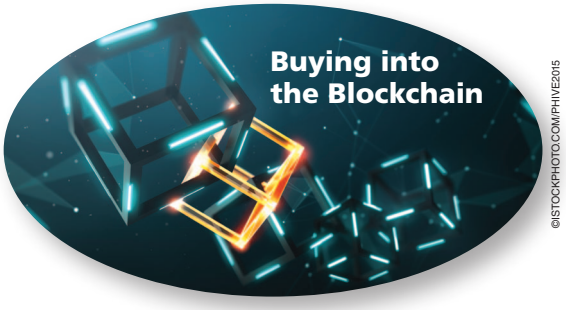

Blockchains Can Work for Car Insurance

Using smart contracts and sensors to provide on-demand coverage.

By Fabrizio Lamberti, Valentina Gatteschi, Claudio Demartini, Matteo Pelissier, Alfonso Gómez, and Victor Santamaria

Digital Object Identifier 10.1109/MCE.2018.2816247
Date of publication: 13 June 2018

BLOCKCHAINS AND SENSORS INSTALLED ON A VEHICLE COULD be combined to semiautomatically activate/deactivate car insurance coverage in an envisaged on-demand insurance scenario. We present a prototype that includes a mobile application (app) and a portable electronic device to be installed onboard. The mobile app lets the driver dynamically change the status of specific insurance coverage (in some cases, after pictures of the vehicle have been taken to

2162-2248/18©2018IEEE

attest to its conditions). Each modification and picture hash (a fixed-length alphanumeric summary of data content) are saved on the blockchain within a smart contract to certify changes made as well as the vehicle's status. Sensors embedded in the electronic device are used to collect passengers' and the vehicle's data. Data are then used to automatically modify insurance coverage based on car/environment conditions and the preferences set. The proposed solution could help lower policy modification costs and limit insurance fraud.

THE ULTIMATE CONSUMER ELECTRONICS PRODUCT

During the last couple of years, cars have started to be regarded as "the ultimate consumer electronics (CE) product" [1], mainly because of the recent research efforts and advancements in the development of autonomous vehicles [2], [3]. During this time, the wider public has also come to recognize the potential of blockchain, which is regarded as a disruptive technology able to bring tremendous changes to existing processes. A blockchain is a public, decentralized ledger managed by a peer-to-peer network, which records transactions among network nodes [4]. Past transactions cannot be modified and could be inspected by every node. The blockchain uses a digital signature mechanism in which the issuer of a transaction uses his or her private key, stored in a wallet, to sign a message broadcast to the network to guarantee the integrity, authenticity, and reliability of transactions made.

Even though the blockchain was created initially to keep track, in a decentralized way, of financial transactions (bitcoins and, later, other cryptocurrencies), researchers began to devise solutions to record different types of information (e.g., images and text messages) or even store and run programs, the so-called smart contracts [5]. A smart contract is an autonomous piece of code saved on the blockchain (hence, the code is immutable) that is programmed to behave in a defined manner when certain conditions are met. From the technical point of view, the code for smart contracts can contain variables and functions. When a programmer "publishes" (i.e., deploys) a smart contract on the blockchain, it becomes accessible by every node of the network through a unique address. By sending transactions to this address, anyone can invoke the smart contract's functions and inspect values recorded in its variable. Many companies are currently investigating blockchain technology and developing prototype solutions in different sectors, such as the Internet of Things (IoT) [6], [7], supply chain management [8], and autonomous vehicles [9].

To actively contribute to this movement, we propose a solution for an on-demand insurance service that combines blockchain technology and sensors installed on a vehicle to 1) semiautomatically modify car insurance coverage, 2) certify a coverage's activation/deactivation, and 3) attest to a vehicle's status at a given time. In particular, a prototype has been created that includes a mobile app and a portable electronic device to be installed onboard. By using the mobile app, drivers can dynamically activate/deactivate coverage against passenger injury as well as theft, fire, and weather events. Modifications are saved on the blockchain in a smart contract to certify changes have been made. In addition, for some coverage (theft, fire, and weather events), the driver is required to use the mobile app for taking photos of the vehicle. Each photo's hash is recorded in the blockchain together with the aforementioned information. In this way, the insurance company has proof that the vehicle was not damaged at the time of coverage activation. The electronic device gathers, through several sensors, information about the car location, the number of passengers, and the status of safety belts and uses these data to automatically modify insurance coverage based on car/environment conditions and the preferences set.

Sensors embedded in the electronic device are used to collect passengers' and the vehicle's data.

The proposed solution is meant to complement traditional insurance practices and could help lower policy modification costs and reduce fraud. In fact, in a traditional insurance scenario, coverage changes are recorded with a formal modification to the contract made in the presence of the insurer. With the proposed solution, costs could be cut since customers may directly modify coverage by interacting with a smart contract. To address the issue of fraud, the electronic device periodically gathers data from the vehicle and stores them (together with location data and pictures taken by the customer before each coverage activation) in an immutable way, providing the insurer with a proof of the vehicle's state before the occurrence of an insured event.

Lessons learned could be easily extended to other services, such as peer-to-peer insurances, where blockchain could be successfully used to build decentralized autonomous organizations. Other application scenarios could be envisaged for the devised solution. For instance, in on-demand home insurances, a home hub communicating with different sensors could be used to dynamically activate coverage, detect damages, automatically ask for intervention/refunds, and so forth.

RELATED WORKS

Among the advantages of blockchain technology, probably the most significant ones are transparency and automation. Transparency is linked to the fact that everyone can inspect the blockchain and that transactions cannot be repudiated. Automation is enabled by smart contracts and could be particularly relevant in an IoT context [6], [7]. Similar to other scenarios, CE could benefit from the adoption of this technology, as proven by the increasing interest by the field's experts [10], [11]. In fact, blockchain could complement existing electronic payment means [12] by enabling peer-to-peer payments without the need for an intermediary. In smart homes, blockchain could be used to enable intelligent appliances to automatically order or pay for spare parts, when damaged, or as a foundation layer

where devices can bargain energy [9] for optimized energy consumption [13]. Smart cities and their different components, such as smart energy, smart transportation, and smart health care [14], could benefit as well. In smart energy, the blockchain may be used to enable energy trading among neighbors. In smart transportation, it could allow intelligent vehicles to pay for fast lanes. In smart health care, the blockchain may be used as a shared ledger recording patients' medical histories. Finally, the blockchain could successfully enable the traceability of CE products [8], e.g., to identify counterfeit items.

The proposed work aims to address one of the emerging fields of CE, in-vehicle technology, and investigate how the joint use of sensors embedded in an electronic device installed onboard and of blockchain could support on-demand insurance. Several projects have examined the added value of blockchain in the context of in-vehicle technology, intelligent (autonomous) vehicles and, in general, mobility. The Oaken project [17] proposes a solution for allowing vehicles to autonomously pay tolls using cryptocurrencies. The work reported in [9] suggests a similar approach, where a vehicle could use the blockchain for purchasing electricity. The La'Zooz project [18] has a more ambitious goal, planning to be the "blockchained version of Uber." The Dovu project [19] focuses on collecting mobility data using vehicles' sensors and exploiting the blockchain to reward users based on shared data.

Blockchain technology has been extensively investigated in the insurance field [15]. The Dynamis project [20] provides peer-to-peer unemployment insurances based on a user's LinkedIn profile data. The InsurETH project [21] uses smart contracts to automatically refund insured customers in case of flight delays. The LenderBot project [16] focuses on microinsurance and records data related to loaned items on the blockchain. Finally, the EverLedger [22] initiative records diamond data on the blockchain to reduce jewelry fraud.

Similar to the Dynamis, InsurETH, and LenderBot projects, we propose using the blockchain to record undersigned policies. Nonetheless, in these initiatives policies could only be activated manually by the users, while our approach relies on sensors embedded in an electronic device to be installed onboard to make coverage activation/deactivation semiautomatic. We propose storing on the blockchain additional vehicle data, such as the vehicle's pictures hash. In this way, the vehicle status at any given time can be certified and inspected by the insurer before paying claims, helping to reduce the occurrence of fraud.

PROPOSED SYSTEM

The proposed system is composed of a mobile app and a portable electronic device to be installed on the vehicle. The app is used to manually activate/deactivate coverage, whereas the electronic device is used to enable changes in automatic coverage based on the number of passengers onboard and the vehicle's location. Figure 1 depicts the architecture of the system. Two parts can be distinguished, corresponding to two phases, i.e., policy undersigning and policy inspection/modification.

During policy undersigning, the customer can make a request for a new on-demand policy using the mobile app. The server creates a new insurance smart contract and deploys it on the Ethereum blockchain (a well-known blockchain supporting smart contracts). After deployment, the customer is requested to

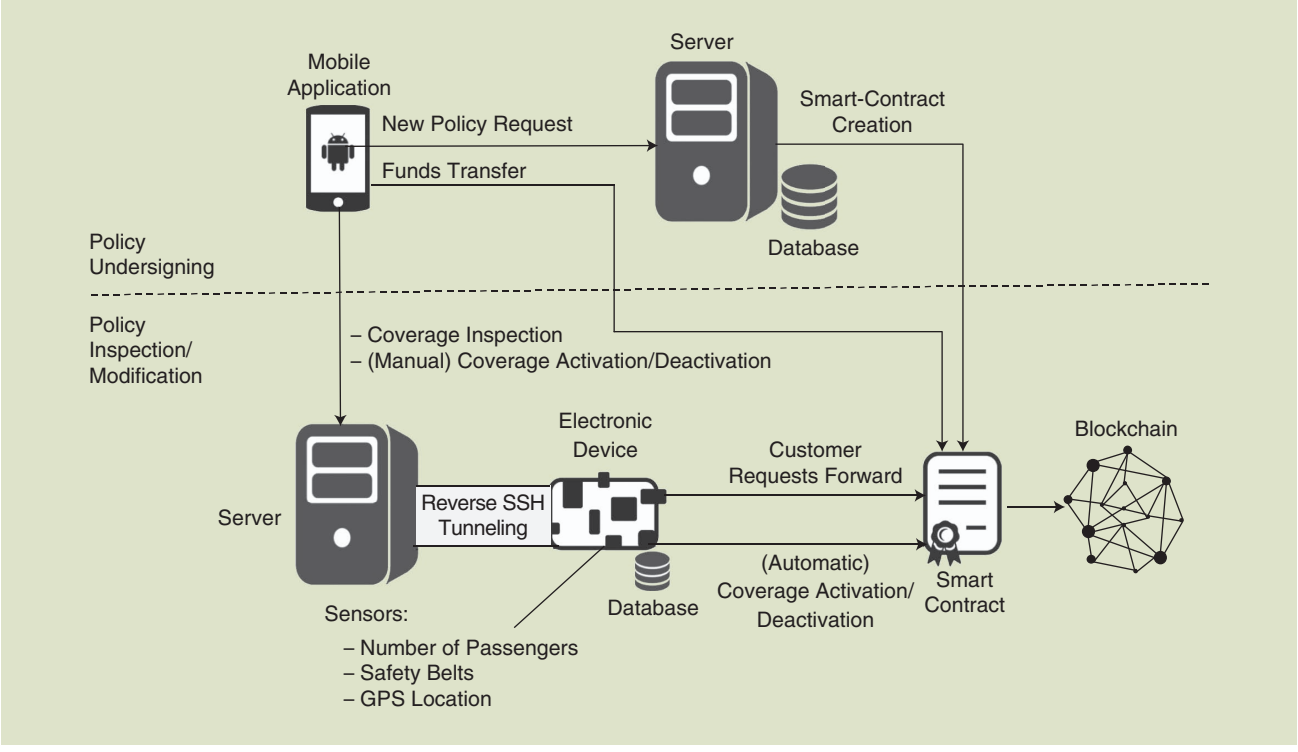

FIGURE 1. The architecture of the system. GPS: global positioning system; SSH: secure shell.

make a transfer from his or her wallet to the smart contract to provide funds for the policy. In the prototype created, Ethers, i.e., the cryptocurrencies of the Ethereum blockchain, are used. The smart contract stores the sender's address to eventually transfer the money to him or her. The customer's data and other information, such as the address of the newly created smart contract, are stored in a database. In this phase, a portable electronic device is assigned to the customer to be installed on the customer's vehicle. This device acts as a server and forwards any customer requests (e.g., activation/deactivation of coverage) to the blockchain. Hence, the device is provided with a wallet, which is used to sign transactions sent to the smart contract. It can also act on its own, e.g., based on data collected by embedded sensors, and trigger transactions autonomously.

The customer can interact with the electronic device by using the mobile app. The device is connected to the Internet network through a mobile subscriber identification module (SIM). Phone companies do not provide public Internet protocol addresses (which are needed to let the device receive requests by external applications). Therefore, a reverse secure shell (SSH) tunneling has been created between the electronic device and a specific port of a dedicated server, allowing incoming data to a server's port to be automatically redirected in a secure way.

Once the smart contract is deployed on the blockchain, the customer can interact with it (the policy inspection/modification phase in Figure 1). In this phase, the customer can inspect the active coverages and manually modify them if needed. The electronic device receives the customer's requests and eventually updates the smart contract.

To have a better view of the functionalities offered by the system, Figure 2 reports the use-case diagram for a customer. To undersign a policy, the customer has to specify personal data. After smart contract deployment, the customer can activate the policy by transferring some Ethers to it and then view the state of passengers, theft, fire, and weather events coverage. At that point, the customer can either schedule an automatic coverage modification (i.e., making the system change the coverage at/in a given time/place) or change the status of the coverage manually. To deal with conflicts between automatic and manual modification, when the customer triggers a new manual modification of a coverage, the previously defined automatic modification is discarded (and vice versa, letting recent modifications prevail over older ones).

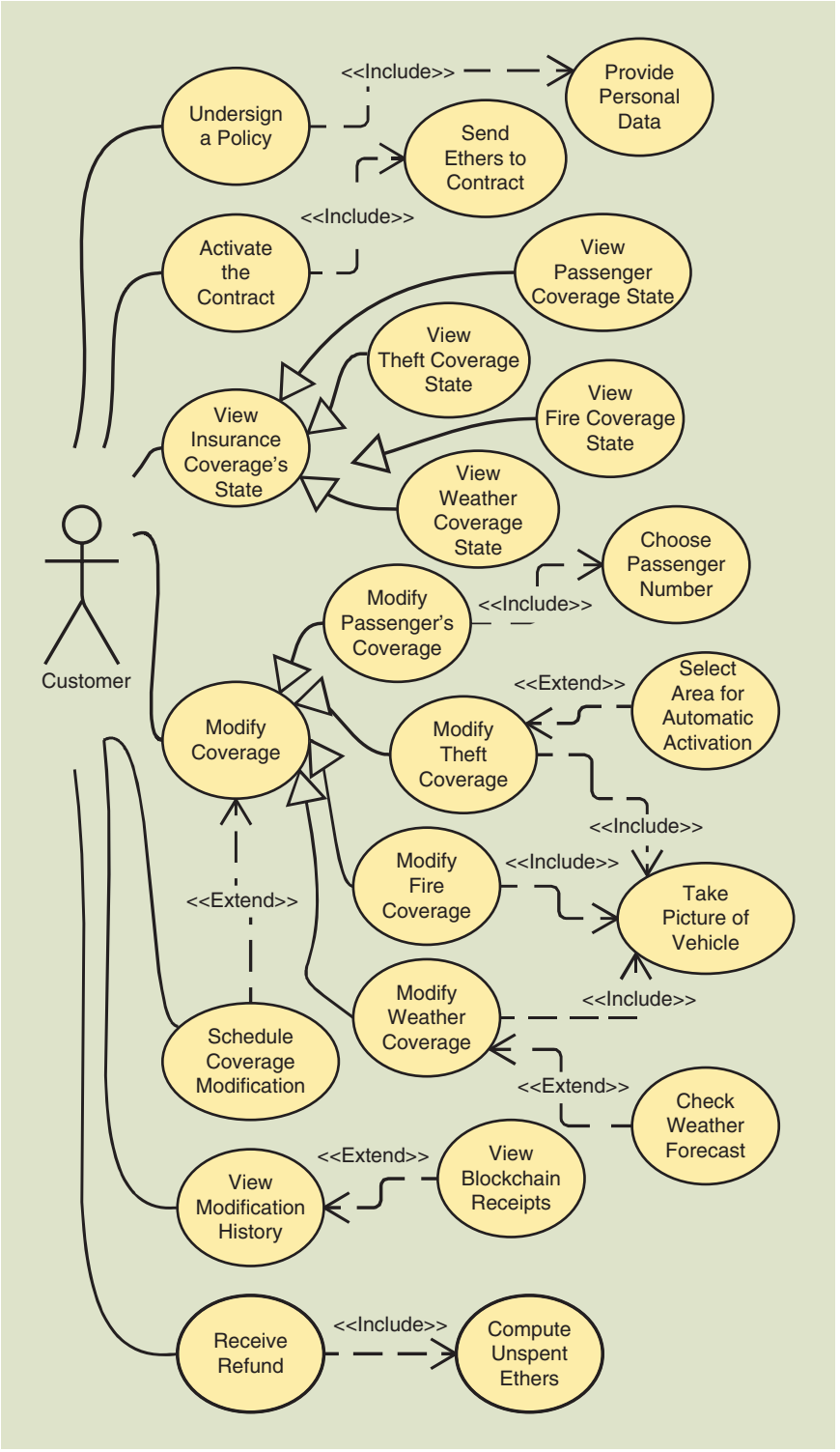

FIGURE 2. A use case diagram for the customer.

Many companies are currently investigating blockchain technology and developing prototype solutions in different sectors.

Depending on the type of coverage, different tasks are performed. For passengers' coverage modification, the customer must select the number of passengers to be insured against accidents, even though the electronic device subsequently performs a further check on the number of passengers onboard. Theft, fire, and weather events coverage can be activated only after pictures of the vehicle have been taken (to avoid fraud). The theft coverage can be programmed to be automatically modified in certain areas. For weather events, however, the customer can eventually inspect weather forecasts. The customer also can view the modification history and the receipts of transactions made or ask for a refund of unspent Ethers.

SMART CONTRACT

The smart contract is responsible for storing the state of a policy and recording changes to coverage. It can be used to certify changes as follows: when the customer wants to modify a coverage, he or she sends a message (i.e., a transaction), signed with the customer's private key, to the smart contract's address. The smart contract receives the message, stores information on the coverage to be changed, and simultaneously records a timestamp. The certification of a coverage change is provided by

- the fact that the customer signed the message with a private key (certifying that the message was sent from the customer's wallet)
- the timestamp assigned by the smart contract (guaranteeing that a change was made at a given time)
- the fact that blockchain transactions are public and cannot be deleted (everyone could inspect the history of coverage changes).

Since users sending a transaction have to pay (using Ethers previously transferred to the electronic device) some gas (a tax computed on the basis of the amount of requested data stored on the blockchain or computational resources required to run a smart contract's function), we decided to limit the amount of data stored on the smart contract in the following ways:

- For passengers' coverage, changes are recorded on the smart contract when the number of passengers (detected by the electronic device) varies or after a manual activation by the customer (in case of conflicts between the number of passengers specified by the customer and the number retrieved by the electronic device, the highest value is considered). Information related to the correct use of safety belts is stored on the device database and replicated on the company's servers (to have a backup copy in case the device is damaged).
- For theft coverage, the smart contract changes are triggered manually or automatically (in this case, the electronic device monitors the vehicle position and interacts with the smart contract to record the change only when the vehicle leaves or enters the area specified by the customer). GPS coordinates are stored on the device/company's database.
- For theft, fire, and weather events coverage, the smart contract stores the hash of pictures taken before coverage activation, and pictures are stored on the device/company's database.

In general, in the Ethereum blockchain, transactions for which users paid a higher gas are executed more quickly than others. For example, on the Ethereum blockchain, Ether transactions generally cost between US$0.001 and US$0.5 and are executed in fewer than 10 min (the cheaper ones) or 15 s (the more expensive ones), respectively. In some extraordinary situations, which are not under the control of the user (e.g., in the case of a high number of transaction on the network), transaction execution time could be longer. Based on all of these elements, we decided to let the customer specify the amount he or she is willing to spend for a transaction (hence, how fast the modification will be recorded) using the mobile app. Alternatively, to reduce latency, a private blockchain connecting the insurance company and its customers could be built. Insurance staff can check information related to the insurance contract at any time by reading data from the smart contract and the electronic device's or company's databases.

The following is a list of the functions defined in the smart contract. In the devised architecture, the insurance smart contract is created by the insurance server. During creation, some additional information is provided and stored on the contract, such as the duration of the policy and the address of the electronic device installed on the vehicle (i.e., the policy's owner). At the end of the creation process, an address is assigned to the deployed smart contract for future interactions.

As soon as the customer makes an Ether transaction from his or her personal wallet (providing funds to the smart contract), the `activate()` function is invoked. The function identifies the sender of the request (i.e., the address of the customer's wallet) and stores this information on the smart contract as the financier of the policy and the recipient of future refunds. In addition, the function also records the policy activation time. The distinction between the owner (whose private keys are stored in the electronic device) and the financier (whose private keys are managed by the customer) has been made to increase the security of the system since, in the case of hacker attacks to the system, only the owner's wallet (having a limited balance) would be compromised. If needed, the customer can eventually send other Ethers during the policy's lifetime to provide additional funds to the smart contract (e.g., in case the customer wants to pay for the policy on a monthly basis).

Each time a coverage is changed, the `changeState()` function is called. This function checks the validity of the policy (e.g., if it is active and not expired) as well as the

sender of the message (as only the owner is allowed to change the state). If the aforementioned checks are successful, the function stores the new state and the timestamp of the modification. The timestamps of each activation or deactivation will be used at a later stage to compute the amount to be paid to the insurance company.

As mentioned previously, during the activation of theft, fire, and weather events coverage, the customer is required to take several pictures of the vehicle (in particular, photos of the front and back showing the license plate, plus both sides and the top of the vehicle). Photos can be taken using the mobile app only to avoid the upload of pictures shot at a different time. Because photos are checked by the insurance staff during claims to assess the vehicle's state prior to coverage activation, an alert reminds the customer to take pictures in a well-lighted place and to avoid out-of-focus images (even though, in the future, image recognition techniques could be used to automatically give feedback to the customer on the quality of the acquired files).

Once each photo has been taken, the mobile app converts it into a Hex string and sends it to the electronic device (through the SSH protocol, enabling a secure channel between the mobile app and the device). The electronic device computes the hash of the Hex string by using the MD5 algorithm. We chose this algorithm to limit the amount of data stored on the contract because the MD5 algorithm produces a 16-byte summary of picture's data, though other (stronger) hashing algorithms could be exploited as well. The hash is stored in the smart contract by means of the `setPictureHash()` function, which also records the related coverage change and a timestamp, acting as a "proof of existence" of the picture at a given time. Pictures are stored on the electronic device and the company's database, but other decentralized solutions, e.g., leveraging the InterPlanetary File System, could be adopted going forward. To reduce the amount of data transferred by the customer to the system and expedite the coverage change process, we decided to rely only on photos for certifying the state of a vehicle. Nonetheless, using a similar approach, videos could also be stored.

The insurance company can withdraw money from the smart contract by using the `withdrawToInsurance()` function. This function evaluates the amount spent by the customer for the active coverage and transfers the resulting Ethers to the insurance company's wallet. Similarly, the customer can trigger a withdrawal of an unspent balance with the `withdrawToCustomer()` function. For security reasons, the `killContract()` function was coded. This function, which is called by the insurance company, is meant to clear smart contract data and transfer spent Ethers to the insurance company or unspent Ethers to the customer. To read smart contract's variables modified with the functions previously described, several getter functions also have been developed.

ELECTRONIC DEVICE

The electronic device is meant to be installed onboard. Its objective is to detect the number of passengers, the connection of their safety belts, and the vehicle's location. It has been devised to make coverage changes automatically, easing the work required by customers and reducing insurance fraud. The architecture of the device is shown in Figure 3. The prototype created is based on a Raspberry Pi 3 model B [23], a single-board computer, which has been used to control three sensing devices.

> The smart contract is responsible for storing the state of a policy and recording changes to coverage.

- To detect the number of onboard passengers, the Omron D6T [24] sensor has been selected. This module is a noncontact infrared thermal sensor using the microelectro-mechanical systems sensing technology. It has been chosen since it is able to identify the presence of stationary humans by detecting their body heat, in contrast with the typical pyroelectric human presence sensors that rely on motion detection. In addition, it can measure the temperature of an entire area in a contactless way, unlike standard thermal sensors that need a contact point. It is connected to the Raspberry Pi and mounted on the front of the electronic device to acquire data from the vehicle's interior.
- To detect the connection of safety belts, Hall-effect-based sensors have been used. These sensors are transducers that vary their output voltage in response to a magnetic field. By connecting them to each safety belt (mounting a magnet on the belt's buckle and the sensor on its tongue), it is possible to know if a person complies with the road safety rules. In the prototype architecture, these sensors are connected to the Raspberry Pi.
- To detect the location of the vehicle, the Adafruit Ultimate Breakout GPS [25] has been used. The GPS is connected to the Raspberry Pi.

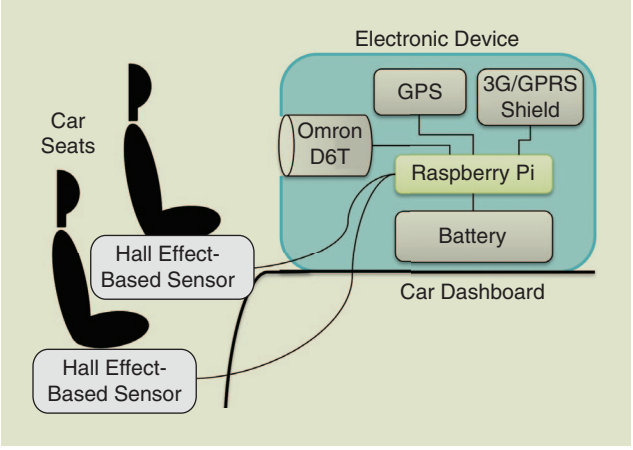

FIGURE 3. The architecture of the electronic device. 3G: third generation; GPRS: General Packet Radio Service.

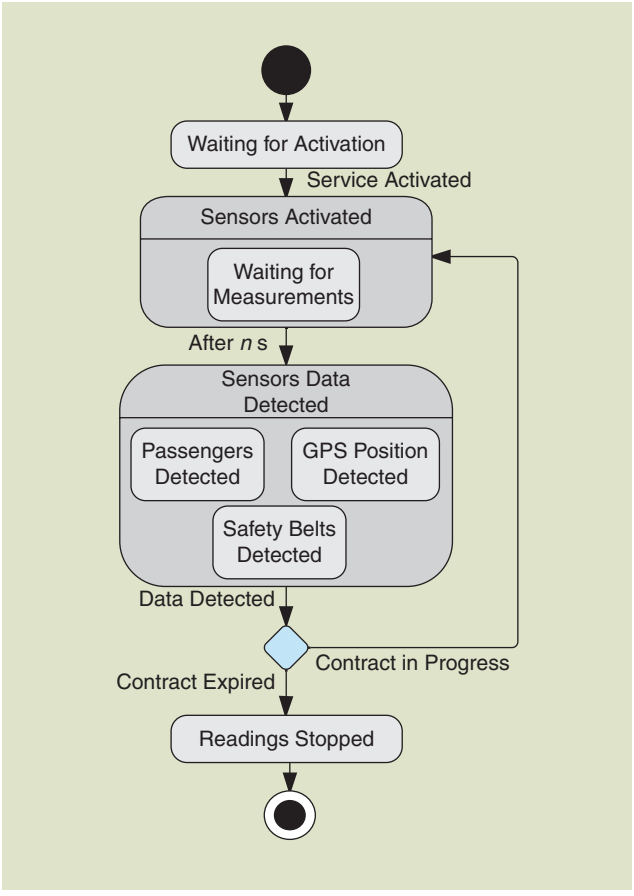

FIGURE 4. A state chart diagram of the device.

In addition to these modules, the Raspberry Pi has been equipped with the 3G/GPRS shield [26], enabling 3G connection through a mobile SIM, and with the Powerbank RS Pro battery [27]. Some of the described functionalities are available in black boxes installed by insurance companies or may be accessed through existing control units. However, our goal was to develop a prototype of a portable retrofitting device integrating all functionalities, which is not yet available.

Figures 4–6 depict the state chart diagram of the electronic device and the activity diagrams for automatic changes detection and coverage activations. To gather data from the sensors, several Python scripts have been written. Once sensors are activated, some scripts continuously check data coming from them to detect changes with respect to the smart contract configuration. Data are gathered until the contract expires. Currently, data are extracted every 5 s, even though the insurance company could set a different interval.

In particular, as soon as the number of passengers varies, a script invokes the `changeState()` function of the smart contract to modify passengers' coverage. Similarly, should the vehicle leave or enter a predefined area (set by the customer through the mobile app), another script triggers the function to modify the associated coverage. The location and safety belt connection data are periodically saved on the device or company's database.

USAGE SCENARIO

Figure 7(a) depicts the app main screen. From here, the customer can see an overview of the state of each coverage.

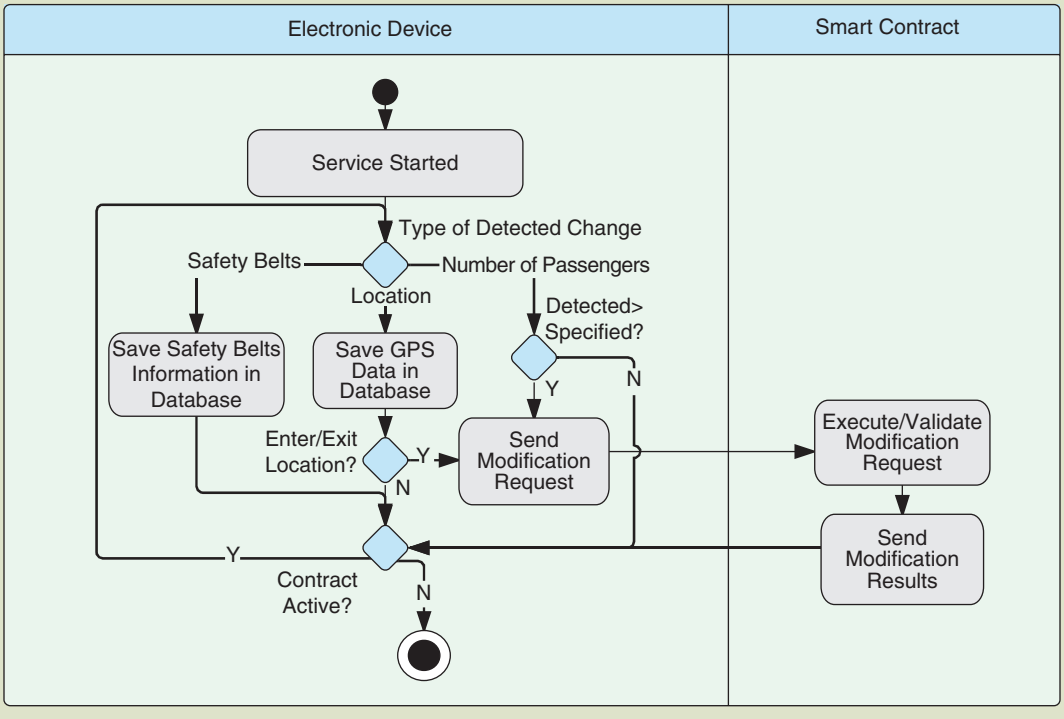

FIGURE 5. An activity diagram for changes detection.

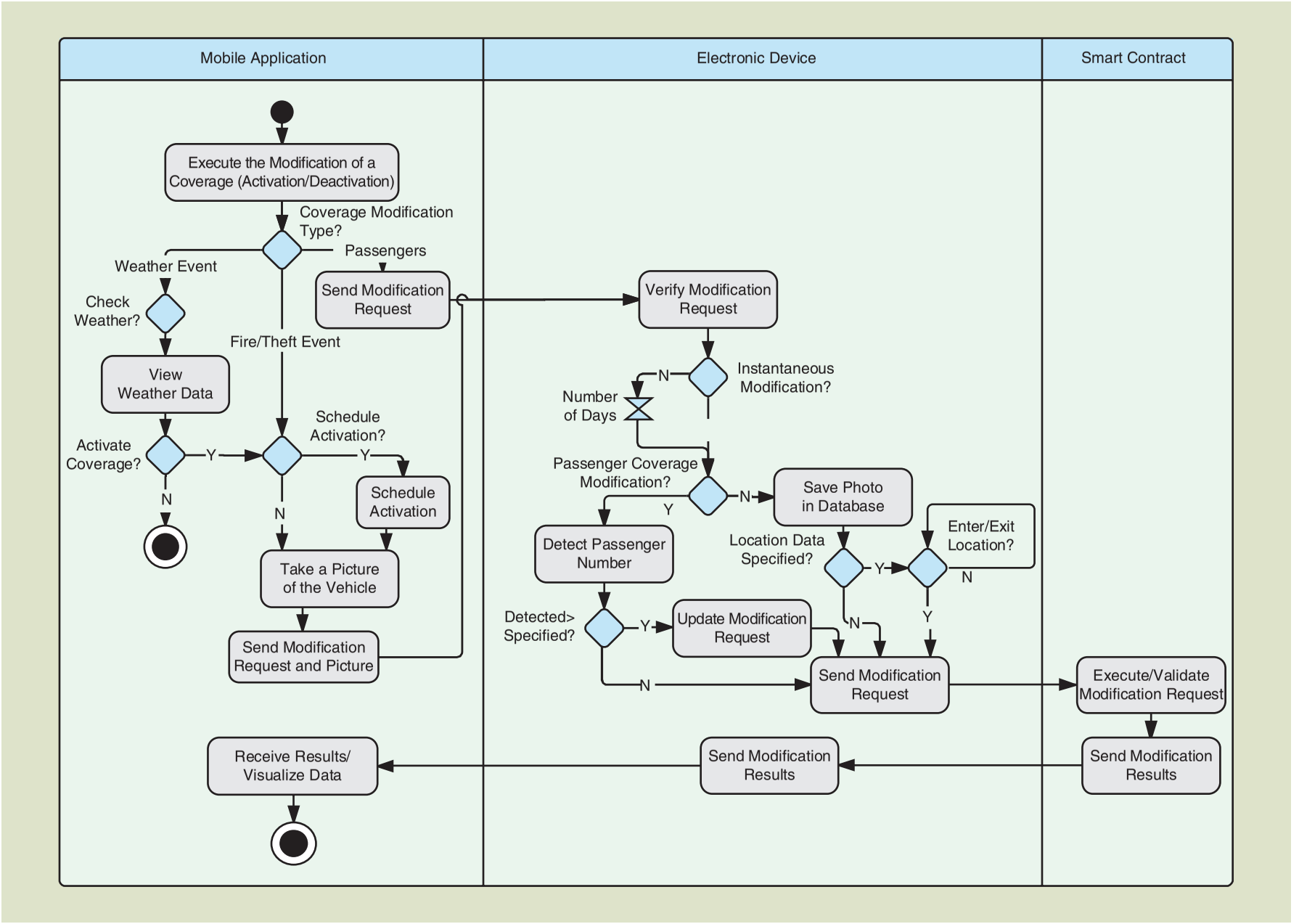

FIGURE 6. An activity diagram for activation of coverage.

When the customer clicks on "passengers' covers," he or she can add or remove passengers [Figure 7(b)]. At this stage, the electronic device checks whether the number of passengers indicated by the customer corresponds to the number of passengers onboard (as detected by the electronic device) and modifies the coverage by specifying the highest value between inserted/detected ones.

For each coverage, the customer can get additional information, e.g., about the amount of time the coverage has been active, the date and time of the last modification, and the total amount of Ethers saved while the coverage was not active [Figure 7(c)]. When the customer decides to modify the coverage [Figure 7(d)], a transaction is made toward the smart contract address by invoking the `changeState()` function. Details such as transaction hash, block number, block hash, used gas, and information about the specific modification performed are shown in the mobile app [Figure 7(e)]. Coverage can also be automatically modified by exploiting the GPS sensor of the electronic device. To this purpose, the customer can schedule coverage modification and specify the area where the coverage (e.g., a theft cover) should be automatically activated/deactivated [Figure 7(f)]. In case of theft (especially if a

FIGURE 7. Screenshots of the prototype app: (a) the main window, (b) the passenger's coverage modification, (c) fire coverage details, (d) deactivation of the fire coverage, (e) details on the executed transaction, and (f) a selection of areas for automatic coverage change.

thief should steal the insured vehicle and then enter/exit from an area specified for automatic coverage change, thus deactivating/activating the coverage), the insurance staff could cross the police report data with the smart contract activations and location history to verify the state of the coverage when the theft happened. Location data history could also potentially be used to find the vehicle.

CONCLUSION

In this article, we have shown how a system for on-demand insurance could be realized using a smart contract and sensors data. A prototype has been created comprising a mobile app and an electronic device to be installed on customers' vehicles. The app is used to manually modify policy coverage by interacting with the smart contract to schedule automatic modifications and take photos of the vehicle to certify its state. The electronic device monitors environmental conditions and triggers modifications.

With the proposed solution, policy modifications costs could be lowered because the activation/deactivation of a coverage could be directly controlled by the customer interacting with the smart contract. Fraud could also be reduced since the system would record vehicle's state at each change. Future work could be devoted to extend the proposed prototype, e.g., by empowering the electronic device with additional sensors. Sensors could be used, e.g., to detect damages, and smart contracts leveraged to automatically trigger reimbursements. Additionally, the application of the devised solution to other insurance contexts (e.g., home insurance) could be evaluated.

ABOUT THE AUTHORS

Fabrizio Lamberti (fabrizio.lamberti@polito.it) is an associate professor at Politecnico di Torino, Turin, Italy.

Valentina Gatteschi (valentina.gatteschi@polito.it) is a postdoctoral research assistant at Politecnico di Torino, Turin, Italy.

Claudio Demartini (claudio.demartini@polito.it) is a full professor at Politecnico di Torino, Turin, Italy.

Matteo Pelissier (matteo.pelissier@gmail.com) earned his M.S. degree in management engineering. He is a postgraduate student at Politecnico di Torino, Turin, Italy.

Alfonso Gómez (alfonso.gomez@realeites.com) is an information technology engineer with Technology and Digital Innovation at Reale Information Technology Engineering Service, Madrid, Spain.

Victor Santamaria (victor.santamaria@realeites.com) is with Technology and Digital Innovation at Reale Information Technology Engineering Service, Madrid, Spain.

REFERENCES

[1] A. Desi. (2017, Jan. 25). The evolution of consumer electronics: Autonomous vehicles and digital assistants. FlexEnable. [Online]. Available: http://www.flexenable.com/blog/the-evolution-of-consumer-electronics-autonomous-vehicles-and-digital-assistants/

[2] B. Markwalter, "The path to driverless cars," *IEEE Consum. Electron. Mag.*, vol. 6, no. 2, pp. 125–126, Apr. 2017.

[3] A. Munir, "Safety assessment and design of dependable cybercars: For today and the future," *IEEE Consum. Electron. Mag.*, vol. 6, no. 2, pp. 69–77, Apr. 2017.

[4] S. Nakamoto, "Bitcoin: A peer-to-peer electronic cash system," Bitcoin.org, White Paper, 2008.

[5] N. Szabo. (1994). Smart contracts 1994. Virtual School. [Online]. Available: https://archive.is/zQ1p8

[6] P. Corcoran, "The Internet of Things: Why now, and what's next?" *IEEE Consum. Electron. Mag.*, vol. 5, no. 1, pp. 63–68, 2016.

[7] K. Christidis and M. Devetsikiotis, "Blockchains and smart contracts for the Internet of Things," *IEEE Access*, vol. 4, pp. 2292–2303, May 2016.

[8] J.-H. Lee and M. Pilkington, "How the blockchain revolution will reshape the consumer electronics industry," *IEEE Consum. Electron. Mag.*, vol. 6, no. 3, pp. 19–23, 2017.

[9] T. Lundqvist, A. de Blanche, and H. R. H. Andersson, "Thing-to-thing electricity micro payments using blockchain technology," in *Proc. Global Internet of Things Summit (GIoTS)*, 2017, pp. 1–6.

[10] D. Puthal, N. Malik, S. P. Mohanty, E. Kougianos, and C. Yang, "The blockchain as a decentralized security framework," *IEEE Consum. Electron. Mag.*, vol. 7, no. 2, pp. 18–21, Mar. 2018.

[11] J. H. Lee and H. Kim, "Security and privacy challenges in the Internet of Things," *IEEE Consum. Electron. Mag.*, vol. 6, no. 3, pp. 134–136, 2017.

[12] S. Ghosh, A. Majumder, J. Goswami, A. Kumar, S. P. Mohanty, and B. K. Bhattacharyya, "Swing-pay: One card meets all user payment and identity needs: A digital card module using NFC and biometric authentication for peer-to-peer payment," *IEEE Consum. Electron. Mag.*, vol. 6, no. 1, pp. 82–93, 2017.

[13] E. Rodriguez-Diaz, J. C. Vasquez, and J. M. Guerrero, "Intelligent dc homes in future sustainable energy systems: When efficiency and intelligence work together," *IEEE Consum. Electron. Mag.*, vol. 5, no. 1, pp. 74–80, 2016.

[14] S. P. Mohanty, U. Choppali, and E. Kougianos, "Everything you wanted to know about smart cities: The Internet of Things is the backbone," *IEEE Consum. Electron. Mag.*, vol. 5, no. 3, pp. 60–70, 2016.

[15] F. Lamberti, V. Gatteschi, C. Demartini, C. Pranteda, and V. Santamaria. (2017). "Blockchain or not blockchain, that is the question of the insurance and other sectors," *IT Professional*. [Online]. Available: http://ieeexplore.ieee.org/document/7950836/

[16] J. Redman. (2016). LenderBot: A micro-insurance proof of concept by Stratumn. Bitcoin. [Online]. Available: https://news.bitcoin.com/stratumn-deloitte-blockchain-bot/

[17] Oaken Innovations. (2018). Oaken Innovations blockchain and IoT. [Online]. Available: https://www.oakeninnovations.com

[18] La'Zooz. (2018). New application for ride sharing. [Online]. Available: http://lazooz.org

[19] Dovu. (2018). [Online]. Available: https://dovu.io

[20] Dynamis. (2018). Introducing Dynamis. [Online]. Available: http://www.dynamis app.com

[21] InsurETH. (2018). [Online]. Available: http://insureth.mkvd.net

[22] Everledger. (2018). Welcome to the digital vault of the future. [Online]. Available: https://www.everledger.io

[23] Raspberry Pi Foundation. (2018). Raspberry Pi model B. [Online]. Available: https://www.raspberrypi.org/products/raspberry-pi-3-model-b

[24] Omron. (2017). D6T MEMS thermal sensors. [Online]. Available: https://www.omron.com/ecb/products/sensor/11/d6t.html

[25] Adafruit. (2018). Adafruit Ultimate GPS Breakout—66 channel with10 Hz updates—Version 3. [Online]. Available: https://www.adafruit.com/product/746

[26] Cooking Hacks. (2018). 3G/GPRS shield for Arduino. [Online]. Available: https://www.cooking-hacks.com/3g-gprs-shield-for-arduino-3g-gps

[27] RS Components. (2018). RS Pro PB-A5200 5000 mAh 5V power bank portable charger. [Online]. Available: http://uk.rs-online.com/web/p/power-banks/7757508