Smart Contract-based Car Insurance Policies

Lennart Bader*, Jens Christoph Bürger*
Communication and Distributed Systems
RWTH Aachen University
Germany
{lennart.bader1, jens.buerger}@rwth-aachen.de

Roman Matzutt, Klaus Wehrle
Communication and Distributed Systems
RWTH Aachen University
Germany
{matzutt, wehrle}@comsys.rwth-aachen.de

***Abstract*—Processes in the insurance economy are often cumbersome and expensive because of the inherently opposing interests of insurers and customers. Smart contracts bear a large potential to simplify these processes and thereby reduce costs. In this paper, we present CAIPY, our smart contract-based ecosystem for simple and transparent car insurance. In CAIPY, smart contracts do not replace but support current processes to enable significant cost savings, e.g., by removing the necessity for manual inspection of insurance claims in presence of tamper-resistant car sensors. However, the involved parties can resort to well-established processes at any time, trading off cost efficiency against process reliability. CAIPY thus showcases how smart contracts can support insurers without introducing new risks.**

***Index Terms*—Blockchain, Ethereum, Smart Contracts, IoT, Sensory, Insurance, Access Control**

I. Introduction

The traditional insurance ecosystem relies on complex contracts between the insurer and the customer as well as strict decision processes. This complexity is currently required as insurer and customer have inherently opposing interests and can also act maliciously: On the one hand, the insurer seeks to minimize required payouts to her customers. This enables the insurer to reduce insurance premiums for all customers as well as to retain a higher profit. A dishonest insurer could, for instance, try to block rightful payouts by exploiting loopholes in the insurance contract. On the other hand, the customer gets insurance because of the promise to be reimbursed in case of unforeseen damages such as car accidents. By committing insurance fraud, a malicious customer could also try to trick the insurer into paying unjustified reimbursements [1]. Hence, insurance policies must account for a plethora of possible eventualities, and thus become hard to understand for the customer as well as hard to validate.

In case of a claim by the customer, the insurer currently has to manually validate whether the preconditions of the insurance policy are met by the claim. This process quickly becomes tedious and expensive, as it requires consulting surveyors and intensive paperwork, and makes up around $25\,\%$ of the insurer's costs [2]. While this involved process is necessary for certain cases such as attempted insurance fraud, there is a high potential for improvement in clear cases or cases of only minor value that do not warrant costly investigation.

It is thus desirable for insurers to further automate the processing of their customers' claims in order to reduce costs. However, since both the insurer and the customer may behave maliciously, an independent third party must still be involved in the process. Since this is seemingly in contrast to the automation desires of insurers, in this paper, we investigate how to provide a further automated, trustless, and independent oversight system for car insurances to reduce overall costs.

In this paper, we thus present CAIPY, our Ethereum-based [3] **ca**r **i**nsurance **p**olic**y** framework relying on tamper-resistant sensors. Public blockchain systems such as Ethereum provide an immutable ledger that enables transparent processes between mutually distrusting parties. Further, their attached digital currencies are becoming widely accepted and can be used to automatically reimburse customers at low overheads.

Our design takes advantage of these properties and consists of smart contracts that realize a trustless ledger of car insurance-related events, e.g., crashes or other component malfunctions, as well as the current status of any customer claim. CAIPY benefits from automated damage detection based on tamper-resistant sensors to reduce processing costs of insurance claims. Nevertheless, if in doubt both parties can request a manual inspection of the claim, e.g., involve an independent surveyor, at increased costs. This enables insurers to carefully gauge how much to rely on smart contract-based automation of their processes. Hence, by combining blockchain-based automation with the option to have an independent third party manually investigate insurance claims further CAIPY can decrease insurance costs without additional required trust relations between insurer and customer.

The remainder of this paper is organized as follows. In Section II, we define the scenario and outline challenges and building blocks for smart contract-based car insurance. Section III then describes our framework, CAIPY, which we evaluate w.r.t. costs, security, and reliability in Section IV. Section V discusses limitations and future work. Section VI discusses related work and Section VII concludes this paper.

II. Scenario and Challenges

To motivate the benefits of smart contract-based car insurance, we first derive our assumed scenario from analyzing current approaches to car insurance (Section II-A). We then identify the emerging technology of tamper-resistant sensors as a valuable building block to further automate car insurance processes for potential cost savings (Section II-B). Finally, we argue that the application of blockchain technology, especially

* Equal Contribution

 DOI: 10.1109/GLOCOMW.2018.8644136

smart contracts, is key to realize this potential (Section II-C) and outline challenges for smart contract-based car insurance (Section II-D).

A. The Current State of Car Insurance

Processes in the car insurance industry are currently complex and cost-intensive [2], and they typically involve the *insurer*, the *customer*, external *surveyors*, and a *judge*.

To insure her car, the customer negotiates an *insurance policy* with the insurer. The insurance policy is a legally binding contract that determines conditions under which the insurer reimburses the customer for damages the customer cannot be held accountable for, e.g., certain crashes or malfunction of a component. To finance such reimbursements, the insurer anticipates that costly damages are seldom and collects a regular fee from all customers that is small in relation.

Customers have to apply for reimbursements. To avoid insurance fraud, the insurer can choose to task a surveyor with inspecting the alleged damage. The surveyor creates a report of the inspection, which is used by the insurer to check whether the customer should be reimbursed according to the insurance policy. In case that the report is not decisive, for instance, because a customer is accountable for the damage but tries to cover this fact up, the insurer can also include further external sources such as police reports into the decision.

Since insurance policies are often complex, the insurer's decision can be intransparent to the customer, who can contest that decision. In this case, both parties can negotiate a settlement or ask the judge for a definitive decision.

B. Sources for Reliable Data in Process Automation

The availability of reliable data is crucial for the insurer to decide whether or not to reimburse a customer. Thus, insurers currently task external surveyors to inspect customer claims to ensure data reliability. Any alternatives must ensure the same level of data reliability via technical means.

Unfortunately, traditional sensors and electronic control units (ECUs), as deployed in most current cars to enable driver-assistance or safety systems, are often not designed for transferring data between an insurance customer and the respective insurer in a non-manipulable manner [4]. However, due to the increasing importance of vehicle telematics, *tamper-resistant* sensors and ECUs recently emerged, which make manipulation attempts by the customer either nearly impossible or at least immediately detectable.

One approach to implement tamper-resistant devices is motivated by privacy needs and uses special cryptographic modules that delete cryptographic secrets upon detecting manipulation. The NIST specifies requirements for such modules in four different levels in FIPS 140-2 [5], where any manipulation to a Level 4 cryptographic module needs to be detected with very high probability. Notably, first tamper-resistant devices for increased data privacy are already available [6], albeit currently arguably too expensive to be mass-produced for cars. Although we expect these costs to be reduced once the market for tamper-resistant sensors and ECUs grows, insurers can already trade off costs against the security needs w.r.t. reported sensor readings. When relying on only detecting physical manipulation of sensors or ECUs, e.g., due to broken seals, the insurer still requires to conduct manual inspection to identify such manipulations. However, verifying that such a seal is still intact is easier than assessing the customer's insurance claim and can thus be performed at lower costs, potentially even without a third-party surveyor (e.g., via online photo proof).

In conclusion, tamper-resistant sensors and ECUs are an emerging technology with a high potential for future utilization in automated and reliable communication between mutually distrusting parties. Since cheaper variants with sufficient potential cost savings, e.g., sealed sensors, are already widely deployable, we assume their availability for the remainder of this paper.

C. Benefits of Blockchain-based Car Insurance

We have argued that processes related to car insurance can be further automated and that tamper-resistant sensors provide a valuable building block in achieving this goal. In this section, we argue that blockchain technology, especially smart contracts, is key to seizing this potential.

As discussed in Section II-A, current processes for car insurance often rely on a manual inspection conducted by an independent surveyor. Although this dependency is the main obstacle for further automation attempts, the surveyor plays a crucial role as a trusted third party for both the customer and the insurer. Hence, overcoming the dependency on external surveyors is a promising approach for further automation, but its automated replacement must be equally trustworthy.

Blockchain technology promises to constitute exactly this trustworthy replacement by providing a decentralized and immutable event ledger. While initially only recording financial data, blockchains are now also being used to record non-financial data [7]. Subsequently, Ethereum [3] smart contracts further extended the functionality of cryptocurrencies by allowing for the enforcement of digital payments if freely definable conditions are satisfied. Smart contracts thus provide the foundation for defining conditions for the automated reimbursement of customers of car insurances based on reliable data, e.g., originating from tamper-resistant sensors. The utilization of the Ethereum blockchain thus promises transparency of insurance processes as well as an additional layer of resistance to data manipulation by both the insurer and the customer. However, this promising approach comes along with new challenges, as we detail in the next section.

D. Challenges for Smart Contract-based Car Insurance

Integrating smart contracts into current insurance processes bears the potential to reduce costs by simplifying and automating them, but comes with the following challenges.

Data Reliability. As argued in Section II-B, it is the main challenge for further process automation in car insurance that insurers have access to reliable event data. Any smart contract-based optimization requires that the smart contract is provided with equally reliable data for its decisions. While

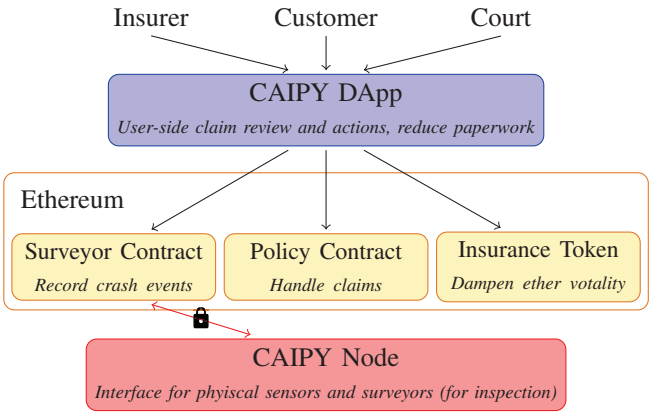

Fig. 1. Overview of CAIPY's overall design

tamper-resistant sensors are promising to emit reliable data, we still must assume that sensor readings are occasionally erroneous. Hence, insurers need to be able to intervene in case of suspicious sensor readings. Finally, the sensors and smart contract must be aware that sensor readings might be delayed or even lost due to temporarily bad connectivity.

Cost Efficiency. Automation of insurance processes via smart contracts is no guarantee for cost reductions. The extremely volatile [8], but generally comparably high prices of popular cryptocurrencies such as Bitcoin or Ethereum render it increasingly challenging to design cost-efficient blockchain-based processes. This is mainly due to transaction fees, which are paid per byte of transaction size, as well as gas costs. In order to communicate with smart contracts, Ethereum users have to pay its operation with so-called gas, a subdivision of Ether, which is directly proportional to the complexity of the interaction with the smart contract. To reduce costs of current insurance processes, a smart contract-based alternative must be aware of these additional and non-negligible cost factors.

Customer Privacy. Blockchain data is inherently public to all participants, i.e., sensitive customer information such as event locations could be leaked if sensor-recorded events were stored in the clear. Storing only encrypted data on the blockchain instead requires access control such that only authorized parties can decrypt and further process event data.

III. CAIPY Design

We now present CAIPY, our smart contract-based **ca**r **i**nsurance **p**olic**y**. We give a high-level overview in Section III-A and then discuss the involved smart contracts in Section III-B as well as interaction with CAIPY in Section III-C. Finally, we discuss customer privacy in Section III-D.

A. Design Overview

The goal of CAIPY is *not* to replace classical processes of car insurers entirely with smart contracts, but it rather acknowledges that the insurer must remain in power to overrule the smart contracts' decisions. Hence, it is a central design element of CAIPY that the smart contracts reliably record insurance processes and can make suggestions on behalf of the insurer, but that the insurer can choose to nevertheless involve independent third parties into the process, albeit at higher costs. This element of CAIPY is crucial, for instance in the case of unforeseen sensor manipulation, i.e., jeopardized data reliability. We thus rather *extend* current insurance processes with smart contracts whenever we can either simplify the process, reduce its costs, or increase transparency for customers.

Figure 1 provides a high-level overview of CAIPY. CAIPY is based on Ethereum and uses standard smart contracts to create an immutable ledger of insurance-related events such as crashes or component malfunctions as well as the status of customer claims. We chose to base CAIPY on a public blockchain system instead of a permissioned system such as Hyperledger Fabric [9] because of two reasons. First, Ethereum's cryptocurrency, Ether, is widely accepted outside the context of insurance and thus allows directly reimbursing customers. Secondly and more importantly, insurers have an incentive to act malicious-but-cautiously [10] and hence a smart contract-based oversight of insurance processes must not be controlled exclusively by different insurers.

At the core of CAIPY's design is the *policy contract*, a smart contract that models relevant parts of the process defined by the physical insurance policy between insurer and customer and thus can mediate between both parties. Most notably, the smart contract moderates currently available options to both parties in case of a dispute over an insurance claim.

A significant cost factor for the insurer is the consultation of external surveyors to verify customer claims. To reduce this overhead, CAIPY assumes that cars contain a comprehensive set of tamper-resistant sensors (cf. Section II-B), which can reliably detect insurance-relevant events such as crashes or malfunctioning components. In CAIPY, these sensors communicate with a *surveyor contract*, which is responsible for storing relevant event data persistently on the blockchain.

CAIPY channels all user-based interaction with smart contracts through a *DApp*, a browser-based frontend for Ethereum smart contracts. Finally, CAIPY uses a dedicated *insurance token* to mitigate effects of high market price volatility in Ethereum. In the remainder of this section, we further discuss the individual components of CAIPY.

B. Smart Contract-based Event Ledger

As shown in Figure 1, smart contracts constitute the backend of CAIPY. CAIPY uses three different types of Ethereum smart contracts: the *policy contract*, the *surveyor contract*, and a smart contract implementing the *insurance token*.

Policy Contract. The policy contract moderates the steps of processing a customer claim between the customer and insurer. Figure 2 shows a simplified version of the policy contract's underlying state transition model. Once the surveyor contract reports a crash because of readings of the tamper-resistant sensors, the policy contract automatically opens up a customer claim. The customer can either decide to withdraw the claim (e.g., to prevent a rise in premiums) or pursue it. In the latter case, the customer requests a reimbursement, which the insurer must either approve or reject. If the insurer

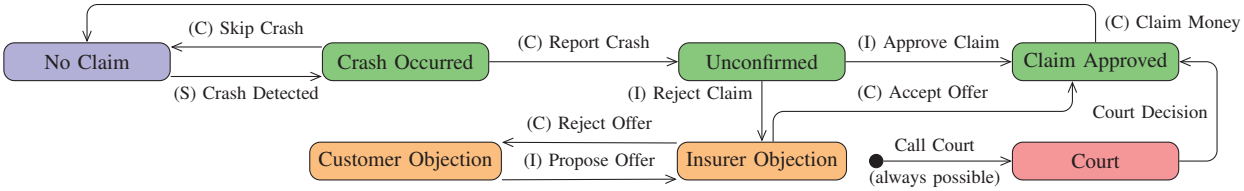

Fig. 2. Simplified state transition model underlying CAIPY. In the best case, insurer and customer agree on a decision about an insurance claim. Alternatively, they can either settle the case or involve a judge.

approves the claim, the customer is reimbursed, and the claim processing concludes. Otherwise, both parties can negotiate a more appropriate reimbursement. As a last resort, both parties can call a court at each stage to enforce a decision using the well-established, but costly, traditional insurance processes.

Surveyor Contract. The surveyor contract serves as a brand-specific proxy between cars and the policy contract of the insurer. Since car sensors only report events to the surveyor contract independent of the contract's state, the car is not required to store the Ethereum blockchain. Instead, it only needs to be capable of creating Ethereum transactions to be sent to the surveyor contract. The surveyor contract records insurance-relevant events reported by a car on behalf of the customer on the blockchain and informs the policy contract of any new events. Since we assume that tamper-resistant sensors report the events, the surveyor contract can reliably report relevant events. Thereby, the currently often mandatory manual claim inspections by surveyors can be avoided. However, the insurer can further minimize its risk of reimbursing false claims by opting to supplement this first assessment with an additional manual inspection if deemed necessary. As we further detail in Section III-D, the surveyor contract also maintains the confidentiality of recorded events until an authorized party requests to disclose that data.

Insurance Token. As we discussed in Section II-D, the market prices of cryptocurrencies tend to be highly volatile [8]. To *optionally* mitigate these effects, the insurance token constitutes a sub-currency to be used for reimbursements by the insurer and subsequently accepted by further business partners, e.g., car repair services. By creating isolated insurance token via an additional smart contract, we can decouple the transferred values within the car insurance ecosystem from volatile cryptocurrency prices, which mitigates financial risks for both the insurer and the customer. Insurers could also tie the insurance tokens' value to real-world currencies in order to make them more easily spendable for their customers.

In the subsequent section, we describe how the parties involved in car insurance processes can conveniently interact with these smart contract in order to make CAIPY usable for a broad audience of insurers and customers.

C. A User-Friendly Interface for Insurance Processes

CAIPY intends two different forms of interactions with its backend, i.e., the smart contracts we introduced in Section III-B: While car sensors communicate with the surveyor contract directly, user interaction with any of the smart contracts is channeled through the CAIPY DApp[1].

As discussed in Section II-B, CAIPY relies on the availability of tamper-resistant sensors within insured cars. These sensors monitor the car and send relevant information directly to the surveyor contract in case of an event. We realized a simple sensor with key management on a Raspberry Pi using Python. While we implemented the necessary cryptographic aspects of a sensor node, we did not focus on the actual tamper-resistance of the hardware setup. The sensor can cache recorded events in case of bad connectivity to send them to the surveyor contract at a later point. Further, it encrypts all event data prior to sending it to the surveyor contract and provides integrity protection using the keccak-256 hash function [11], as this enables automatic on-chain verification (cf. Section III-D). The sensor offloads all blockchain operations to a trusted Ethereum node via its RPC provider, e.g., an Ethereum node the customer runs at home to potentially receive reimbursements. This unburdens the sensor manufacturer from dealing with blockchain specifics as much as possible while still providing a trustworthy environment for the customer.

For all human interaction of the customer, insurer, and court with the smart contract, we implemented the CAIPY DApp using the web3.js API, the cryptojs library and several other Node.js libraries. The CAIPY DApp provides different views that are tailored towards the different roles users can assume. Most notably, customers and insurers get an overview of recorded events and customer claims as well as the current state of any open claims. Furthermore, they can instruct the policy contract based on the current state according to our simple process described in Figure 2.

D. Privacy-preserving Data Access

In order to provide a transparent car insurance ecosystem, CAIPY requires to store information about insured cars and insurance-related events, e.g., detected crashes or malfunction of a car component, on the blockchain. In CAIPY, car sensor nodes only upload punctual information on such events instead of all available data to protect the customer's privacy against the insurer to the best extent possible. While the remaining event data is crucial for processing insurance claims, CAIPY must protect the customer's privacy against outsiders monitoring the public Ethereum blockchain. We thus only store AES-encrypted information on the blockchain. The corresponding keys are distributed to the authorized parties, i.e., the customer, the insurer, and the court, by encrypting them asymmetrically

[1]Demo available at http://caipy.comsys.rwth-aachen.de

using ECIES for each involved party and storing the encrypted AES keys on the blockchain as well. This approach ensures the availability of all data on insurance claims to each party. Furthermore, this way CAIPY can be extended to allow for more sophisticated access control in the future, e.g., provide anyone interested in purchasing a car with a trustworthy history of the car to reduce the threat of scams.

IV. Evaluation

We evaluate and discuss CAIPY w.r.t. cost (Section IV-A), security (Section IV-B), and timing constraints (Section IV-C).

A. Costs of Smart Contract Interaction

In this section, we analyze the costs that arise from interacting with the smart contracts constituting the backend of CAIPY. Costs can stem from transaction fees as well as the gas costs required to execute the smart contract functionality. We only consider operations that alter the state of the smart contracts, as reading operations can be performed locally and are thus considered to be free.

Table I shows the proposed as well as actual gas costs of the most common operations in CAIPY and their real-world prices in EUR as well as USD. For our analysis, we consider the average gas price of 17 GWei (1 GWei $= 1 \times 10^9$ Wei) during May 2018 [12] and the Ether market price of approximately 580 EUR and 683 USD as of May 22nd, 2018 [8].

According to our analysis the most expensive operation in our system is `addDecryptKey`, costing almost 7 EUR (8.24 USD). This operation is used to store the ECIES-encrypted AES keys (cf. Section III-D) on the blockchain. Notably, this operation is only performed when setting up the insurance policy or on change of authorized parties, e.g., if the customer sells her car. Hence, these comparably high costs amortize over the validity period of the insurance policy.

All other operations are performed for each insurance-relevant event either when the sensors report to the surveyor contract or the customer or insurer make decisions about the open claim. Here, the most expensive operation is `addEvent`, which costs around 2 EUR (2.35 USD). The costs of `addEvent` are strongly influenced by the amount of data that must be uploaded in case of a relevant event. In our analysis, we assumed small payloads of up to 50 bytes. We thus propose that sensors aggregate data before reporting an event so that the data on the blockchain still provides evidence of the event, but the amount of data is nevertheless minimized.

We conclude that incorporating smart contracts into car insurance processes comes at negligible costs for the insurer.

TABLE I
Gas Costs of the Most Common Smart Contract Operations

Caller	Operation (Short name)	Gas Costs (Wei)	Proposed Gas (Wei)	Real Costs (EUR)	(USD)
Sen. Node	`addEvent`	208 840	-	2.06	2.43
Sen. Node	`addDecrKey`	690 506	-	6.81	8.02
Customer	`reportCrash`	129 568	168 439	1.28	1.50
Customer	`accept`	34 446	44 780	0.33	0.40
Customer	`claimMoney`	107 230	139 399	1.06	1.25
Insurer	`approve`	72 528	94 287	0.71	0.84
Insurer	`disapprove`	46 094	59 923	0.45	0.54
Cust./Ins.	`callCourt`	34 632	47 253	0.36	0.40

B. Security Discussion

The substantial monetary values that are transferred in insurance ecosystems necessitate that CAIPY is secure. We now discuss (i) how CAIPY prevents data manipulation and privacy breaches and (ii) the security of its smart contracts.

Data Manipulation. Since CAIPY relies on tamper-resistant sensors, we can safely assume that only unaltered data ends up on the blockchain. Subsequently, neither party can delete nor manipulate the data anymore. In case that one party lies about the plain data, the other party can decrypt the blockchain data using her own copy of the used key and can prove that the key is correct via her ECIES identity.

Privacy Breaches. No event data is recorded in the clear on the blockchain. Hence, the customer data is well-protected from outsider access. Misbehavior by the insurer, e.g., disclosing the data to third parties is a general threat and, for instance, the GDPR allows taking such cases to court. We protect the integrity of the plain data by storing a checksum over the data and a random salt in addition to the encrypted data. By applying the random salt, we prevent that information about the plain data can be derived from the keccak-256 checksum.

Smart Contract Security. Previous incidents have proven the high risks stemming from erroneous smart contracts [13]. Most common vulnerabilities can be avoided by using formal testing methods [14]. However, CAIPY additionally protects insurers against errors by not relying entirely on smart contracts. At any point, the parties can complement an open claim via manual inspection (at higher costs).

C. Timing and Loss of Event Records

The semi-automation of insurance processes enabled by CAIPY has the potential for significant time reductions when recording and settling insurance claims.

We expect cases to be settled in the order of hours in case of undisputed claims. However, public blockchains can suffer from congestion [15], which can cause delays when processing insurance events. While Ethereum currently has a comparably stable number of pending transactions [16], congestion can increase block and transaction propagation times [17].

Another threat is real data loss, e.g., when the sensor node is not aware that its event-recording transactions might be delayed indefinitely. In case of lost transactions during the processing of an open claim via CAIPY's DApp, involved parties can recognize unexpected behavior and can react accordingly. For instance, they can defer the decision to the court at all time. We thus propose that CAIPY-enabled sensors also verify that transactions are recorded via their RPC provider and cache and log events to counter short-term outages.

V. Limitations and Future Work

CAIPY showcases how car insurance policies can be partially managed using smart contracts in order to reduce overall

costs of current processes. However, we identify the following current limitations of our approach that motivate future work.

Smart Contract Functionality. In our design, CAIPY can mediate decisions between insurer and customer without the need for an external surveyor inspecting a customer's insurance claim. Shifting even more functionality to the involved smart contracts, for instance, direct reimbursement decisions without the insurer in the loop is promising to increase transparency for the user and to simplify insurance processes even further. However, the insurer must carefully gauge which decisions *can* and *should* be offloaded to smart contracts. As the infamous DAO incident of Ethereum [13] has shown, mistakes can in turn become extremely costly for the insurer. Another problem is the question whether a customer is accountable for the reported damage. We propose to investigate means to incorporate other trusted external information such as police reports into smart contract decisions, e.g., by also outsourcing those to the blockchain in a secure manner.

Data Privacy vs. Automation. The requirement for data privacy especially limits what functionality can be shifted to smart contracts, as all data visible to the smart contract must be assumed to be public. A potential remedy to this is to use (fully) homomorphic encryption schemes in the future instead of our current approach of using symmetric encryption keys that are shared within asymmetrically encrypted envelopes. However, homomorphic encryption is no standard feature offered by smart contracts and manual implementation of such schemes likely results in expensive-to-execute smart contracts. In fact, a more fine-granular management of insurance data, e.g., combining both approaches based on concrete use cases, opens up a new design space for future work.

Data Correctness. As briefly discussed in Section II-D, smart contract-based decisions heavily rely on data correctness. CAIPY thus requires and motivates further research into tamper-resistant sensors that are feasible to be deployed in cars beyond the recent advances we discussed in Section II-B. While the requirements of CAIPY for such sensors to facilitate simple and clear decisions are comparably low, the availability of such sensors would also benefit other areas of interest such as supply chain management or crowd sensing.

Scalability. We anticipate scalability limitations for CAIPY with respect to Ethereum's blockchain capacities. According to Etherscan, the highest daily transaction volume Ethereum has experienced to this day were about 1.35 million transactions on January 4th, 2018 [18], which results in a maximum experienced throughput of 15.6 transactions per second. In comparison, the police reported a number of 2.6 million car accidents only in Germany during 2017 [19], i.e., about five accidents per minute. Hence, assuming wide-spread adoption in Germany alone CAIPY could become responsible for well over $2\,\%$ of Ethereum's maximum throughput even if all insurance claims are handled with minimal overhead. Furthermore, CAIPY currently does not consider the deletion of past events, which could become a burden once CAIPY would be extensively used for a longer time. A potential relief for this scenario could be to build CAIPY on top of a special-purpose permissioned blockchain instead of the general-purpose Ethereum blockchain. While this allows tailoring such parameters to the special needs of CAIPY, it requires a careful distribution of permissioned blockchain nodes among the involved parties to avoid advantaging one of them. Finally, using general-purpose cryptocurrencies facilitates smart contract-based reimbursements as its payments have immediate value for the customer.

VI. Related Work

Previous approaches to blockchain-based insurance mainly focus on the automation capabilities of the blockchain to cut costs as well as accelerating the processing of claims, enabling new payment forms, and improving the overall customer experience instead of privacy [20]–[22]. A platform to fully transfer mainstream insurance business to the blockchain is currently being created by the Blockchain Insurance Industry Initiative (B3i) [23], which is backed by multiple large insurers. Instead of using a public and existing blockchain such as Ethereum, this platform is based on the Hyperledger Fabric framework [9]. This necessitates to create a trustless consortium of blockchain nodes in order to avoid cartel issues. Relying on a public blockchain instead solves this issue and integrates well with already-existing blockchain ecosystems. Finally, blockchain-based vehicular forensics [24], [25] is in part orthogonal to this work, which focuses on simplified processing of insurance claims, but can further improve the decision-making of insurers using CAIPY.

Ensuring privacy on public blockchains while preserving the automation capabilities and the transparency of smart contracts is still technically challenging [26]. One stream of research [27]–[29] utilizes homomorphic encryption [30], [31] in order to implement privacy-preserving blockchain applications. Another approach to ensure privacy in smart contracts is to not only encrypt the processed data, but the whole smart contract and the corresponding transactions [32]. However, this approach does not allow a public validation of the contract, since only blockchain users with the decryption key can execute the contract and read the transaction payload [32].

Further, also the highly volatile exchange rates can quickly become an issue when established businesses want to use the blockchain. In contrast to most popular cryptocurrencies, so-called stablecoins such as Tether [33] measure their own value and apply countermeasures in case exchange rate fluctuations occur [34], [35]. As a result, stablecoins could be used as an appropriate alternative to classic currencies [35].

VII. Conclusion

We presented CAIPY, an Ethereum-based framework for the cost-efficient and privacy-preserving management of car insurance policies. Traditional processes are complex and cumbersome for both insurers and customers: Manual inspection of insurance claims is expensive, time-consuming, and prone to intransparent decisions and attempted insurance fraud.

CAIPY remedies this situation by complementing the traditional inspection of insurance claims with a semi-automated

and trustless approach. With CAIPY, we showcase that car insurers can simplify their processes, and thus reduce their costs, for common cases of insurance claims by outsourcing basic operations to smart contracts without disclosing confidential information to third parties. To enable this shift, CAIPY relies on and motivates a wide deployment of trusted sensors in the car industry, which allows a smart contract associated with the car to reliably recognize events that are relevant for potential claims, e.g., damages or malfunction. Another smart contract, which represents the insurance policy, subsequently manages the processing of such claims and is orchestrated via the CAIPY DApp, a simple-to-use browser-based frontend. This enables both parties to resolve insurance claims without extensive paperwork or having to consult external surveyors. However, CAIPY acknowledges that insurance involves complex decisions, which likely cannot be modeled appropriately by a smart contract. Hence, our design deliberately enables insurers and customers to opt for the consultation of external entities as in today's insurance ecosystem.

Our prototypic implementation of CAIPY shows that simple decisions such as detection of a car crash could be settled at costs of under five dollars, which showcases the potential for blockchain-based insurance ecosystems.

Acknowledgements

This work has been funded by the German Federal Ministry of Education and Research (BMBF) under funding reference number 16KIS0443. The responsibility for the content of this publication lies with the authors, who would also like to thank the German Research Foundation DFG for the kind support within the Cluster of Excellence "Integrative Production Technology for High-Wage Countries".

References

[1] Association of British Insurers, "The con's not on – Insurers thwart 2,400 fraudulent insurance claims valued at £25 million every week," 2018, last accessed: 09/14/2018. [Online]. Available: https://www.abi.org.uk/news/news-articles/2017/07/the-cons-not-on--insurers-thwart-2400-fraudulent-insurance-claims-valued-at-25-million-every-week

[2] McKinsey&Company, "Successfully reducing insurance operating costs."

[3] G. Wood, "Ethereum: A secure decentralised generalised transaction ledger," *Ethereum project yellow paper*, vol. 151, pp. 1–32, 2014.

[4] F. Sagstetter, M. Lukasiewycz, S. Steinhorst, M. Wolf, A. Bouard, W. R. Harris, S. Jha, T. Peyrin, A. Poschmann, and S. Chakraborty, "Security challenges in automotive hardware/software architecture design," in *Proc. of DATE 2013*. EDA Consortium, 2013, pp. 458–463.

[5] "Security requirements for cryptographic modules," Tech. Rep., may 2001, last accessed: 09/14/2018. [Online]. Available: https://doi.org/10.6028/nist.fips.140-2

[6] ORWL, "ORWL — The worlds first physically secure open source computer," last accessed: 09/14/2018. [Online]. Available: https://orwl.org

[7] R. Matzutt, J. Hiller, M. Henze, J. H. Ziegeldorf, D. Müllmann, O. Hohlfeld, and K. Wehrle, "A Quantitative Analysis of the Impact of Arbitrary Blockchain Content on Bitcoin," in *Proc. of FC 2018*. Springer, 2018.

[8] ethereumprice, "Ethereum Average Price," last accessed: 09/14/2018. [Online]. Available: https://ethereumprice.org/eth-eur

[9] Hyperledger, "Hyperledger Fabric Framework," last accessed: 09/14/2018. [Online]. Available: https://www.hyperledger.org/projects/fabric

[10] M. D. Ryan, "Enhanced Certificate Transparency And End-to-End encrypted Mail," in *Proc. of NDSS 2014*. The Internet Society, 2014.

[11] G. Bertoni, J. Daemen, M. Peeters, and G. Assche, "The keccak SHA-3 submission. Submission to NIST (Round 3)(2011)."

[12] Etherscan, "Gas Price," last accessed: 09/14/2018. [Online]. Available: https://etherscan.io/chart/gasprice

[13] M. del Castillo, "The DAO Attacked: Code Issue Leads to $60 Million Ether Theft," 2016, last accessed: 09/14/2018. [Online]. Available: https://www.coindesk.com/dao-attacked-code-issue-leads-60-million-ether-theft

[14] K. Bhargavan *et al.*, "Formal verification of smart contracts: Short paper," in *Proc. of the 2016 ACM Workshop on Programming Languages and Analysis for Security*, 2016, pp. 91–96.

[15] Blockchain Luxembourg S.A., "Bitcoin Mempool Transaction Count," last accessed: 07/14/2018. [Online]. Available: https://blockchain.info/charts/mempool-count?timespan=1year

[16] Etherscan, "Pending Transactions," last accessed: 09/14/2018. [Online]. Available: https://etherscan.io/chart/pendingtx

[17] ——, "Ethereum Block Times," last accessed: 09/14/2018. [Online]. Available: https://etherscan.io/chart/blocktime

[18] ——, "Transactions Per Day," last accessed: 09/14/2018. [Online]. Available: https://etherscan.io/chart/tx

[19] Statistisches Bundesamt, "Road traffic accidents - Accidents registered by the police," last accessed: 09/14/2018. [Online]. Available: https://www.destatis.de/EN/FactsFigures/EconomicSectors/TransportTraffic/TrafficAccidents/Tables/AccidentsRegisteredPolice.html

[20] McKinsey & Company, "Blockchain in insurance– opportunity or threat?" last accessed: 07/14/2018. [Online]. Available: https://www.mckinsey.com/industries/financial-services/our-insights/blockchain-in-insurance-opportunity-or-threat

[21] EY, "Blockchain in insurance: applications and pursuing a path to adoption," last accessed: 07/14/2018. [Online]. Available: https://webforms.ey.com/Publication/vwLUAssets/EY-blockchain-in-insurance/$FILE/EY-blockchain-in-insurance.pdf

[22] IBM, "Three areas in the insurance industry to use blockchain," last accessed: 09/14/2018. [Online]. Available: https://www.ibm.com/blogs/blockchain/2018/03/three-areas-in-the-insurance-industry-to-use-blockchain

[23] B3i, "B3i: Our product," last accessed: 09/14/2018. [Online]. Available: https://b3i.tech/our-product.html

[24] M. Cebe *et al.*, "Block4Forensic: An Integrated Lightweight Blockchain Framework for Forensics Applications of Connected Vehicles," *CoRR*, 2018, last accessed: 09/14/2018. [Online]. Available: http://arxiv.org/abs/1802.00561

[25] C. Oham, S. Kanhere, R. Jurdak, and S. Jha, "A Blockchain Based Liability Attribution Framework for Autonomous Vehicles," *CoRR*, 2018, last accessed: 09/14/2018. [Online]. Available: http://arxiv.org/abs/1802.05050

[26] V. Buterin, "Privacy on the Blockchain," last accessed: 09/14/2018. [Online]. Available: https://blog.ethereum.org/2016/01/15/privacy-on-the-blockchain

[27] C. Gentry, "Fully homomorphic encryption using ideal lattices," in *Proc. of STOC 2009*, 2009, pp. 169–178.

[28] C. Gentry and S. Halevi, "Implementing Gentry's Fully-Homomorphic Encryption Scheme," in *Proc. of Eurocrypt 2011*, 2011, pp. 129–148.

[29] P. Scholl and N. P. Smart, "Improved key generation for Gentry's fully homomorphic encryption scheme," in *IMA International Conference on Cryptography and Coding*, 2011, pp. 10–22.

[30] R. L. Rivest, A. Shamir, and L. Adleman, "A method for obtaining digital signatures and public-key cryptosystems," *Communications of the ACM*, vol. 21, no. 2, pp. 120–126, 1978.

[31] R. L. Rivest, L. Adleman, and M. L. Dertouzos, "On data banks and privacy homomorphisms," *Foundations of secure computation*, vol. 4, no. 11, pp. 169–180, 1978.

[32] Parity.IO, "Private Transactions," last accessed: 09/14/2018. [Online]. Available: https://wiki.parity.io/Private-Transactions.html

[33] "Tether: Fiat currencies on the Bitcoin blockchain," last accessed: 09/14/2018. [Online]. Available: https://tether.to/wp-content/uploads/2016/06/TetherWhitePaper.pdf

[34] V. Buterin, "The Search for a Stable Cryptocurrency," last accessed: 09/14/2018. [Online]. Available: https://blog.ethereum.org/2014/11/11/search-stable-cryptocurrency

[35] A. T., "The Rise Of Stablecoins," last accessed: 09/14/2018. [Online]. Available: https://coinjournal.net/the-rise-of-stablecoins/2