

INTERNET OF THINGS ARCHITECTURE FOR DEVICE SHARING

TECHNICAL FIELD

[0001] Various aspects of this disclosure generally relate to Internet of things (IoT), and more particularly, to an IoT architecture for device sharing.

BACKGROUND

[0002] The Internet of things (IoT) is the network of devices such as vehicles and home appliances that contain electronics, software, actuators, and connectivity which allows these things to connect, interact and exchange data. The IoT involves extending Internet connectivity beyond standard devices, such as desktops, laptops, smartphones and tablets, to any range of traditionally non-internet-enabled physical devices and everyday objects. Embedded with technology, these devices can communicate and interact over the Internet, and they can be remotely monitored and controlled.

[0003] Mobile device sharing is a transportation innovation that is growing rapidly across cities. It solves the “last mile” problem by providing users an alternative device at better estimated time of arrival (ETA) and price than cars in crowded cities, reduces carbon emission, and provides a smarter transportation network to cities around the world. The future of urban mobility is shared, seamless, smart and environmentally-sustainable. Motorized scooters would be a good addition to the active mobility landscape, serving unmet demands in the first-mile-last-mile (FMLM) travel segment.

[0004] The key of mobile device sharing is IoT communications. The success of mobile device sharing depends on how well these devices respond to user commands (by mobile app) or instructions from the cloud. Consumers are demanding on the experiences so any information delay is going to add friction to the user experience. Security is a core requirement as the devices are intelligent assets being deployed in the wild on their own and potentially vulnerable to hacking, theft, as well as other forms of sabotage.

SUMMARY

[0005] The following presents a simplified summary of one or more aspects in order to provide a basic understanding of such aspects. This summary is not an extensive overview of all contemplated aspects, and is intended to neither identify key or critical elements of all aspects nor delineate the scope of any or all aspects. Its sole purpose is to present some concepts of one or more aspects in a simplified form as a prelude to the more detailed description that is presented later.

[0006] In an aspect of the disclosure, a method, a computer readable medium, and an apparatus for operating a motorized scooter are provided. The apparatus may be a part of the motorized scooter. The apparatus may receive an unlock request from a remote server. The apparatus may unlock itself in response to the unlock request. The apparatus may continuously upload its status to the remote server based on a schedule when it is unlocked. The apparatus may receive a lock request from the remote server. The apparatus may lock itself in response to the lock request.

[0007] In another aspect of the disclosure, a method, a computer readable medium, and an apparatus for managing motorized scooters are provided. The apparatus may be a server. The apparatus may receive, from a mobile device, a vehicle identifier of a motorized scooter. The apparatus may retrieve a vehicle status of the motorized scooter based on the vehicle identifier. The apparatus may generate an unlock request when the vehicle status is a normal usable status. The apparatus may transmit the unlock request to the motorized scooter. The apparatus may continuously receive, from the motorized scooter, the vehicle status of the motorized scooter based on a schedule. The apparatus may receive a message from the mobile device, the message requesting locking the motorized scooter. The apparatus may generate a lock request in response to the message when the motorized scooter is in an unlocking status. The apparatus may transmit the lock request to the motorized scooter.

[0008] To the accomplishment of the foregoing and related ends, the one or more aspects include the features hereinafter fully described and particularly pointed out in the claims. The following description and the annexed drawings set forth in detail certain illustrative features of the one or more aspects. These features are indicative, however, of but a few of the various ways in which the principles of various aspects may be employed, and this description is intended to include all such aspects and their equivalents.

BRIEF DESCRIPTION OF THE DRAWINGS

- [0009] FIG. 1 is a diagram illustrating an example of a networking IoT system for device sharing.
- [0010] FIG. 2 is a diagram illustrating an example of triple lock methods that include scanning Quick Response Code, geo-fence, and backend remote control.
- [0011] FIG. 3 is a diagram illustrating an example of an IoT device.
- [0012] FIG. 4 is a diagram illustrating an example of hardware structure of an IoT communication module.
- [0013] FIG. 5 is a diagram illustrating an example of the components of an IoT system.
- [0014] FIG. 6 is a flowchart of a method of operating a motorized scooter.
- [0015] FIG. 7 is a conceptual data flow diagram illustrating the data flow between different means/components in an exemplary apparatus.
- [0016] FIG. 8 is a diagram illustrating an example of a hardware implementation for an apparatus employing a processing system.
- [0017] FIG. 9 is a flowchart of a method of managing motorized scooters.
- [0018] FIG. 10 is a conceptual data flow diagram illustrating the data flow between different means/components in an exemplary apparatus.
- [0019] FIG. 11 is a diagram illustrating an example of a hardware implementation for an apparatus employing a processing system.

DETAILED DESCRIPTION

[0020] The detailed description set forth below in connection with the appended drawings is intended as a description of various configurations and is not intended to represent the only configurations in which the concepts described herein may be practiced. The detailed description includes specific details for the purpose of providing a thorough understanding of various concepts. However, it will be apparent to those skilled in the art that these concepts may be practiced without these specific details. In some instances, well known structures and components are shown in block diagram form in order to avoid obscuring such concepts.

[0021] Several aspects of an IoT architecture for device sharing will now be presented with reference to various apparatus and methods. These apparatus and methods will be described in the following detailed description and illustrated in the accompanying drawings by various blocks, components, circuits, processes, algorithms, etc. (collectively referred to as “elements”). These elements may be implemented using electronic hardware, computer software, or any combination thereof. Whether such elements are implemented as hardware or software depends upon the particular application and design constraints imposed on the overall system.

[0022] By way of example, an element, or any portion of an element, or any combination of elements may be implemented as a “processing system” that includes one or more processors. Examples of processors include microprocessors, microcontrollers, graphics processing units (GPUs), central processing units (CPUs), application processors, digital signal processors (DSPs), reduced instruction set computing (RISC) processors, systems on a chip (SoC), baseband processors, field programmable gate arrays (FPGAs), programmable logic devices (PLDs), state machines, gated logic, discrete hardware circuits, and other suitable hardware configured to perform the various functionality described throughout this disclosure. One or more processors in the processing system may execute software. Software shall be construed broadly to mean instructions, instruction sets, code, code segments, program code, programs, subprograms, software components, applications, software applications, software packages, routines, subroutines, objects, executables, threads of execution, procedures, functions, etc., whether referred to as software, firmware, middleware, microcode, hardware description language, or otherwise.

[0023] Accordingly, in one or more example embodiments, the functions described may be implemented in hardware, software, or any combination thereof. If implemented in software, the functions may be stored on or encoded as one or more instructions or code on a computer-readable medium. Computer-readable media includes computer storage media. Storage media may be any available media that can be accessed by a computer. By way of example, and not limitation, such computer-readable media may include a random-access memory (RAM), a read-only memory (ROM), an electrically erasable programmable ROM (EEPROM), optical disk storage, magnetic disk storage, other magnetic storage devices, combinations of the aforementioned types of computer-readable

media, or any other medium that can be used to store computer executable code in the form of instructions or data structures that can be accessed by a computer.

[0024] Most of the shared bike and scooter operations in the market have limited IoT capabilities. Bluetooth connection is the most common solution used. There are a few drawbacks came with the Bluetooth solution. First, the Bluetooth solution lacks centralized control, which makes effective management of the devices almost impossible. No vehicle information such as Global Positioning System (GPS) location, vehicle status, battery level is maintained at the backend. This adds difficulty to daily operations and maintenance. Second, compatibility problems of the Bluetooth solution is well known among mobile devices. As a result, the communication between the phones and the smart locks may malfunction in quite a few circumstances. Third, the user experiences of the Bluetooth solution are far from satisfactory due to the significant latency incurred during the communication with the devices.

[0025] To address the problems mentioned above, a networking IoT system is provided. In the IoT system, IoT devices have to maintain persistent connection round-the-clock (with possible retries to connect). This way, the IoT devices are free to upload any device information to the server side at any time. And the cloud service may store and maintain the status of all the devices in databases, which may be checked by the operations staff as per their permissions. Besides, when necessary, the server cluster may push certain commands to the IoT devices to proactively control their behaviors, such as locking/unlocking, forcing over-the-air (OTA) update, setting top speed, and stopping any abnormal behaviors. In addition, unified subscriber identity module (SIM) cards may be installed into all the IoT devices, which eliminates the potential compatibility problems. Lastly, the fully bidirectional networking channel, along with the extremely lean data packets transferred, enables super-fast communication between the server and the IoT devices and smooth user experiences of using the vehicles.

[0026] FIG. 1 is a diagram 100 illustrating an example of a networking IoT system for device sharing. In the example, the IoT system may include a server 102 and several IoT devices 110, 112 ..., 114. The server 102 may include one or more computing devices (e.g., apparatus 1002/1002' described below with reference to FIG. 10 or 11). In some embodiments, each of the IoT devices 110, 112 ..., 114 may be a motorized scooter. In

some embodiments, each of the IoT devices 110, 112 ..., 114 may be the apparatus 702/702' described below with reference to FIG. 7 or 8).

[0027] In some embodiments, the server 102 and the IoT devices 110, 112 ..., 114 may maintain persistent connections round-the-clock (with possible retries to connect/re-connect). Each of the IoT devices 110, 112 ..., 114 may upload its device information to the server 102. The server 102 may push commands to each of the IoT devices 110, 112 ..., 114 to control their behaviours, such as locking/unlocking, forcing over-the-air (OTA) update, setting top speed, and stopping any abnormal behaviors.

[0028] In some embodiments, binary data format may be used to carry out the communication between the IoT devices 110, 112 ..., 114 and the server 102. This binary data format may be proprietary and thus hard to decrypt. In some embodiments, end-to-end encryption may add an extra layer of security to the data transferred. In some embodiments, security check may be triggered when an IoT device tries to establish connections with the server 102, which only allows the right pair of International Mobile Equipment Identity (IMEI) and vehicle number to pass.

[0029] In some embodiments, binary compact data format may make the data hard to crack and small in size. Also, the least amount of bytes may be used to present each data type accurately. In some embodiments, different endpoint calls may be combined into a single one whenever possible to reduce the latency of core flows such as unlocking/locking, that is to say, to carry out batch processing. For example, during unlocking, the commands to unlock the scooter, ask for vehicle location, set proper speed limit may be combined into one to issue to the scooter.

[0030] In some embodiments, scanning Quick Response Code (QR code) to unlock a motorized scooter may be completed in under 5 seconds with reminder music; and this reminder music may be dynamically configured based on time, location and the identifier of the motorized scooter, e.g., Jingle Bells on Christmas Eve. In some embodiments, the locking/unlocking process may include several interactions, but still may be completed under 5 seconds.

[0031] There is always a risk of IoT connection failure (i.e., due to the failure of Internet connection). In some embodiments, a supplementary process may be added to establish regular checks (initiated from backend server) on the connection, lock status, and order status. In some embodiments, the supplementary process may be implemented to

scan the full system and live order status every few seconds. In the situation where there is no order but the scooter unlocked, locking may be triggered (to prevent loss and theft). In situations where user order active but scooter locked, the user order may be terminated to prevent overcharging. This makes the system intelligent and capable of self-recovery from any issue caused by IoT connection failure.

[0032] An example of the unlocking process is described below:

B1: User app scans the QR code and passes the vehicle number to the backend server;

B2: Backend server retrieves vehicle status. If it is in normal usable status, then the unlock request is sent to IoT after conversion protocol;

B3: IoT receives the unlock request and decides whether the unlock request is valid. If it is valid, it unlocks and plays the unlock music. If it is invalid, it returns unlock failure to the backend server.

[0033] An example of the locking process is described below:

B1: The user clicks the lock button through the user app, which transmits the lock information to the backend server;

B2: Retrieve the vehicle status in the backend server. If the vehicle is unlocking status, the lock request will be sent to IoT after the conversion protocol.

B3: IoT receives the lock instruction to determine whether the lock instruction is valid. If it is valid, it locks the vehicle and plays the lock music. If it is invalid, it returns lock failure to the backend server.

[0034] An example of the supplementary locking process is described below:

B1: IoT uploads the current status to the backend server every 30 seconds when it is in unlocking status;

B2: After receiving the status via IoT upload, the backend server determines whether there is an illegal lock status, and if there is, generates the lock instruction to avoid illegal riding.

[0035] In some embodiments, wireless mobile communication data and short message service (SMS) seamless switch backup communication ensures higher than 90% unlock successful rate. In some embodiments, X seconds after the server issues unlocking/locking command to the IoT device, the server would send an SMS message down to the IoT device if no networking response is given from the device side. And the same is applied for

the uplink from IoT device to the server. The X above may be dynamically adjusted at the backend server according to the real-time distribution of the unlocking durations among the sessions during a certain time period.

[0036] FIG. 2 is a diagram 200 illustrating an example of triple lock methods that include scanning QR code, geo-fence, and backend remote control. In some embodiments, triple lock methods may ensure higher than 95% lock rate. In the example, at 204, the user may scan parking QR code associated with the parking location. At 206, the backend server may check the parking location and the location of the IoT device. At 208, if the parking location and the location of the IoT device matches, the backend server may send a locking request to the IoT device to lock the motorized scooter.

[0037] In some embodiments, the highest riding speed of the motorized scooter may be dynamically remote controlled, which ensures user safety. In some embodiments, top speed for scooters may be lowered in certain scenarios. Examples of these scenarios include: low speed zones for certain locations, low speed profiles for uncertified/beginner users, and low speed periods after dark. In some embodiments, fleet (or any individual scooter's) top speed may be controlled based on specific user, geo-fence, lighting conditions, hours of day, weather condition, etc.

[0038] In some embodiments, vehicle data may be tracked close to real time (e.g., every 30 seconds upload when vehicles are on trip, every 30 min upload when vehicle are not hired). This allows several additional functions to be performed on operations. In some embodiments, vehicle location, status, as well as basic information such as battery level may be tracked to aid fleet management.

[0039] In some embodiments, operations personnel may be able to get real-time map and status information on high-risk scooters such as those that are low on battery power, and those that are not returned to proper parking lots by users. In some embodiments, a proprietary-designed app may facilitate the search of stray scooters by allowing operations personnel to tap on either the Chirp or Flash button features to sound the chime or switch on the headlights on scooters, to make it easier for staff to locate improperly parked scooters. In some embodiments, a new feature may be introduced within the operations management system to automate stray scooter alerts. For example, scooters that are not parked at the designated lots are being alerted in real time through the operations management system. This will help reduce stray recover time. In some embodiments,

visualization of all stray scooters and their locations relative to deployment areas and proper parking lots may be presented onto a map.

[0040] In some embodiments, it may be possible to remotely trigger commands to the fleet. The remotely triggered commands may include lock, unlock, and managing power, speed, IoT connection status, etc. In an effort to increase safety of users and pedestrians, some embodiments may implement auto-on of headlights after dark for a selected fleet of scooters. Some embodiments may implement always-on headlight after dark for all rides.

[0041] In some embodiments, the firmware of each motorized scooter may be dynamically upgraded based on time, location, and scooter identifier (ID) through encrypted over-air channel. And this over-the-air (OTA) upgrade may be executed automatically according to the preset conditions.

[0042] An example of the OTA upgrade process is described below:

A1: Backend servers may actively issue forced OTA upgrade instructions to specified IoT or IoT timing retrieval (random distribution of retrieval time points for each IoT device) if there is a valid new version available for upgrading.

A2: After starting OTA upgrade, the IoT device downloads the matching version from OTA server. After downloading, it upgrades automatically to the latest version and completes the OTA upgrade process.

[0043] In some embodiments, the OTA upgrade process may be initiated by server side. To realize sustainability (in case of emergency OTA, hot fixes, and frequent needs of upgrade), the OTA upgrade capability may be implemented from the backend server. As a result, good stability may be achieved even with the additional challenges.

[0044] In some embodiments, the OTA upgrade may be scheduled in specific times. This is very critical because it is undesirable to do OTA upgrade when fleet is in service. In some embodiments, it may be possible to schedule the OTA upgrade for a specific fleet during a specific time when the fleet is in maintenance or before-deployment. Also due to concern of jamming network during peak time (e.g., connection failures are likely to happen if the entire 5000 fleet performs OTA upgrade), an intelligent solution may be implemented to automatically optimize this OTA upgrade process for best stability and success rate. First, a few scooters within the fleet may be picked randomly to initiate OTA upgrade. Depending on the result of the OTA upgrade for the first batch of scooters, the system may automatically trigger the next batch (including determining the size of the next

batch) and send alerts if needed during the process. The size of each batch may be optimized for best results and success rate as well as to avoid network jam.

[0045] FIG. 3 is a diagram illustrating an example of an IoT device 300. In some embodiments, the IoT device 300 may be a motorized scooter equipped with an IoT communication module 302, which may be an external box attached to the motorized scooter. The IoT communication module 302 may enable optimization of the operations or maintenance of the motorized scooter so that efficiency may be achieved. For example, the motorized scooter may be unlocked/locked by scanning the QR code. The unlock time may be less than 5 seconds, and the success rate may be higher than 90%. In some embodiments, user riding data (e.g., when you ride, where you are riding, how long you ride, etc.) may be collected in real-time. In some embodiments, the real-time data of the motorized scooter (e.g., the speed, the power volume, how far has already ride, falling down on the ground or not, etc.) may be collected. In some embodiments, customized human-machine interface (e.g., unlock or lock reminder, alarm, etc.) may be provided to users.

[0046] In some embodiments, the IoT communication module 302 may serve to track the scooters' service status and location at any given time. The IoT communication module 302 also makes it possible to transmit commands to the scooters via cloud server. In addition, the IoT communication module 302 may contain an internal speaker for unlocking/locking indication and alerting the sensors for any abnormal activities detected (e.g., falling down, abnormal movement, etc.) to enhance operational efficiency.

[0047] In some embodiments, the design of the antenna of the IoT communication module 302 may be optimized for sharing scenario. In some embodiments, the electronics of the IoT device 300 may be fail safe for electric failure and battery safety. The design of the IoT device 300 is focused on optimizing for user experience (e.g., unlock, lock) so user spends as little time interacting with the app as possible. The product is designed specifically for situations of theft and sabotage, which is known issues in the shared device industry.

[0048] FIG. 4 is a diagram illustrating an example of hardware structure of an IoT communication module 400. In some embodiments, the IoT communication module 400 may be the IoT communication module 302 described above in FIG. 3. In the example, the IoT communication module 400 may include a communication module 402, a lock control

module 404, an application processing module 406, a sensor module 408, an energy management module 410, a control module 412, a driver module 414, a chipset platform 418, and a system management module 416.

[0049] In some embodiments, IoT firmware may support “cellular wireless data and SMS seamless backup” unlock method, which ensure the higher than 90% unlock successful rate. In some embodiments, IoT firmware may support triple lock methods (scanning QR code, geo-fence, and backend remote control) to ensure higher than 95% lock rate. In some embodiments, the highest riding speed may be dynamically remotely configured for each motorized scooter by the control module 412.

[0050] FIG. 5 is a diagram illustrating an example of the components of an IoT system 500. In the example, the IoT system 500 may include a message queue 502, a Message Queuing Telemetry Transport (MQTT) server cluster 506, smart devices 508, MQTT host routing and message assembly/resolution component 510, a registration center 512, a core transaction layer 516, an app gateway 518, a consumer app 520, an operation gateway 522, and an operations app 526.

[0051] The smart devices 508 may include several IoT devices described above with reference to FIGS. 1-4. The highly optimized MQTT server cluster 506 or the extremely lightweight data protocol may be easily extended to support millions of connections to IoT devices. The message queue 502 sits between the transaction system (e.g., the core transaction layer 516) and the MQTT server cluster 506, and all the communications between these two parties have to go through the message queue 502. This effectively eliminates the coupling between these two sets of systems, and the entire architecture benefits from the high scalability of the message queue 502. In some embodiments, the IoT system 500 may have theoretically as many frontend MQTT servers as possible. The MQTT server cluster 506 connects directly with the IoT devices, and throw arbitrary number of messages received from the IoT devices to the message queue 502. Then, the multiple transaction systems sitting on the other side of the message queue 502 may choose to consumer the messages at any speed as they like, depending on their computing capacity at any given time, and may even decide to give up those messages that they do not care about.

[0052] The message queue 502 takes the role of routing various messages between the MQTT server cluster 506 and the core transaction layer 516. In some embodiments, micro-

service backend transaction system makes every service decoupled from each other, thus enabling managing different aspects of vehicle data separately, which helps to improve the robustness and security of the entire set of services.

[0053] FIG. 6 is a flowchart 600 of a method of operating a motorized scooter. The motorized scooter may be an IoT device described above with reference to FIGS. 1-5. The method may be performed by an apparatus (e.g., apparatus 702/702' described below with reference to FIG. 7 or 8). In some embodiments, the apparatus may be a motorized scooter. In some embodiments, the apparatus may be the IoT component of a motorized scooter.

[0054] At 602, the apparatus may receive an unlock request from a remote server. In some embodiments, an application program executing on a mobile device may scan a machine-readable optical label (e.g., a QR code) on the motorized scooter and send a vehicle identifier corresponding to the machine-readable optical label to the remote server. The unlock request may be generated and transmitted, by the remote server, based on the vehicle identifier. In some embodiments, the unlock request may be generated when the motorized scooter is in normal usable status. In some embodiments, the apparatus may establish a wireless connection with the remote server based on a wireless communication identifier associated with the motorized scooter and a vehicle identifier of the motorized scooter.

[0055] At 604, the apparatus may unlock the motorized scooter in response to the unlock request. In some embodiments, the apparatus may receive a speed control command from the remote server. The apparatus may set the maximum speed of the motorized scooter based on the speed control command.

[0056] At 606, the apparatus may continuously upload status of the motorized scooter to the remote server based on a schedule when the motorized scooter is unlocked.

[0057] At 608, the apparatus may receive a lock request from the remote server. In some embodiments, an application program executing on a mobile device may receive a user input to lock the motorized scooter, and send a message to the remote server to request locking the motorized scooter in response to the user input. The lock request may be generated by the remote server in response to the message when the motorized scooter is in an unlocking status.

[0058] At 610, the apparatus may lock the motorized scooter in response to the lock request.

[0059] FIG. 7 is a conceptual data flow diagram 700 illustrating the data flow between different means/components in an exemplary apparatus 702. The apparatus 702 may be an IoT device (e.g., the IoT device 110, 112, 114, or 300, or the smart devices 508). The apparatus 702 may include a reception component 704 that receives control commands from a remote server 750. In one embodiment, the reception component 704 may perform the operations described above with reference to 602 or 608 in FIG. 6.

[0060] The apparatus 702 may include a transmission component 710 that transmits status of the apparatus 702 to the remote server 750. In one embodiment, the transmission component 710 may perform the operations described above with reference to 606 in FIG. 6. The reception component 704 and the transmission component 710 may collaborate to coordinate the communication of the apparatus 702.

[0061] The apparatus 702 may include a control component 706 that is configured to control the operation of the apparatus 702 based on the commands received from the reception component 704. In one embodiment, the control component 706 may perform the operations described above with reference to 604 or 610 in FIG. 6.

[0062] The apparatus 702 may include a data collection component 708 that is configured to collect status data of the apparatus 702 and provide the collected status data to the transmission component 710 for transmission to the remote server 750. In one embodiment, the data collection component 708 may perform the operations described above with reference to 606 in FIG. 6.

[0063] The apparatus 702 may include additional components that perform each of the blocks of the algorithm in the aforementioned flowchart of FIG. 6. As such, each block in the aforementioned flowchart of FIG. 6 may be performed by a component and the apparatus may include one or more of those components. The components may be one or more hardware components specifically configured to carry out the stated processes/algorithm, implemented by a processor configured to perform the stated processes/algorithm, stored within a computer-readable medium for implementation by a processor, or some combination thereof.

[0064] FIG. 8 is a diagram 800 illustrating an example of a hardware implementation for an apparatus 702' employing a processing system 814. In one embodiment, the apparatus 702' may be the apparatus 702 described above with reference to FIG. 7. The processing system 814 may be implemented with a bus architecture, represented generally

by the bus 824. The bus 824 may include any number of interconnecting buses and bridges depending on the specific application of the processing system 814 and the overall design constraints. The bus 824 links together various circuits including one or more processors and/or hardware components, represented by the processor 804, the components 704, 706, 708, 710, and the computer-readable medium / memory 806. The bus 824 may also link various other circuits such as timing sources, peripherals, voltage regulators, and power management circuits, which are well known in the art, and therefore, will not be described any further.

[0065] The processing system 814 may be coupled to a transceiver 810. The transceiver 810 is coupled to one or more antennas 820. The transceiver 810 provides a means for communicating with various other apparatus over a transmission medium. The transceiver 810 receives a signal from the one or more antennas 820, extracts information from the received signal, and provides the extracted information to the processing system 814, specifically the reception component 704. In addition, the transceiver 810 receives information from the processing system 814, specifically the transmission component 710, and based on the received information, generates a signal to be applied to the one or more antennas 820.

[0066] The processing system 814 includes a processor 804 coupled to a computer-readable medium / memory 806. The processor 804 is responsible for general processing, including the analyzation of data gathered by the apparatus itself through its own sensors and the execution of software stored on the computer-readable medium / memory 806. The software, when executed by the processor 804, causes the processing system 814 to perform the various functions described *supra* for any particular apparatus. The computer-readable medium / memory 806 may also be used for storing data that is manipulated by the processor 804 when executing software. The processing system 814 further includes at least one of the components 704, 706, 708, 710. The components may be software components running in the processor 804, resident/stored in the computer readable medium / memory 806, one or more hardware components coupled to the processor 804, or some combination thereof.

[0067] FIG. 9 is a flowchart 900 of a method of managing motorized scooters. Each of the motorized scooters may be an IoT device described above with reference to FIGS. 1-8. The method may be performed by an apparatus (e.g., apparatus 1002/1002' described

below with reference to FIG. 10 or 11). In some embodiments, the apparatus may be a server that includes one or more computing devices.

[0068] At 902, the apparatus may receive, from a mobile device, a vehicle identifier of a motorized scooter. In some embodiments, an application program executing on the mobile device may scan a machine-readable optical label (e.g., a QR code) on the motorized scooter and send the vehicle identifier corresponding to the machine-readable optical label to the server. In some embodiments, the apparatus may establish a wireless connection with the motorized scooter based on a wireless communication identifier associated with the motorized scooter and a vehicle identifier of the motorized scooter.

[0069] At 904, the apparatus may retrieve the vehicle status of the motorized scooter based on the vehicle identifier.

[0070] At 906, the apparatus may generate an unlock request when the vehicle status is a normal usable status.

[0071] At 908, the apparatus may transmit the unlock request to the motorized scooter. In some embodiments, the apparatus may generate a speed control command limiting the maximum speed of the motorized scooter based on the vehicle status of the motorized scooter. The apparatus may transmit the speed control command to the motorized scooter.

[0072] At 910, the apparatus may continuously receive, from the motorized scooter, the vehicle status of the motorized scooter based on a schedule. In some embodiments, the apparatus may determine a time instance to upgrade the firmware of the motorized scooter based on the vehicle status of the motorized scooter. The apparatus may upgrade the firmware of the motorized scooter at the determined time instance.

[0073] At 912, the apparatus may receive a message from the mobile device. The message may request locking the motorized scooter.

[0074] At 914, the apparatus may generate a lock request in response to the message when the motorized scooter is in an unlocking status.

[0075] At 916, the apparatus may transmit the lock request to the motorized scooter. In some embodiments, the apparatus may determine whether there is an active order associated with the motorized scooter when the motorized scooter is unlocked. The apparatus may transmit a lock request to the motorized scooter when there is no active order associated with the motorized scooter and the motorized scooter is unlocked. In some embodiments, the apparatus may determine whether an order associated with the motorized

scooter is active when the motorized scooter is locked. The apparatus may terminate the order when the order is active and the motorized scooter is locked.

[0076] FIG. 10 is a conceptual data flow diagram 1000 illustrating the data flow between different means/components in an exemplary apparatus 1002. The apparatus 1002 may be a server (e.g., the server 102 or the MQTT server cluster 506). The apparatus 1002 may include one or more computing devices. The apparatus 1002 may include a reception component 1004 that receives status data and/or vehicle ID from a motorized scooter 1050. The reception component 1004 may further receive messages from a mobile device 1055 that executes consumer app. In one embodiment, the reception component 1004 may perform the operations described above with reference to 902, 910, or 912 in FIG. 9.

[0077] The apparatus 1002 may include a transmission component 1010 that transmits commands to the motorized scooter 1050. In one embodiment, the transmission component 1010 may perform the operations described above with reference to 908 or 916 in FIG. 9. The reception component 1004 and the transmission component 1010 may collaborate to coordinate the communication of the apparatus 1002.

[0078] The apparatus 1002 may include a status management component 1006 that is configured to store and manage the status data and/or vehicle ID received from the reception component 1004. In one embodiment, the status management component 1006 may perform the operations described above with reference to 904 in FIG. 9.

[0079] The apparatus 1002 may include a command generation component 1008 that is configured to generate commands based on messages received from the reception component 1004 and status data provided by the status management component 1006. The generated commands may be provided to the transmission component 1010 for transmission to the motorized scooter 1050. In one embodiment, the command generation component 1008 may perform the operations described above with reference to 906 or 914 in FIG. 9.

[0080] The apparatus 1002 may include additional components that perform each of the blocks of the algorithm in the aforementioned flowchart of FIG. 9. As such, each block in the aforementioned flowchart of FIG. 9 may be performed by a component and the apparatus may include one or more of those components. The components may be one or more hardware components specifically configured to carry out the stated processes/algorithm, implemented by a processor configured to perform the stated

processes/algorithm, stored within a computer-readable medium for implementation by a processor, or some combination thereof.

[0081] FIG. 11 is a diagram 1100 illustrating an example of a hardware implementation for an apparatus 1002' employing a processing system 1114. In one embodiment, the apparatus 1002' may be the apparatus 1002 described above with reference to FIG. 10. The processing system 1114 may be implemented with a bus architecture, represented generally by the bus 1124. The bus 1124 may include any number of interconnecting buses and bridges depending on the specific application of the processing system 1114 and the overall design constraints. The bus 1124 links together various circuits including one or more processors and/or hardware components, represented by the processor 1104, the components 1004, 1006, 1008, 1010, and the computer-readable medium / memory 1106. The bus 1124 may also link various other circuits such as timing sources, peripherals, voltage regulators, and power management circuits, which are well known in the art, and therefore, will not be described any further.

[0082] The processing system 1114 may be coupled to a transceiver 1110. The transceiver 1110 is coupled to one or more antennas 1120. The transceiver 1110 provides a means for communicating with various other apparatus over a transmission medium. The transceiver 1110 receives a signal from the one or more antennas 1120, extracts information from the received signal, and provides the extracted information to the processing system 1114, specifically the reception component 1004. In addition, the transceiver 1110 receives information from the processing system 1114, specifically the transmission component 1010, and based on the received information, generates a signal to be applied to the one or more antennas 1120.

[0083] The processing system 1114 includes a processor 1104 coupled to a computer-readable medium / memory 1106. The processor 1104 is responsible for general processing, including the analyzation of data gathered by the apparatus itself through its own sensors and the execution of software stored on the computer-readable medium / memory 1106. The software, when executed by the processor 1104, causes the processing system 1114 to perform the various functions described supra for any particular apparatus. The computer-readable medium / memory 1106 may also be used for storing data that is manipulated by the processor 1104 when executing software. The processing system 1114 further includes at least one of the components 1004, 1006, 1008, 1010. The components

may be software components running in the processor 1104, resident/stored in the computer readable medium / memory 1106, one or more hardware components coupled to the processor 1104, or some combination thereof.

[0084] In the following, various aspects of this disclosure will be illustrated:

[0085] Example 1 is a method or apparatus for operating a motorized scooter. The apparatus may be a part of the motorized scooter. The apparatus may receive an unlock request from a remote server. The apparatus may unlock itself in response to the unlock request. The apparatus may continuously upload its status to the remote server based on a schedule when the apparatus is unlocked.

[0086] In Example 2, the subject matter of Example 1 may optionally include that the apparatus may further receive a lock request from the remote server, and lock itself in response to the lock request.

[0087] In Example 3, the subject matter of Example 2 may optionally include that an application program executing on a mobile device may receive a user input to lock the apparatus, where the application program may send a message to the remote server to request locking the apparatus in response to the user input, where the lock request may be generated in response to the message when the apparatus is in an unlocking status.

[0088] In Example 4, the subject matter of any one of Examples 1 to 3 may optionally include that an application program executing on a mobile device may scan a machine-readable optical label on the apparatus and send a vehicle identifier corresponding to the machine-readable optical label to the remote server, where the unlock request may be generated and transmitted based on the vehicle identifier.

[0089] In Example 5, the subject matter of Example 4 may optionally include that the unlock request may be generated when the motorized scooter is in normal usable status.

[0090] In Example 6, the subject matter of any one of Examples 1 to 5 may optionally include that the apparatus may further receive a speed control command from the remote server, and set the maximum speed of itself based on the speed control command.

[0091] In Example 7, the subject matter of any one of Examples 1 to 6 may optionally include that the apparatus may establish a wireless connection with the remote server based on a wireless communication identifier associated with the apparatus and a vehicle identifier of the apparatus.

[0092] Example 8 is a method or apparatus for managing motorized scooters. The apparatus may be a server. The apparatus may receive, from a mobile device, a vehicle identifier of a motorized scooter. The apparatus may retrieve the vehicle status of the motorized scooter based on the vehicle identifier. The apparatus may generate an unlock request when the vehicle status is a normal usable status. The apparatus may transmit the unlock request to the motorized scooter.

[0093] In Example 9, the subject matter of Example 8 may optionally include that the apparatus may further continuously receive, from the motorized scooter, the vehicle status of the motorized scooter based on a schedule.

[0094] In Example 10, the subject matter of any one of Examples 8 to 9 may optionally include that the apparatus may further: receive a message from the mobile device, the message requesting locking the motorized scooter; generate a lock request in response to the message when the motorized scooter is in an unlocking status; and transmit the lock request to the motorized scooter.

[0095] In Example 11, the subject matter of any one of Examples 8 to 10 may optionally include that an application program executing on the mobile device may scan a machine-readable optical label on the motorized scooter and send the vehicle identifier corresponding to the machine-readable optical label to the apparatus.

[0096] In Example 12, the subject matter of any one of Examples 8 to 11 may optionally include that the apparatus may further: generate a speed control command limiting the maximum speed of the motorized scooter based on the vehicle status of the motorized scooter; and transmit the speed control command to the motorized scooter.

[0097] In Example 13, the subject matter of any one of Examples 8 to 12 may optionally include that the apparatus may further establish a wireless connection with the motorized scooter based on a wireless communication identifier associated with the motorized scooter and the vehicle identifier of the motorized scooter.

[0098] In Example 14, the subject matter of any one of Examples 8 to 13 may optionally include that the apparatus may further: determine whether there is an active order associated with the motorized scooter when the motorized scooter is unlocked; and transmit a lock request to the motorized scooter when there is no active order associated with the motorized scooter and the motorized scooter is unlocked.

[0099] In Example 15, the subject matter of any one of Examples 8 to 14 may optionally include that the apparatus may further: determine whether an order associated with the motorized scooter is active when the motorized scooter is locked; and terminate the order when the order is active and the motorized scooter is locked.

[00100] In Example 16, the subject matter of any one of Examples 8 to 15 may optionally include that the apparatus may further: determine a time instance to upgrade a firmware of the motorized scooter based on the vehicle status of the motorized scooter; and upgrade the firmware of the motorized scooter at the determined time instance.

[00101] It is understood that the specific order or hierarchy of blocks in the processes / flowcharts disclosed is an illustration of exemplary approaches. Based upon design preferences, it is understood that the specific order or hierarchy of blocks in the processes / flowcharts may be rearranged. Further, some blocks may be combined or omitted. The accompanying method claims present elements of the various blocks in a sample order, and are not meant to be limited to the specific order or hierarchy presented.

[00102] The previous description is provided to enable any person skilled in the art to practice the various aspects described herein. Various modifications to these aspects will be readily apparent to those skilled in the art, and the generic principles defined herein may be applied to other aspects. Thus, the claims are not intended to be limited to the aspects shown herein, but is to be accorded the full scope consistent with the language claims, wherein reference to an element in the singular is not intended to mean “one and only one” unless specifically so stated, but rather “one or more.” The word “exemplary” is used herein to mean “serving as an example, instance, or illustration.” Any aspect described herein as “exemplary” is not necessarily to be construed as preferred or advantageous over other aspects. Unless specifically stated otherwise, the term “some” refers to one or more. Combinations such as “at least one of A, B, or C,” “one or more of A, B, or C,” “at least one of A, B, and C,” “one or more of A, B, and C,” and “A, B, C, or any combination thereof” include any combination of A, B, and/or C, and may include multiples of A, multiples of B, or multiples of C. Specifically, combinations such as “at least one of A, B, or C,” “one or more of A, B, or C,” “at least one of A, B, and C,” “one or more of A, B, and C,” and “A, B, C, or any combination thereof” may be A only, B only, C only, A and B, A and C, B and C, or A and B and C, where any such combinations may contain one or more member or members of A, B, or C. All structural and functional equivalents

to the elements of the various aspects described throughout this disclosure that are known or later come to be known to those of ordinary skill in the art are expressly incorporated herein by reference and are intended to be encompassed by the claims. Moreover, nothing disclosed herein is intended to be dedicated to the public regardless of whether such disclosure is explicitly recited in the claims. The words “module,” “mechanism,” “element,” “device,” and the like may not be a substitute for the word “means.” As such, no claim element is to be construed as a means plus function unless the element is expressly recited using the phrase “means for.”