according to some embodiments. As depicted, the vehicle **410**E includes a CAN bus **408**E connecting elements **412**E-**426**E 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 **412**E, Electronic Control Units **414**E, autonomous features or Advanced Driver Assistance Systems (ADAS) **416**E, and the navigation system **418**E. In some embodiments, the vehicle **410**E includes a processor **420**E, a memory **422**E, a communication unit **424**E, and an electronic display **426**E.

[0130] The processor **420**E 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.

[0131] 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.

[0132] 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.

[0133] 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.

[0134] 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.

[0135] 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.

[0136] 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.

[0137] 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.

[0138] 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.
[0139] 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.
[0140] 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.
[0141] 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.
[0142] 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.
[0143] 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.
[0144] FIG. **4**E illustrates an example **400**E of vehicles **4021** and **408**I performing secured V2V communications using security certificates, according to example embodiments. Referring to FIG. **4**E, the vehicles **4021** and **408**I may communicate via V2V communications over a short-range network, a cellular network, or the like. Before sending messages, the vehicles **4021** 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 **4041** and **4101** are associated with the vehicles **4021** and **408**I, respectively, in one example.
[0145] Upon receiving the communications from each other, the vehicles may verify the signatures with a certificate authority **4061** or the like. For example, the vehicle **408**I may verify with the

certificate authority **4061** that the public key certificate **404**I used by vehicle **402**I to sign a V2V communication is authentic. If the vehicle **4081** 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 **4061** 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.

[0146] 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.

[0147] 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.

[0148] 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.

[0149] 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.

[0150] 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.

[0151] 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.

[0152] 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.

[0153] 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.

[0154] 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.

[0155] 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.

[0156] 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.

[0157] 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.

[0158] 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.
[0159] 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.
[0160] 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.
[0161] 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 if 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.
[0162] 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.
[0163] 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.

[0164] 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.

[0165] 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.

[0166] 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.

[0167] 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.

[0168] 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.

[0169] 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.

[0170] 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.

[0171] 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.

[0172] 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,

512E, and **513**E may perform the role of endorser node, committer node, or both.

[0173] 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.

[0174] 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.

[0175] 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.

[0176] Endorsing nodes receive transactions from clients and endorse the transaction based on simulated results. Endorsing nodes hold smart contracts which simulate the transaction proposals.

[0177] 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.

[0178] 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.

[0179] 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.

[0180] 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.

[0181] 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.

[0182] Referring to FIG. 5F **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. 5F, 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. 5E) 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.

[0183] 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. 5E), 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.

[0184] In one embodiment of the instant solution, the block data may include data comprising one or more of determining a charging window to charge an electric vehicle (EV) battery of a vehicle from a first charge level to a second charge level, determining a delivery window during which an item will be delivered to the vehicle based on the charging window, and dispatching a delivery of the item to the vehicle.

[0185] Although in FIG. 5F 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**. In some embodiments, the charging window to charge an electric vehicle battery of a vehicle from a first charge level to a second charge level, the delivery window during which an item will be delivered to the vehicle based on the charging window, and the status of the dispatching of the delivery of the item to the vehicle, etc. may be written to the blockchain data **563** and committed to a blockchain ledger.

[0186] 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. 5E. Meanwhile, a committer of the block (such as blockchain node **512**E in FIG. 5E) 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.

[0187] 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.

[0188] 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.

[0189] 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**, computing system **601** is operational within numerous other general-purpose or special-purpose computing system environments or configurations.

[0190] Computing 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 computing system **601**, to keep the presentation as simple as possible.

[0191] Computing system **601** may be located in a cloud, even though it is not shown in a cloud in FIG. **6**. On the other hand, computing system **601** is not required to be in a cloud except to any extent as may be affirmatively indicated. Computing system **601** may be described in the general context of computer system-executable instructions, such as program modules, executed by a computing 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**, computing system **601** in computing environment **600** is shown in the form of a general-purpose computing device. The components of computing 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 processing unit **602**.

[0192] 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.

[0193] Network adapter **603** enables the computing 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.

[0194] Computing 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 computing system **601** is required to have a large amount of storage (for example, where computing system **601** locally stores and manages a large database), then this storage may be provided by 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.

[0195] The operating system **611** is software that manages computing 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.

[0196] 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 Interconnect (PCI) bus. The bus **620** is the signal conduction paths that allow the various components of computing system **601** to communicate with each other.

[0197] 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 computing system **601**, memory **630** is in a single package and is internal to computing system **601**, but alternatively or additionally, the volatile memory may be distributed over multiple packages and/or located externally with respect to computing 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 computing 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 computing 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**.

[0198] Computing 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 computing system **601**; and/or any devices (e.g., network card, modem, etc.) that enable computing 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 computing system **601** via bus **620**.

[0199] 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. Computing system **601** connects to network **650** via network adapter **603** and bus **620**.

[0200] User devices **651** are any computer systems used and controlled by an end user in connection with computing system **601**. For example, in a hypothetical case where computing system **601** is designed to provide a recommendation to an end user, this recommendation may typically be communicated from network adapter **603** of computing 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.

[0201] 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 computing 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. 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 computing system **601** across a network **650**.

[0202] 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 applications 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.

[0203] 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.

[0204] 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.

[0205] 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.

[0206] 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.

[0207] 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.

[0208] 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.

[0209] 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.
[0210] 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.

Claims

1. A method comprising: determining a charging window to charge an electric vehicle (EV) battery of a vehicle from a first charge level to a second charge level; determining a delivery window during which an item will be delivered to the vehicle based on the charging window; and dispatching a delivery of the item to the vehicle.
2. The method of claim 1, comprising: determining a type of a charging point used to charge the vehicle and a type of the vehicle, and wherein the charging window is based on the type of the charging point and the type of the vehicle; and predicting the delivery window based on the type of the charging point and the type of the vehicle.
3. The method of claim 1, comprising: determining whether the item is consumable by a vehicle occupant of the vehicle during the charging window, and wherein the item is one or more of content and a consumable product.
4. The method of claim 3, comprising: subtracting an amount of time for the delivery from the charging window to identify a remaining time window; determining the item may be consumed during the remaining time window; and determining the dispatching of the delivery is acceptable.
5. The method of claim 1, comprising: determining a location of the item; determining a status of a delivery vehicle associated with the item; and determining whether to perform the dispatching of the delivery of the item to the vehicle based on the location, the status and the delivery window being shorter than the charging window.
6. The method of claim 1, comprising: determining to perform the dispatching when the delivery window will be completed prior to the vehicle achieving the second charge level.
7. The method of claim 1, comprising: determining the first charge level of the vehicle is above a threshold first charge level; and cancelling the delivery.
8. A system comprising a memory and a processor configured to: determine a charging window to charge an electric vehicle (EV) battery of a vehicle from a first charge level to a second charge level; determine a delivery window during which an item will be delivered to the vehicle based on the charging window; and dispatch a delivery of the item to the vehicle.
9. The system of claim 8, wherein the processor is further configured to: determine a type of a charging point used to charge the vehicle and a type of the vehicle, and wherein the charging window is based on the type of the charging point and the type of the vehicle; and predict the delivery window based on the type of the charging point and the type of the vehicle.
10. The system of claim 8, wherein the processor is further configured to: determine whether the item is consumable by a vehicle occupant of the vehicle during the charging window, and wherein the item is one or more of content and a consumable product.
11. The system of claim 10, wherein the processor is further configured to: subtract an amount of time for the delivery from the charging window to identify a remaining time window; determine the item may be consumed during the remaining time window; and determine the dispatching of the delivery is acceptable.
12. The system of claim 8, wherein the processor is further configured to: determine a location of the item; determine a status of a delivery vehicle associated with the item; and determine whether to perform the dispatching of the delivery of the item to the vehicle based on the location, the status and the delivery window being shorter than the charging window.
13. The system of claim 8, wherein the processor is further configured to: determine to perform the

dispatch when the delivery window will be completed prior to the vehicle achieving the second charge level.

14. The system of claim 8, wherein the processor is further configured to: determine the first charge level of the vehicle is above a threshold first charge level; and cancel the delivery.

15. A non-transitory computer readable storage medium configured to store instructions that when executed cause a processor to perform: determining a charging window to charge an electric vehicle (EV) battery of a vehicle from a first charge level to a second charge level; determining a delivery window during which an item will be delivered to the vehicle based on the charging window; and dispatching a delivery of the item to the vehicle.

16. The non-transitory computer readable storage medium of claim 15, wherein the processor is further configured to perform: determining a type of a charging point used to charge the vehicle and a type of the vehicle, and wherein the charging window is based on the type of the charging point and the type of the vehicle; and predicting the delivery window based on the type of the charging point and the type of the vehicle.

17. The non-transitory computer readable storage medium of claim 15, wherein the processor is further configured to perform: determining whether the item is consumable by a vehicle occupant of the vehicle during the charging window, and wherein the item is one or more of content and a consumable product.

18. The non-transitory computer readable storage medium of claim 17, wherein the processor is further configured to perform: subtracting an amount of time for the delivery from the charging window to identify a remaining time window; determining the item may be consumed during the remaining time window; and determining the dispatching of the delivery is acceptable.

19. The non-transitory computer readable storage medium of claim 15, wherein the processor is further configured to perform: determining a location of the item; determining a status of a delivery vehicle associated with the item; and determining whether to perform the dispatching of the delivery of the item to the vehicle based on the location, the status and the delivery window being shorter than the charging window.

20. The non-transitory computer readable storage medium of claim 15, wherein the processor is further configured to perform: determining to perform the dispatching when the delivery window will be completed prior to the vehicle achieving the second charge level.
