Blockchains can work for car insurance: Using smart contracts and sensors to provide on-demand coverage

On-demand Blockchain-based Car Insurance Using Smart Contracts and Sensors

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

The last couple of years witnessed the explosion of two major technological revolutions, with vehicles, that started to be regarded as the next frontier for artificial intelligence and blockchain, which began to be considered by the wider public for the great changes it could bring in a number of heterogeneous domains. In order to take part in these revolutions, in this paper we show how blockchain and sensors installed on a vehicle could be combined to (semi-)automatically activate/deactivate car insurance covers in an envisaged on-demand insurance scenario. We present a prototype, which includes a mobile app and a portable electronic device to be installed onboard. The mobile app lets the driver dynamically change the status of specific insurance covers (in some cases, after pictures of the vehicle have been taken to attest its conditions). Each modification (and picture hash) is saved on the blockchain within a smart contract, in order to certify changes made (and vehicle's status). Sensors embedded in the electronic device are used to collect passengers' and vehicle's data. Data are then used to automatically modify insurance covers based on car/environment conditions and preferences set. The proposed solution could help lowering policy modification costs and limiting insurance frauds.

1. INTRODUCTION

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 made in the development of autonomous vehicles [2], [3].
The same years have also witnessed the recognition, by the wider public, of the potentialities 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 (where the issuer of a transaction uses his/her *private key*, stored in a *wallet*, to sign a message broadcasted to the network) in order to guarantee the integrity, authenticity and non-repudiability of transactions made.
Even though, initially, the blockchain was created to keep track, in a decentralized way, of financial transactions (Bitcoins and, later, other crypto-currencies), as time passed researchers devised solutions to record different types of information (e.g., images, text messages, etc.) 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), which is programmed to behave in a defined manner when certain conditions are met. From the technical point of view, smart contracts' code could 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.
A high number of companies are currently investigating blockchain technology and developing prototype solutions in different sectors, such as Internet of Things (IoT) [6], [7], supply chain management [8], and autonomous vehicles [9].
To actively contribute to this movement, in this paper we propose a solution for an on-demand insurance service, which combines blockchain technology and sensors installed on a vehicle to a) (semi-)automatically modify car insurance covers; b) certify cover's activation/deactivation; c) attest vehicle's status at a given time. In particular, a prototype has been created, which includes a mobile app and a portable electronic device to be installed onboard. By using the mobile app, drivers can dynamically activate/deactivate covers against passengers' injury, theft, fire and weather events. Modifications are saved on the blockchain in a smart contract, to certify changes made. In addition, for some covers (theft, fire and weather events), the driver is required to use the mobile app for taking pictures of the vehicle. Pictures' hash (a fixed-length alphanumeric summary of data content) is recorded in the blockchain together with the above information. In this way, the insurance company can have a proof that the vehicle was not damaged at the time of cover 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 covers based on car/environment conditions and on preferences set.
The proposed solution is meant to complement traditional insurance practice and could help lowering policy modification costs and reducing frauds. In fact, in a traditional insurance scenario, cover changes are recorded with a formal modification to the contract made in presence of the insurer. With the proposed solution, costs could be cut since customers may directly modify covers by interacting with a smart contract. Concerning frauds, the electronic device periodically gathers data from the vehicle and stores them (together with location data and pictures taken by the customer before each cover activation) in an immutable way, thus providing the insurer with a proof of vehicle's state before the occurrence of an insured event.
Lessons learnt could be easily extended to other services, such as peer-to-peer insurances, where blockchain could be successfully used to build Decentralized Autonomous Organizations (DAOs). Other application scenarios could be easily

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 covers, detect damages and automatically ask for intervention/refunds, etc.
The rest of the paper is organized as follows: Section 2 presents how CE could benefit from blockchain and reports related works exploiting blockchain technology for (autonomous) vehicles and insurance. Section 3 presents the proposed solution, whereas Section 4 shows the expected use for it through a practical example. Lastly, conclusions are drawn in Section 5.

2. RELATED WORKS

Among all the advantages brought by 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]. Similarly 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 to rely on an intermediary. In smart homes, it could be used to enable intelligent appliances to automatically order/pay for spare parts, when damaged, or as a foundation layer where devices can bargain energy [9], for an optimized energy consumption [13]. Smart cities and their different components, such as smart energy, smart transportation, and smart healthcare [14] could benefit as well. In smart energy, the blockchain could be used to enable energy trading among neighbors. In smart transportation, it could allow intelligent vehicles to pay for fast lanes. In smart healthcare, the blockchain could be used as a shared ledger recording patients' medical history. Finally, it must be underlined that blockchain could be successfully used to 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 to investigate how the joint use of sensors embedded in an electronic device installed onboard and of blockchain could support on-demand insurance.
Several projects already investigated the added value of blockchain in the context of in-vehicle technology, intelligent (autonomous) vehicles and, more in general, of mobility. The Oaken project (https://www.oakeninnovations.com) proposes a solution for letting vehicles autonomously pay tolls using crypto-currencies. The work reported in [9] suggests a similar approach, where a vehicle could use the blockchain for purchasing electricity. The La'Zooz project (http://lazooz.org) has a more ambitious goal, as it plans to be the "blockchained version of Uber". The Dovu project (https://dovu.io) focuses on collecting mobility data using vehicles' sensors, and exploits the blockchain to reward users based on shared data.
In the insurance field, blockchain technology has been widely investigated [15]. The Dynamis project (http://www.dynamisapp.com) provides peer-to-peer unemployment insurances based on LinkedIn profiles data. The InsurETH project (http://insureth.mkvd.net) uses smart contracts to automatically refund insured customers in case of flight delays. The LenderBot project [16] focuses on micro-insurances, and records on the blockchain data related to lent items. Finally, the EverLedger (https://www.everledger.io) initiative records diamonds' data on the blockchain, to reduce jewelry frauds.
Similarly to the Dynamis, InsurETH and LenderBot projects, in this paper we propose to use the blockchain to record undersigned policies. Nonetheless, if in the above initiatives policies could only be activated manually by the users, our approach relies on sensors embedded in an electronic device to be installed onboard to make covers activation/deactivation (semi-)automatic. We propose to store on the blockchain also additional vehicle's data, such as vehicle's pictures hash. In this way, vehicle status at a given time can be certified and inspected by the insurer before paying claims, thus reducing the number of frauds occurring when a customer claims a damage happened before cover activation.

3. PROPOSED SYSTEM

As said, the proposed system is composed of a mobile app and of a portable electronic device to be installed on the vehicle. The app is used to manually activate/deactivate covers, whereas the electronic device is used to enable automatic covers changes based on number of passengers onboard and 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*.

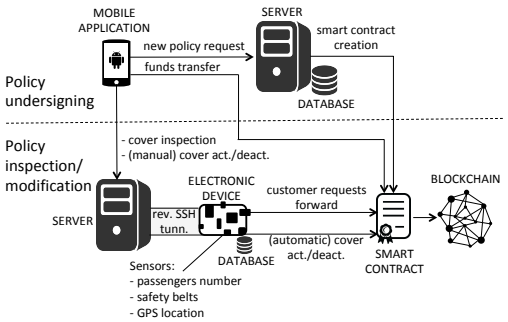

FIGURE 1. Architecture of the system.

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 make a transfer from his/her wallet to the smart contract, to provide funds for the policy. In the prototype created, Ethers, i.e., the crypto-currencies of the Ethereum blockchain are used. The smart contract stores the address of the sender to eventually transfer back the money to him or her. 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

his/her vehicle. This device acts as a server and forwards customer's requests (e.g., activation/deactivation of covers) to the blockchain. Hence, it 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. Since the electronic device is connected to the Internet network through a mobile SIM and given the fact that phone companies do not provide public IP addresses (which are needed to let the device receive requests by external applications), a reverse SSH tunneling has been created between the electronic device and a specific port of a dedicated server, letting all the incoming data to a server's port be automatically redirected in a secure way.

Once the smart contract is deployed on the blockchain, the customer can interact with it (*policy inspection/modification* phase in Figure 1). In this phase, he/she can inspect the active covers, and manually modify them if needed. The electronic device receives 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, he/she has to specify his/her personal data. After smart contract deployment, he/she can activate the policy by transferring some Ethers to it. Then, he/she can view the state of passengers, theft, fire and weather events covers. He/she can either schedule an automatic cover modification (i.e., making the system change the cover at/in a given time/place), or change the status of the cover manually. To deal with conflicts between automatic and manual modification, when the customer triggers a new manual modification of a cover, the previously defined automatic modification is discarded (and vice versa, letting recent modifications prevail over old ones). Depending on the type of cover, different tasks are performed. For passengers' cover modification, the customer has to 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 covers, instead, can be activated only after pictures of the vehicle have been taken (to avoid frauds). Theft cover can be programmed to be automatically modified in certain areas, whereas, for weather events, the customer can eventually inspect weather forecasts. He/she can also view the modification history and the receipts of transactions made, or ask for a refund of unspent Ethers.

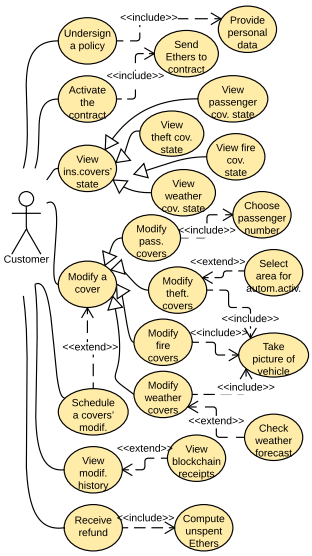

FIGURE 2. Use case diagram for the customer.

The next sub-sections provide more details on the smart contract and on the electronic device.

A. Smart Contract

The smart contract is responsible for storing the state of a policy and for recording changes to covers. It can be used to certify cover changes as follows: when the customer wants to modify a cover, he/she sends a message (i.e., a transaction), signed with his/her private key to the smart contract's address. The smart contract receives the message, stores information on the cover to be changed and simultaneously records a timestamp. The certification of a cover's change is provided by: a) the fact that the customer signed the message with his/her private key (certifying that the message was sent from his/her wallet), b) the timestamp assigned by the smart contract (guaranteeing that a change was made at a given time), c) the fact that blockchain transactions are public and cannot be deleted (everyone could inspect the history of cover changes).

Since users sending a transaction have to pay (using Ethers previously transferred to the electronic device) some *gas* (a tax computed on the the basis of the amount of data they want to store 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 particular:

- for passengers' cover, 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 safety belts correct use is stored on the device database, and is replicated on company's servers (to have a backup copy in case the device is damaged);
- for theft covers, the smart contract changes are triggered either 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/enters the area specified by the customer). GPS coordinates are stored on the device/company's database;
- for theft, fire and weather events covers, the smart contract stores the hash of pictures taken before covers activation, whereas 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 quicker than others. As a matter of example, on the Ethereum blockchain Ether transactions generally cost between $0.001 and $0.5, and are executed in less than 10 minutes (the cheaper ones) or 15 seconds (the more expensive ones), respectively. In some extraordinary situations, which are not under the control of the user (e.g., in case of a high number of transaction on the network), transaction execution time could be longer. Based on all of the above, we decided to let the customer specify the amount he/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.

At any time, insurance staff can check information related to the insurance contract, by reading data from the smart contract and the electronic device/company's databases.
In the following, a list of the functions defined in the smart contract is provided. 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/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 *financier* of the policy and recipient of future refunds. In addition, the function also records the policy activation time. It is worth remarking that the distinction between *owner* (whose private keys are stored in the electronic device) and *financier* (whose private keys are managed by the customer) has been made to increase the security of the system as, in 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 lifetime) to provide additional funds to the smart contract (e.g., in case he/she wants to pay for the policy on a monthly basis).
Each time a cover is changed, the `changeState()` function is called. This function checks the validity of the policy (e.g., if it is active and not expired yet) as well as the sender of the message (as only the *owner* is allowed to change the state). If the above checks are successful, the function stores the new state and the timestamp of the modification. The timestamps of each activation/deactivation will be used at a later stage to compute the amount to be paid to the insurance company.
As mentioned above, during the activation of theft, fire and weather events covers, the customer is also required to take several pictures of the vehicle (in particular, pictures of front and back showing also the license plate, plus both sides and top). Pictures can be taken only using the mobile app, to avoid the upload of pictures shot at a different time. Given the fact that pictures are checked by the insurance staff during claims to assess the vehicle's state prior to cover activation, an alert reminds the customer to take pictures in a well-lighted place and to avoid out of focus pictures (even though, in the future, image recognition techniques could be used to automatically give a feedback to the customer on the quality of the acquired image). Once each picture has been taken, the mobile app converts it into a Hex string and sends it to the electronic device (through the SSH protocol, thus 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 (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 cover 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 company's database, but other decentralized solutions, e.g., leveraging the InterPlanetary File System (IPFS), could be adopted in the future. It is worth saying that, in order to reduce the amount of data transferred by the customer to the system (to speed up the cover change process), it has been decided to rely only on pictures for certifying the state of a vehicle. Nonetheless, using a similar approach, also videos could 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 covers and transfers the resulting Ethers to the insurance wallet. Similarly, the customer can trigger a withdrawal of 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/unspent Ethers to the customer.
To read smart contract's variables modified with the above functions, several getter functions (not discussed in this paper due to space constraints) have been also developed.

B. 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 vehicle's location. It has been devised to make covers changes automatic, thus easing the work required by customers and at the same time, reducing insurance frauds.
The architecture of the device is shown in Figure 3. The prototype created is based on a Raspberry Pi 3 model B (https://www.raspberrypi.org/products/raspberry-pi-3-model-b), a single-board computer, which has been used to control three sensing devices.

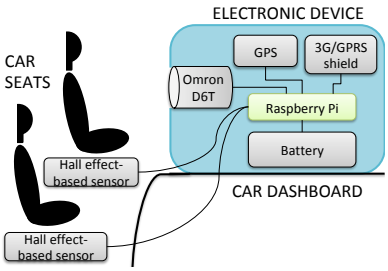

FIGURE 3. Architecture of the electronic device.

- To detect the number of onboard passengers, the Omron D6T (https://www.omron.com/ecb/products/sensor/11/d6t.html) sensor has been selected. This module is a non-contact infrared thermal sensor that uses the Micro-Electro-Mechanical Systems sensing technology. It has been chosen since it is able to detect 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 has been 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 becomes 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 (https://www.adafruit.com/product/746) has been used. The GPS is connected to the Raspberry Pi.

In addition to the above modules, the Raspberry Pi has been equipped with the 3G/GPRS shield (https://www.cooking-hacks.com/3g-gprs-shield-for-arduino-3g-gps), enabling 3G connection through a mobile SIM, and with the Powerbank RS Pro battery (http://uk.rs-online.com/web/p/power-banks/7757508). Indeed, some of the above functionalities are available already 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 retro-fitting device integrating them all, which is not available yet.

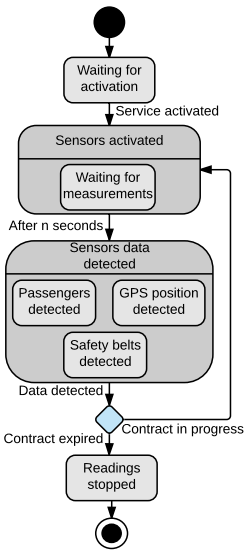

FIGURE 4. Statechart diagram of the device.

Figure 4, 5 and 6 depict the state chart diagram of the electronic device and the activity diagrams for automatic changes detection and covers' activation. 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 expiration of the contract. At the present time, data are extracted every 5 seconds, even though a different interval could be set by the insurance company.

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

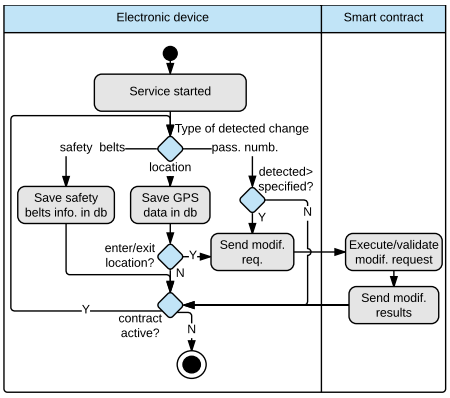

FIGURE 5. Activity diagram for changes detection

4. USAGE SCENARIO

In the following, a practical example is shown, where a customer exploits the mobile app to interact with the smart contract.

Figure 7a depicts the app main screen. From here, the customer can have an overview of the state of each cover. When he/she clicks on "passengers' cover", he/she can add/remove passengers (Figure 7b). 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 cover by specifying the highest value between inserted/detected ones.

For each cover, the customer can also get additional information, e.g., about the amount of time the cover has been active, the date and time of last modification, and the total amount of Ethers saved while the cover was not active (Figure 7c). When he/she decides to modify the cover (Figure 7d), a transaction is made towards 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 7e). Covers can also be automatically modified by exploiting the GPS sensor of the electronic device. To this purpose, the customer can schedule a cover modification and specify the area where the cover (e.g., a theft cover) should be automatically activated/deactivated (Figure 7f). In case of theft (especially, should a thief steal the insured vehicle and then enter/exit from an area specified for automatic cover change, thus deactivating/activating the cover), the insurance staff could cross police report's data with smart contract's activations and location history to verify the state of the cover when the theft happened. Location data history could also potentially be used to find the vehicle.

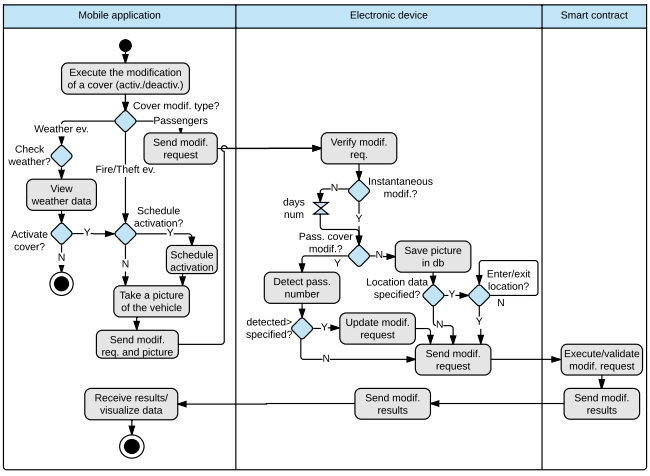

FIGURE 6. Activity diagram for covers' activation

5. CONCLUSIONS

In this paper 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, made up of a mobile app and of an electronic device to be installed on customers' vehicles. The app is used to manually modify policies' covers (by interacting with the smart contract), to schedule automatic modifications and to take pictures of the vehicle to certify its state. The electronic device monitors environment conditions and triggers modifications.

With the proposed solution, policy modifications costs could be lowered, as the activation/deactivation of a cover could be directly controlled by the customer by interacting with the smart contract. Frauds 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 could be leveraged to automatically trigger reimbursements. Additionally, the application of the devised solution to other insurance context (e.g., home insurance) could be evaluated.

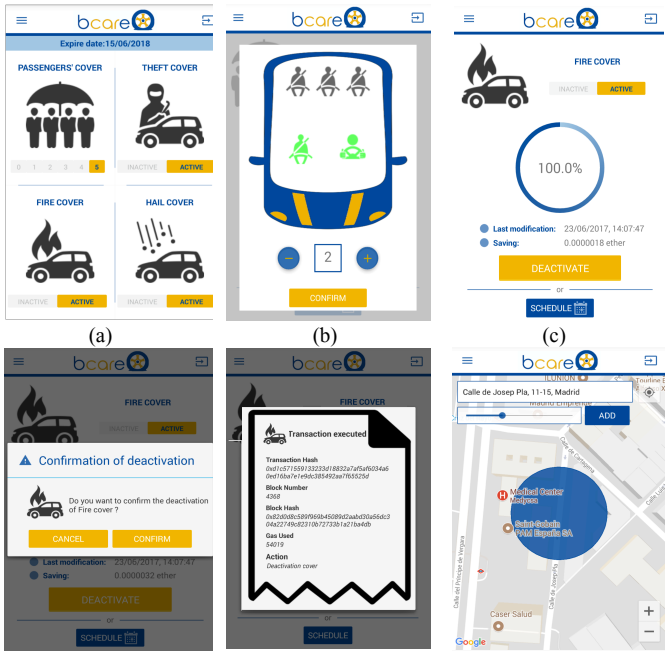

FIGURE 7. Screenshots of the prototype app: main window (a), passengers' cover modification (b), fire cover details (c), deactivation of fire cover (d), details on the executed transaction (e), selection of areas for automatic cover change (f).

ABOUT THE AUTHORS

Fabrizio Lamberti (fabrizio.lamberti@polito.it) is an Associate Professor at Politecnico di Torino, Italy.

Valentina Gatteschi (valentina.gatteschi@polito.it) is a Post-doctoral Research Assistant at Politecnico di Torino, Italy.

Claudio Demartini (claudio.demartini@polito.it) is a Full Professor at Politecnico di Torino, Italy.

Matteo Pelissier (matteo.pelissier@gmail.com) is a Post-graduate with a MS in Management Eng. at Politecnico di Torino, Italy.

Alfonso Gómez (alfonso.gomez@realeites.com) is an IT Engineer in Technology & Digital Innovation at Reale ITES, Spain.

Victor Santamaria (victor.santamaria@realeites.com) is Responsible for Technology & Digital Innovation at Reale ITES, Spain.

REFERENCES

[1] A. Desi, "The evolution of consumer electronics: Autonomous vehicles and digital assistants." [Online]. Available: http://www.flexenable.com/blog/the-evolution-of-consumer-electronics-autonomous-vehicles-and-digital-assistants/. [Accessed: 27-Sep-2017].

[2] B. Markwalter, "The Path to Driverless Cars [CTA Insights]," *IEEE Consumer Electronics Magazine*, 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 Consumer Electronics Magazine*, vol. 6, no. 2, pp. 69–77, Apr. 2017.

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

[5] N. Szabo, "Smart Contracts," 1994. [Online]. Available: http://www.virtualschool.edu/mon/Economics/SmartContracts.html.

[6] P. Corcoran, "The Internet of Things: Why now, and what's next?," *IEEE Consumer Electronics Magazine*, 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, 2016.

[8] J.-H. Lee and M. Pilkington, "How the Blockchain Revolution Will Reshape the Consumer Electronics Industry [Future Directions]," *IEEE Consumer Electronics Magazine*, 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 *Global Internet of Things Summit (GIoTS), 2017*, 2017, pp. 1–6.

[10] D. Puthal, N. . Malik, S. P. . Mohanty, E. Kougianos, and C. . Yang, "The Blockchain - A Decentralized Security Framework," *IEEE Consumer Electronics Magazine*, vol. 8, no. 1, 2018.

[11] J. H. Lee and H. Kim, "Security and Privacy Challenges in the Internet of Things [Security and Privacy Matters]," *IEEE Consumer Electronics Magazine*, 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 Consumer Electronics Magazine*, 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 Consumer Electronics Magazine*, 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 Consumer Electronics Magazine*, vol. 5, no. 3, pp. 60–70, 2016.

[15] F. Lamberti, V. Gatteschi, C. Demartini, C. Pranteda, and V. Santamaria, "Blockchain or not blockchain, that is the question of the insurance and other sectors," *IT Professional*, 2017.

[16] J. Redman, "LenderBot: A Micro-Insurance Proof of Concept by Stratumn," 2016. [Online]. Available: https://news.bitcoin.com/stratumn-deloitte-blockchain-bot/. [Accessed: 28-Sep-2017].