https://doi.org/10.1007/s10207-020-00528-1

REGULAR CONTRIBUTION

DEALER: decentralized incentives for threat intelligence reporting and exchange

Florian Menges[1] · Benedikt Putz [1] · Günther Pernul[1]

Published online: 9 December 2020

Abstract

The exchange of threat intelligence information can make a significant contribution to improving IT security in companies and has become increasingly important in recent years. However, such an exchange also entails costs and risks, preventing many companies from participating. In addition, since legal reporting requirements were introduced in various countries, certain requirements must be taken into account in the exchange process. However, existing exchange platforms neither offer incentives to participate in the exchange process, nor fulfill requirements resulting from reporting obligations. With this work, we present a decentralized platform for the exchange of threat intelligence information. The platform supports the fulfillment of legal reporting obligations for security incidents and provides additional incentives for information exchange between the parties involved. We evaluate the platform by implementing it based on the EOS blockchain and IPFS distributed hash table. The prototype and cost measurements demonstrate the feasibility and cost-efficiency of our concept.

Keywords Threat intelligence sharing · Blockchain · Smart contract

1 Introduction

The threat landscape for IT infrastructures has grown steadily in recent years, and this trend is continuing. At the same time, it is becoming apparent that the countermeasures currently available can hardly keep pace with the ongoing attacks. It has been shown that the exchange of threat information is an effective instrument for improving existing countermeasures and the overall situation. It leads to more knowledge about threats, earlier detection of attacks and thus to more effective countermeasures. The potential benefits of the threat information exchange have recently been recognized in the public sector by introducing corresponding legal regulations. For example, several countries already require the reporting of security incidents, especially for critical infrastructure operators.

While the exchange of threat information offers the aforementioned benefits for the security situation, it can also entail various disadvantages and problems that may prevent companies from participating. These include high additional costs for appropriately trained security personnel and infrastructure, possible data protection problems and the risk of publishing sensitive data. In addition to these problems, a complex set of reporting requirements must be taken into account. Companies must be able to provide non-repudiable proof of accurate reporting, both to avoid penalties and to potentially use the data as evidence in court. Consequently, sustained availability and integrity of the reported data must be ensured. Sharing platforms must address these problems by incorporating legal requirements as part of the design. Additionally, incentive structures must be created for the exchange of threat information, to offset costs and to motivate stakeholders to participate in the long term.

In doing so, we consider two use-cases separately. The platform intends to (1) support the fulfillment of legally **obligatory reporting** and (2) to create economic incentives for **voluntary reporting**. While these scenarios have different requirements and thus follow separate processes, the proposed platform optionally also enables sharing of obligatory reports. Based on these considerations, we formulate the research questions we intend to answer:

✉ Florian Menges
Florian.Menges@ur.de

Benedikt Putz
Benedikt.Putz@ur.de

Günther Pernul
Gunther.Pernul@ur.de

[1] University of Regensburg, Universitätsstr. 31, 93053 Regensburg, Germany

- **RQ1**: How can threat intelligence information be exchanged while ensuring availability, integrity and non-repudiation?
- **RQ2**: How can the exchange of threat intelligence information be incentivized?

To solve these problems, we propose a sharing concept and application prototype for a threat intelligence sharing platform based on Distributed Ledger Technology (DLT). DLT provides specific security characteristics, which can differ depending on the blockchain implementation. These usually include *availability*, *integrity* and *non-repudiation* - the three requirements of RQ1 [1]. *Availability* is ensured by the underlying blockchain network, which consists of a large number of geo-distributed nodes maintaining a replicated ledger around the clock. Please note that Proof of Work (PoW)-based blockchains may suffer availability limitations under heavy load [2]. At the same time, *integrity* assurance is provided through a sequentially linked hash chain, which ensures that the current world state is always the result of all past transactions. The consensus protocol assures that state transitions are append-only and previous entries are *non-repudiable*. Distributed Ledgers enable the verifiable decentralized execution of applications in the form of smart contracts, which also provide the option to implement digital currency in the form of blockchain tokens. These tokens can be used to provide decentralized *incentives* by assigning real value to threat intelligence information.

Existing work has attempted to address some of the aforementioned problems using DLT; however, the research questions have not been sufficiently addressed so far (Sects. 2 and 3.3). For this reason, we propose the blockchain-based DEALER platform (**D**ecentralized Inc**E**ntives for Thre**A**t Inte**L**ligE**n**ce **R**eporting and Exchange). It fulfills requirements for obligatory Cyber Threat Intelligence (CTI) reporting (Sects. 3.1 and 7.1), while also providing an incentive structure to counteract possible participation drawbacks and to encourage voluntary sharing of CTI. Our contribution includes a novel protocol based on verifiers and token-based incentives to encourage fair sharing of high-quality threat intelligence data. To avoid trusting a third-party platform provider, the architecture is fully decentralized and maintained by independent blockchain operators and the participants themselves. In brief, the platform provides the following key features:

- **availability**, **integrity** and **non-repudiation** as requirements for obligatory reporting
- decentralized **incentives** by leveraging blockchain tokens for purchase and sale of threat intelligence
- transactional **fairness** for both seller and buyer
- **quality assurance** through a verifier system

The remainder of this paper is structured as follows. In Sect. 2, we first provide an overview of approaches for platforms to report threat information, in particular with a focus on meeting the aforementioned security goals. In Sect. 3, we define requirements for the development of our platform. Section 4 introduces our concept for the storage and incentivized exchange of threat intelligence information. In Sect. 5, we propose the system design for the application of our concept and present the implementation of our prototype. The cost structure and thus the practical feasibility of our prototypical implementation are evaluated in Sect. 6. The results of this paper are discussed in Sect. 7, and the paper is concluded in Sect. 8.

2 Related work

The exchange of threat information has been the subject of practical and legislative work in recent years. These include laws in different legislations, such as the NIS Directive[1] in Europe and the IT-Sicherheitsgesetz (BSIG)[2] in Germany, which stipulate the reporting of incidents for providers of critical infrastructures. These legislations are also influenced by data protection requirements, which are, for example, specified by the General Data Protection Regulation (GDPR)[3] or the California Consumer Privacy Act (CCPA).[4] Such regulations have also been addressed in the literature. Schwartz et al. point out fundamental legal aspects of the CTI exchange [3], while Laube and Bhme show that not reporting security incidents may lead to fines in different countries[4]. In addition to this, Bauer et al. have shown in their study on Threat Intelligence Platforms that trust, data integrity, a high platform availability, reporting functionalities as well as data quality are among the key characteristics of CTI platforms[5]. At the same time, the actual exchange of CTI data is already being implemented in practice by various platforms. Examples are IBM X-Force [6] or Facebook threat exchange [7] as commercial platforms as well as MISP [8] and OPENCTI [9] as open source platforms. These platforms allow the exchange of threat information; however, data integrity or availability is not conclusively assured and incentive structures are not available. Central providers can advertise data integrity and availability, but ultimately it is always necessary to rely on the provider to ensure the protection goals are met. This is particularly problematic in the area of possible obligations to provide evidence, as manipulation of the data stock

[1] https://eur-lex.europa.eu/eli/dir/2016/1148/oj.

[2] https://www.gesetze-im-internet.de/bsig_2009/BJNR282110009.html.

[3] https://eur-lex.europa.eu/eli/reg/2016/679/oj.

[4] https://leginfo.legislature.ca.gov/faces/billTextClient.xhtml?bill_id=201720180AB375.

cannot be ruled out with central providers. At the same time, a single provider usually also represents a single point of failure when it comes to the availability of the platform. Furthermore, existing providers do not yet offer functionalities for quality-assured trading and thus an incentivized exchange of CTI information. In this context, Liu et al. [10] showed that a lack of incentives can even prevent the exchange process from happening. In addition to this, Wagner et al. [11] pointed out the risks that are associated with sharing CTI, which in turn may prevent companies from participating in the exchange, which in the worst case can even lead to the information exchanged being used to attack participants in the exchange.

A great deal of research has been done on the requirements and challenges of implementing CTI platforms. In an early work, Serrano et al. [12] point out the fundamental problems for the exchange of threat information. Dandurand et al. [13] defined requirements for the exchange of information, emphasizing the necessity of assuring data integrity and availability, which is also supported by the work of Brown et al. [14]. Mohaisen et al. pointed out various open research questions in that field, such as possible dangers and negative incentives that may relate to the exchange of CTI [15]. In addition to this, there are also works that deal with specific implementations of CTI platforms, such as the MISP platform by Wagner et al. [8]. However, neither specific integrity or availability requirements nor the integration of incentives is considered. The literature also provides works that address the necessity of creating incentives for the exchange of CTI. Sauerwein et al. conducted an exploratory study that showed a need for incentivizing stakeholders within the exchange process [16]. This work is supported by Sillaber et al. examining the needs of stakeholders and resulting challenges [17]. While these studies provide possible starting points for the use of incentive procedures, the actual use of such procedures within CTI platforms is not considered. Moreover, there are also first approaches that try to implement CTI exchange on decentralized platforms. Alexopoulus et al. present a method for sharing security data streams based on a smart contract and data stream subscriptions [18]. Since the proposed data streams require a direct connection between the parties, the assurance of integrity and availability cannot be guaranteed. Incentive structures are also included in the work, but the design suffers from various weaknesses. Since the described on- and off-chain interactions of buyer and seller are independent of each other, negative consequences for fraud attempts during data transfer can only be implemented to a limited extent. In addition, the quality of the incident can vary during a stream, but only the entire stream can be evaluated by a buyer. This increases search costs on the marketplace because information about alerts is only available in aggregated form. Gong and Lee follow a similar approach with the proposed BLOCIS framework[19]. Here, incident data are also not seen as individual items, but aggregated in threat intelligence feeds. Accordingly, a dedicated assignment and a separate reporting functionality cannot be implemented here. At the same time, the proposed concept only provides automated quality assurance. Homan et al. pursue a different approach by implementing CTI sharing on a private Hyperledger blockchain. This work also shows the potential of blockchains in the CTI sharing area. Although, the possibilities of incentives are briefly described, they are not specifically addressed. Quality assurance and reporting requirements are not considered [20]. The papers shown here also show how important fair and secure exchange is for decentralized platforms. Accordingly, much research has been done in this area in recent years. For example, Shafagh et al. [21] show how homomorphically encrypted data can be exchanged securely. Wagner et al. [22] propose an approach to exchange digital goods based on mediator smart contracts, which enable dispute resolution. This work provides the means for decentralized exchange of CTI with smart contracts, but does not provide a browsable platform or quality assurance.

In summary, it can be stated that different works exist in the area of threat intelligence exchange that consider the requirements and the application of platforms. However, to the best of our knowledge, there is currently no work that allows an incentive-based, fair exchange of CTI information, while maintaining data integrity and availability to comply with regulatory requirements. Accordingly, we briefly summarize the novel points where our solution goes beyond existing work:

- sharing of **individual** incidents on a decentralized marketplace
- **quality assurance** by independent and incentivized verifiers
- support of **legal requirements** for obligatory threat reporting
- token-based **incentives** for voluntary sharing without transaction fees

3 Objective and requirements

The exchange of threat intelligence information can be categorized into two different areas. On the one hand, unidirectional reports of security incidents are stipulated by law and mostly concern companies that are relevant for the functioning of society. On the other hand, bidirectional exchange of security information between companies is done on a voluntary basis. The goal is an improvement of the information basis on security incidents for all participants and to increase their security level. The platform developed in this work aims to cover both use cases by enabling both reporting and exchange of security incidents. We consider the use cases **obligatory reporting** and **incentive-based exchange** of CTI information separately, as they should be independent features on the platform. However, a combination of both approaches should optionally be possible. There are different and unique requirements that result from each of these use cases, which are described in more detail below.

3.1 Requirements for reporting security incidents

The most important requirement for reporting security incidents is compliance with the underlying legal framework. We have taken the German IT Security Act [23] as the basis for our requirements in this matter. In doing so, we first consider the concrete effects of the reporting obligation and derive reporting requirements from this before specifying them in more detail. In principle, existing reporting obligations usually specifically oblige operators of critical infrastructures to report possible outages. The German IT security law explicitly proposes a contact point for the implementation of the reports, which can be used by various parties subject to reporting. This ultimately corresponds to the platform proposed here as a common reporting infrastructure. In terms of content, the law specifically stipulates that failures must be reported immediately and thus places an initial focus on availability. The information should include various technical details as well as information on the respective operator. Within the legislation, a further focus is put on the auditability of critical infrastructures. This shows that ensuring the integrity of the reports is also a key factor in the scope of reporting. In addition, the legislation provides for penalties for failing to report security incidents. Accordingly, it is also important to be able to provide proof that a report has been carried out.

According to this, the first requirement for a functioning reporting infrastructure is that a company must be able to provide incident data and that the legal authorities can obtain these data. Reports on security incidents are to be regarded as time-critical, as the judicial authorities may have to react to reported incidents in good time. For this reason, the provision of a very high **availability** is a key factor in the operation of a reporting infrastructure.

In the context of reports, it is also of utmost importance to be able to prove who submitted a report. On the one hand, this is necessary so that authorities can take the necessary steps to prevent supply bottlenecks, for example. On the other hand, this also provides a guarantee for the reporting institution, as it enables it to prove that the reporting obligation has been fulfilled and thus avoid penalties. This necessity results in the requirement of **non-repudiation** and unambiguous assignment of reports. In addition, a further requirement results from the actual use of the data. Besides being used to prevent damage, the threat intelligence information obtained may also be used as evidence. Specifically, recorded data may either be used as evidence in court proceedings or as proof of damage against contractual partners such as insurance companies. Following this, ensuring data **integrity** is an additional requirement in the reporting process that needs to be taken into account. Besides regulations that stipulate reports of security incidents, there are also regulations regarding the handling of personal data in different jurisdictions, such as the GDPR in the European Union. According to this, the platform must also provide the necessary tools to allow the protection exchanged data in compliance with legal regulations.

3.2 Requirements for an incentive system

In addition to requirements resulting from legislation, there are also functional requirements for exchange platforms. Every exchange of information on security incidents is accompanied by various risks. When publishing information, companies risk to accidentally leak important data. This may, for example, include company secrets or information about the company infrastructure that may, for example, simplify attacks on that company. In addition, a reporting process involves costs for the collection, processing and dissemination of incident data. At the same time, the benefits of participating in an exchange platform are often difficult to quantify, especially with comparatively low legal penalties for omitted reports. From these points it can be concluded that companies tend to have little intrinsic motivation to report incidents themselves, whereas the motivation to passively obtain information from a reporting platform is likely to be high. As a result, an **incentive system** that motivates every participant on such a platform to actively participate can be defined as a further essential requirement for the sustainable functioning of such a platform (**RQ2**).

3.3 Platform comparison

As shown above, several other platforms already exist that enable the exchange of threat intelligence information. Among them are both centralized and decentralized concepts covering different use cases. This leads to the problem that platforms offer different features and each of them offers its own approach to addressing protection targets. In order to demonstrate the advantages of the DEALER platform compared to existing concepts this section compares the DEALER platform to Facebook Threat Exchange (FB-TX), IBM X-Force, MISP, OpcenCTI and Trident as presented in Sect. 2. Specifically, the key features of DEALER for the creation of incentives as well as the implementation of reporting obligations are compared individually for all platforms. More specifically, the aforementioned protection goals of availability, integrity, non-repudiation, fairness, quality and

Table 1 Comparison of CTI sharing platforms

	FB-TX	X-Force	MISP	OpenCTI	Trident	DEALER
Platform availability	◐	●	◐	◐	●	●
Data availability	●	●	●	●	○	◐
Integrity	○	○	○	○	◐	●
Non-repudiation	○	○	○	◐	●	●
Incentives	○	○	○	○	●	●
Fairness	○	○	○	○	◐	●
Quality assurance	○	○	○	○	◐	◐

○ Not addressed, ◐ insufficiently addressed, ● explicitly addressed

the possibility of creating incentives for exchange are considered. Depending on the use case of the notification or exchange, the availability of the platform itself as well as the availability of the data in the specific case are also distinguished for the determination of the availability protection goal. This separation is introduced since one problem within decentralized platforms is to keep exchangeable data available for trade at all times. The full comparison made here is shown in Table 1 and is explained in more detail below. The comparison rates different platform protection goals with values from "not addressed" through "insufficiently addressed" to "explicitly addressed". Significant differences between centralized and decentralized platforms are apparent at first glance in Table 1. In the following, these are broken down in more detail once again.

Platform availability. In the FB-TX, MISP and OpenCTI platforms, platform availability is not given special consideration. No specific statement can be made for MISP, as it is operated independently by different communities. Moreover, FB-TX already had several outages in Q2 2020. [5] Accordingly, no increased availability can be assumed for these platforms. X-Force addresses this problem using increased parallelization, however, outages regularly occur here as well. Trident and DEALER, on the other hand, are operated on decentralized blockchains ETH and EOS, on which outages are very rare due to the high number of network nodes.

Data availability. This property is naturally very high for all central platforms, as data can be uploaded and is available regardless of the status of the user. In contrast, Trident is dependent on the availability of the user and has a correspondingly low data availability. Despite the decentralized approach, DEALER tries to address this problem by sharing the load between several users. This is explained in more detail in Chapter 4.3.

Integrity. The data integrity is not considered in any of the centralized approaches whereas Trident and DEALER implement them. While this is only partially addressed in Trident as the data exchange is based on a direct stream, the DEALER concept provides for a complete integrity assurance of the data. At the same time, data integrity is a characteristic that can be ensured particularly well by decentralized platforms without the need for trust.

Non repudiation. This property is only addressed in OpenCTI, Trident and DEALER. With OpenCTI, however, this is only partially the case, as this is ensured by the platform administrator and a corresponding level of trust is required. **Incentives:** Both Trident and Dealer offer the possibility of evaluating incident data and exchanging it via a marketplace. Such incentive mechanisms are not provided for in the central platforms.

Fairness. Within the DEALER platform exchange fairness is specifically addressed. Trident also addresses this, however, only peripherally by creating a relationship of trust between the participants. On the central platforms this problem is currently not considered at all.

Quality assurance. This property is only addressed by the decentralized approaches so far. Central platforms do not yet take this into account. However, MISP names quality assurance as an important goal for Future Work.

Overall, this comparison shows that the implementation of a decentralized CTI exchange platform can create various advantages for the exchange. These include features such as the creation of incentives, quality assurance and the integration of fairness mechanisms. These were not implemented on existing platforms although there are no significant technical obstacles. On the other hand, they also include criteria such as platform availability, integrity assurance and non-repudiation. Due to the inherent characteristics of decentralized platforms, these can be mapped very well and without the need of ensuring trust. Although the concept of the DEALER platform does not fully meet all criteria, it is clear that such a system is superior to traditional approaches in many respects.

3.4 Shared CTI data

In general, any data format can be used for the exchange within the platform. Within this work as well as within the development of the platform, we have used the state-of-

[5] https://developers.facebook.com/status/dashboard.

the-art data format STIX2 for the exchange as well as the reporting of incidents.

```
{
  {
    "type": "bundle",
    "id": "bundle--2a25c3c8-5d88-4ae9-862a-
        cc3396442317",
    "objects": [
    {
      "type": "indicator",
      "id": "indicator--1",
      "created": "2014-02-20T09:16",
      "name": "File hash for Poison Ivy
          variant",
      "description": "This file hash
          indicates that a sample of Poison
           Ivy is present.",
      "indicator_types": [
          "malicious-activity"
      ],
      "pattern": "[file:hashes.’SHA-256’ =
          ’9ef537f25c895bfa7825...’]",
      "valid_from": "2014-02-20T09:00"
     },
     {
      "type": "malware",
      "id": "malware--2",
      "created": "2014-02-20T09:16",
      "name": "Poison Ivy",
      "malware_types": [
          "remote-access-trojan"
      ],
      "is_family": false
     },
     {
      "type": "relationship",
      "id": "relationship--3",
      "created": "2020-02-29T18:09",
      "relationship_type": "indicates",
      "source_ref": "indicator--1",
      "target_ref": "malware--2"
     }
    ]
}
}
```

Listing 1 Exemplary STIX 2 bundle

Listing 1 shows a shortened excerpt of such an exemplary STIX2 data packet, which is basically provided in the JSON data format. Such a data packet is always enclosed by the structuring unit *bundle* which allows a unique assignment of the data packet. Such a bundle contains the various STIX objects and references between these objects. The shown example contains with the object "indicator-1" information about an indicator, i.e. a pattern that indicates a possible security incident. Furthermore, the object "malware-2" contains information about the detected Malware "Poison Ivy". Finally, a connection between the two objects is established by the relationship object "relationship-3", which references both objects. In addition to an impression of the basic syntax and design of a security incident with STIX2, this example also gives an insight into the dynamic data model of STIX2. A bundle object can contain any number of STIX objects, which in turn can be dynamically connected to each other by means of relationship objects.

4 The DEALER sharing concept

In this section we present the DEALER concept, which is designed to fulfill the previously defined requirements and to provide an incentives structure for sharing CTI information. This includes an ecosystem describing the stakeholders in the system, their roles and relationships and a marketplace describing the processes and concepts within the ecosystem, designed to guarantee sustainable CTI exchange. This section provides an overview of the relationships within the system and the overall idea of the concept. The individual processes within the system are described subsequently.

The entire system, which is outlined in Fig. 1 consists of five essential components. At the center of the system is a **blockchain** and a **distributed database**. These form the technological basis for the implementation of smart contracts, integrity-secured storage of exchange processes and provide decentralized storage structures for reported security incidents. The starting point for reports within this system is **Critical Infrastructure Compounds**. These include the critical infrastructure operator, an IT service provider if applicable, and a CTI provider. The CTI provider takes care of external communication and acts as a so-called contact point, a construct that can be derived from legal requirements for incident reporting. The information collected is intended for either Associated Institutions or Organizations. **Associated Institutions** describe participants who are interested in the reported information within the scope of reporting obligations. These can, for example, be legal authorities to which a reporting obligation exists. These can also be other institutions, such as insurance companies, to which a possible claim can be made accessible via the platform. On the other hand, there are **Organizations** that are not affected by reporting obligations, but are nevertheless interested in participating, for example to increase their own level of protection. Analyses and services within the system are provided by the **CTI ecosystem**. This enables external service providers to bring their services into the system. For example, verification providers can offer qualitative incident data evaluation, or analytics providers can aggregate information on several incidents and offer it within the system.

DEALER's overall concept defines two central use cases: statutory incident reporting and incentive-based threat intelligence exchange. Both concepts are briefly described below before we take a closer look at the underlying processes.

Obligatory reports are generated by the Critical Infrastructure Compound and transferred to the blockchain. The transmitted data are pseudonymized and encrypted in such a

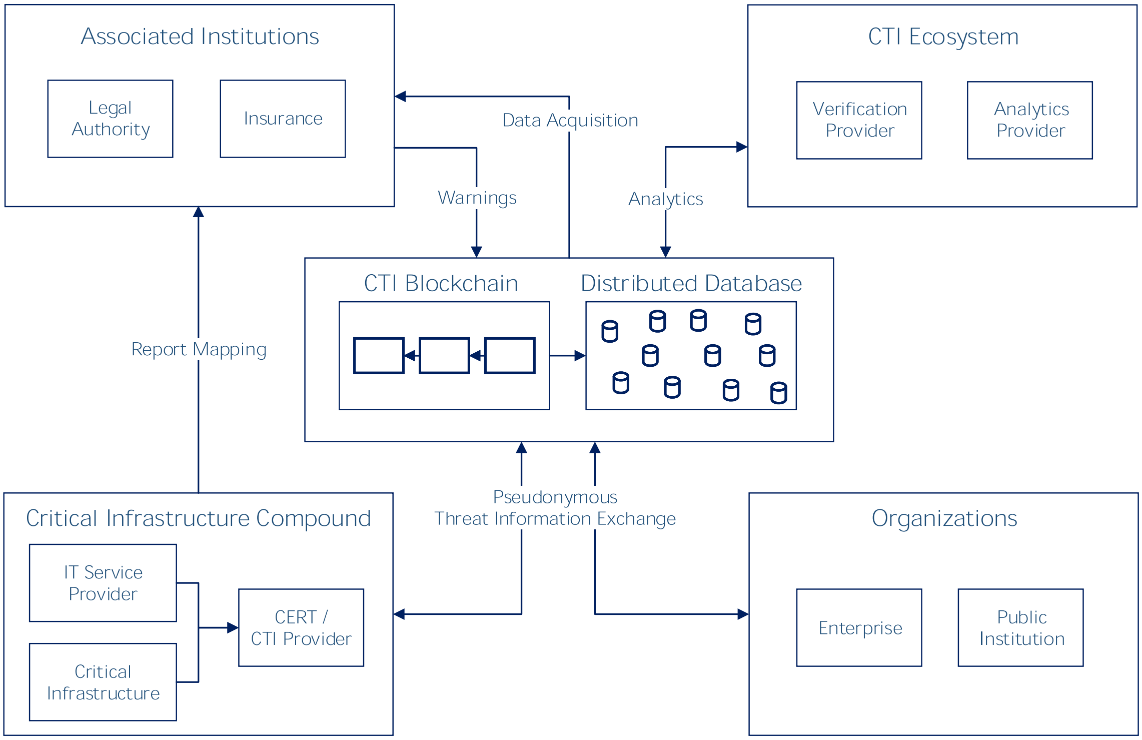


Fig. 1 High level overview of the DEALER threat intelligence sharing concept

way that only the receiving authority can access it. In connection with such a report, the data can also be made available to other users of the platform as part of the incentive-based exchange. However, this step is explicitly optional and must be actively selected.

The **incentive-based exchange** process is based on an economic model, where participants can offer and demand information on security incidents. For this purpose, a separate token is introduced on the platform, which functions as an internal currency and is used as economic reward for active participants. When threat information is provisioned, structured incident data are transferred to the blockchain in encrypted and pseudonymized form. The information provided can then be sold to other participants or made available as a report. The uploaded incident information is assigned to verifiers who ensure its data quality against a fee. After successful verification, the data can be traded on the platform at the previously defined price.

In addition to these two sharing mechanisms, legal authorities may additionally issue global warnings regarding threats to all participants. In some legislations, such as the IT security law in Germany [23], such global warnings are part of the reporting obligation and thus necessary for compliance. The warnings also represent an additional benefit for the platform participants: the free CTI provided by the legal authority supplements purchasable incident information.

After this high-level introduction to the basic concept of DEALER, the core processes of the platform are presented in more detail below. They include Registration (4.1), Sharing (4.2), Verification (4.3), Purchase (4.4) and Fairness (4.5).

4.1 Registration

Initially, participants must register to be able to transact on the decentralized marketplace. Each participant has an account with a balance of fungible tokens, which may be used to trade incidents. To prevent sybil attacks, we require a fixed initial token stake s_i to create the participant's balance. This prepayment requires a meaningful investment, while not deterring new users. The user balance is managed by the platform. Withdrawals are allowed on request up to the initial fee, which must remain until the participant closes the account.

Verifiers are treated separately during registration, as they are given free access to incident information and must evaluate it. The purpose of registration is to achieve a unique identification of the verifier, for example by requesting a tax number, identity documents or a social security number. This registration process is intended to prevent the risk of verifier misuse (i.e. free-riding or submitting default ratings). In contrast to regular participants of the platform, verifiers must be approved before participating in the verification process. During bootstrapping of the verifier pool, approval can be conducted by the platform developer. Once the verifier pool has reached the minimum size (Sect. 7.3), new participants can be approved through majority votes of existing participants.

Additionally, the platform provides an exclusion option for malicious verifiers. Exclusion of a verifier must be approved by a majority of the verifier pool through multisignature votes. Any verifier may initiate such a vote by providing evidence for several instances of misbehavior (i.e. repeatedly submitting default or unrealistic verification reports).

Besides preventing misuse, the goal of authenticating verifiers is to ensure an intrinsic interest in the analysis of security incidents and possession of the necessary technical expertise for actual incident information assessments. Appropriate verifiers could, for example, be threat intelligence vendors, CERTs or security operations professionals.

4.2 Sharing

Figure 2 shows a BPMN diagram of the sharing process from incident detection to data upload, verification and provisioning on the platform. Initially, the participant locally performs required preprocessing steps. These include anonymization (removing private data and identifying details), addition of public descriptive metadata and encryption of the incident with a symmetric key k. The metadata also include a sale price p_s. A signed transaction is submitted to the platform and the incident is uploaded to the distributed database. If the participant decides to sell the incident to other users, a verification fee p_v must be paid once with the initial transaction. We suggest $p_v \sim 0.6p_s$ to reward verifiers depending on the value of the incident. The incident is then made available on the marketplace and verification is initiated. Three random verifiers are chosen from the verifier pool. The seller then uploads three keys $k_{v1}/k_{v2}/k_{v3}$ for each chosen verifier, encrypted with each verifier's public key, and notifies the platform at time T_1. The verifiers retrieve and decrypt the uploaded incident with their individual key file. They assign an initial rating value based on a set of platform-provided quality metrics (Sect. 4.3).

The verifiers submit the verification result to the platform. If all results arrive until time T_2, the verification fee p_v is distributed equally among the verifiers ($\frac{p_v}{3} = 0.2p_s$ per verifier, as noted above). If any verifier does not respond, the seller may trigger a replacement of inactive verifiers. These verifiers must respond until time T_3 ($T_3 > T_2 > T_1$), else the seller may request a removal of the incident from the platform and partial reimbursement of the verification fee ($\frac{p_v}{3}$ per missing verifier).

For obligated incident reporting, the participant may want to keep the incident confidential and not share it with verifiers. In this case, the participant only uploads a key for the regulatory authority and no verification is performed. The platform provides a timestamp and proof of reporting for the incident.

4.3 Verification

The data quality verification conducted by verifiers serves as an incident reputation bootstrapping mechanism. We propose a 5-point rating scale for incident quality from 1 (very low) to 5 (great). The verification needs to be as objective and meaningful as possible to provide guidance for buyers, since the actual data are encrypted. The following items serve as verification guidelines:

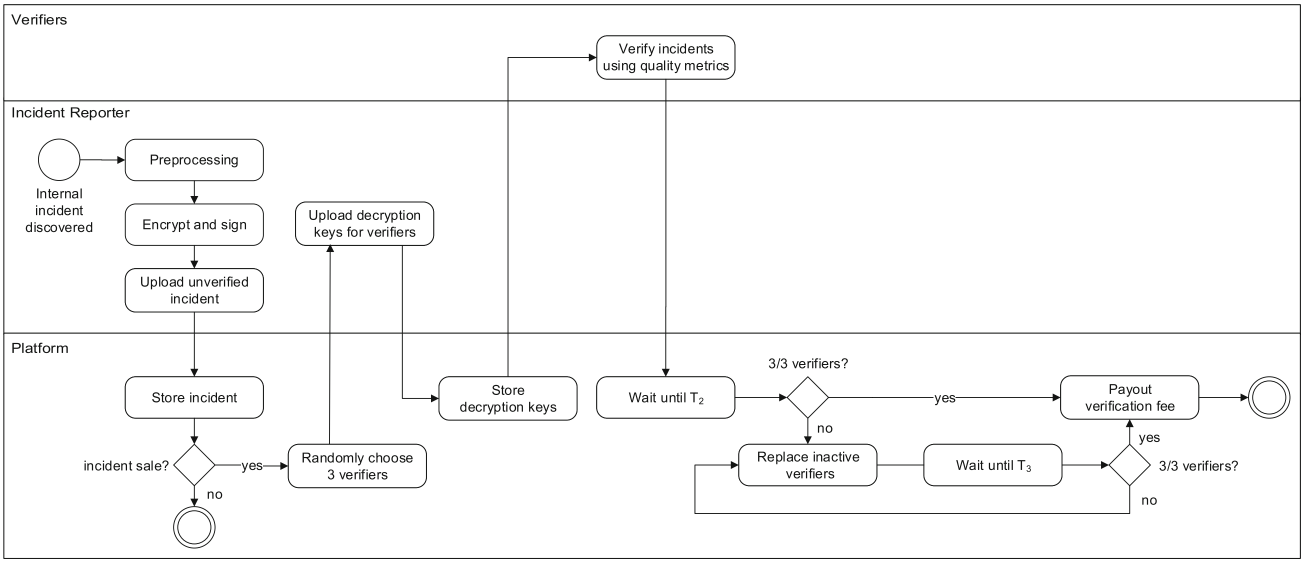


Fig. 2 Incident sharing process on the DEALER platform

- consistency with metadata of the seller's previous incidents
- similarity check for incident metadata and verified incidents
- assessment of various threat intelligence quality indicators [24]

After receiving the incident data, each verifier independently performs a verification of the contained information. A basic consistency check using metadata of the seller's previous incidents verifies that the incident originates from the same industry. To avoid duplicates and resold incidents (Sect. 7), verifiers compute a similarity score to other previously downloaded incidents (i.e. using the *simhash* algorithm [25]). For apparent duplicates, verifiers then submit a low score without additional quality assessment.

Regarding threat intelligence quality indicators, the platform provides a structured assessment process. This procedure is intended to help verifiers make objective and comparable assessments of security incidents by iteratively processing predefined questions.

To achieve this, the implemented questions are based on objective CTI data quality indicators developed for STIX2 [24]. The quality criteria are divided into three major domains. These include information about the contained data, object representations within the data and the completeness of the available information. In particular, the data model domain reflects information about the *representational consistency* of the data representations and the *concise representation* of the stored information. The object metrics area considers the *objectivity* of the data collected as well as metrics about the *relevancy* of the stored data regarding the situation described. The third domain addresses the completeness of the available information in more detail. This includes the examination whether an *appropriate amount of data* is used to convey the facts presented. In addition to this, the *syntactic accuracy* of the data transported as well as the *schema completeness* of the data is checked.

4.4 Purchase

The incident purchase in Fig. 3 process starts off with a potential buyer browsing the repository of previously uploaded incidents. For this purpose the platform front end offers sophisticated search and filter functionality. Metadata and ratings are provided for each individual incident by verifiers and past purchases. Once an incident of interest has been identified for purchase, the buyer retrieves the encrypted incident to verify its availability. If the incident is available, the buyer places an order for the incident and pays tokens corresponding to the sale price p_s to the platform escrow. After the order has been placed successfully, the key for decrypting the data record is released in the next step. In order to speed up this procedure and not to have to wait for the presence of the seller this can be done either by the seller or by the verifiers. This is possible because all verifiers involved also possess a valid key k_1, k_2 or k_3, as shown in Sect. 4.2. For successful purchases, the key is automatically issued in the background where the decryption key is encrypted with the public key of the buyer k_b and uploaded. In case of successful decryption, the buyer notifies the platform by sending a confirmation along with an incident rating.

If the decryption fails, the buyer notifies the platform about the failure, which initiates the dispute resolution process. Any verifier may then provide an independent copy of the decryption key to the buyer. In the unlikely event that the buyer is

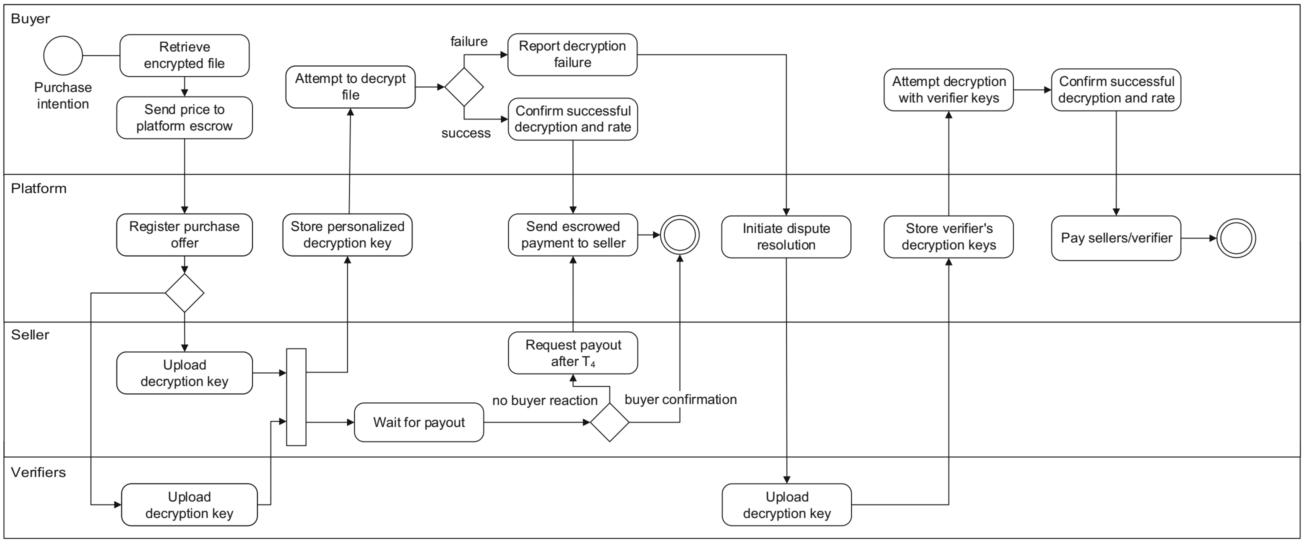

Buyer
Purchase intention
Retrieve encrypted file
Send price to platform escrow
Attempt to decrypt file
failure
success
Report decryption failure
Confirm successful decryption and rate
Attempt decryption with verifier keys
Confirm successful decryption and rate
Platform
Register purchase offer
Store personalized decryption key
Send escrowed payment to seller
Initiate dispute resolution
Store verifier's decryption keys
Pay sellers/verifier
Seller
Upload decryption key
Request payout after T₄
no buyer reaction
buyer confirmation
Wait for payout
Verifiers
Upload decryption key
Upload decryption key

Fig. 3 Purchasing process on the DEALER platform

still unable to decrypt the file, keys must be uploaded by additional verifiers to resolve the dispute. Once the buyer is able to decrypt the file, the buyer submits a rating for the incident and closes the dispute. For providing decryption keys, contributing verifiers receive an equal share of the dispute fee p_d as an incentive. The dispute fee is deducted from the sale price and should be proportionately low for two reasons. First, by monitoring the blockchain for disputes verifiers can upload key copies in an automated fashion, requiring little effort and thus little incentive. Second, sellers should not lose a large amount of the sale price in case of unwarranted disputes caused by dishonest buyers. However, dishonest sellers should still be punished to encourage honest incident sale. We thus suggest an initial value of $p_d \sim 0.10 p_s$, subject to further practical evaluation.

A time lock T_4 is in place to allow parties to redeem their tokens if the counterparty fails to respond. If the buyer does not report decryption success or dispute, the seller may collect the sale price after T_4 has expired. If the seller never accepts the offer, the buyer may redeem the locked tokens after T_4.

4.5 Fairness

Fairness of incident purchase must be considered from two perspectives:

– **Seller fairness**: An honest seller is guaranteed to receive the advertised sale price for providing a correct decryption key.
– **Buyer fairness**: An honest buyer is guaranteed to receive the plaintext of the purchased incident, or is refunded the deposited purchase price.

We guarantee Fairness based on the following assumption: There is always at least one honest verifier that provides a valid decryption key. After verification, there are at least four copies of the decryption key (the seller and three verifiers) available on the platform. It is reasonable to assume that there is at least one honest participant among these four, which provides a decryption key in case of an issue with the seller's key.

We now analyze the various ways how seller and buyer may attempt to cheat, and how the protocol mitigates these attempts.

Buyer fairness. The honest verifier assumption means that the buyer will always receive a decryption key, and that there is no scenario where the buyer will not be able to decrypt the file. Conversely, the buyer will also not receive the deposited price back. In case the seller attempts to cheat by uploading a wrong decryption key for the buyer, the buyer can initiate a dispute to receive a correct key from a verifier. Verifiers receive a dispute fee p_d as participation reward for uploading correct keys during a dispute. The seller is thus disincentivized to send wrong keys, since that increases the likelihood of a dispute and results in a loss of p_d tokens.

In case both seller and verifier keys are incorrect, the buyer may be unable to decrypt the item at all. This will not occur in practice based on the assumption that the majority of verifiers is honest and provides correct keys. This assumption can be made based on two properties of our platform:

1. random assignment of verifiers to incidents makes seller-verifier pairings unlikely, and repeated collusive arrangements are time-consuming
2. misbehavior is disincentivized through significant verifier registration requirements (Sect. 4.1) coupled with the possibility of exclusion

We have thus ensured that the seller is punished for uploading wrong key material, while the buyer is able to decrypt the purchased file. To increase the buyer's confidence in receiving a correct key, the time of last platform activity of an incident's verifiers can be shown in the user interface.

Seller fairness. The buyer may attempt to cheat the seller by not responding after the seller has provided the decryption key. For this reason, there is a deadline for the buyer to respond, which starts from the time the seller has uploaded the key and ends after time T_4. If there is no response after expiry, the seller may redeem the purchase price.

The buyer may also collude with the verifiers to falsely vote for seller misbehavior. In this case the honest seller would lose out on p_d tokens deducted from the sale price. This scenario is unlikely, since the buyer has no incentive to collude with verifier. If buyer and verifier are in contact, they could exchange data and tokens through another channel with a reduced price. In practice, this is unlikely to occur, since there is a large overhead for buyers to contact verifiers for every incident they are interested in.

If not colluding with a verifier, the buyer has no incentive to blame the seller. He cannot receive any tokens back that were paid for the sale, and he is guaranteed to receive a correct decryption key if at least one verifier is active.

These considerations guarantee Seller Fairness, with the restriction that the seller may lose out on a small portion of the sale price p_d in case of a dispute. Disputes cannot be prevented by the seller, but buyers have no incentive to start disputes, so we expect them to be negligible in practice.

5 Application prototype

To implement the sharing concept, we choose a combination of blockchain technology and distributed hash tables. This avoids having to trust a single third-party service provider to provide storage and confidentiality. A data storage distributed in this way can be maintained collaboratively and only by participants interested in sharing data. Blockchain networks also allow utilizing virtual currencies that provide possibilities to realize built-in sharing incentives for participants. In the following we first discuss the technologies used for our prototype (5.1). Subsequently, we develop the conceptual architecture (5.2) and briefly present our prototypical implementation of the sharing platform (5.3).

5.1 Technology selection

In this section we will first discuss the underlying technologies for our sharing platform. This includes the permission model, the approach for storing incident data as well as the chosen blockchain platform.

Blockchain platform. The first consideration when deciding on a blockchain platform is the choice between a permissioned network and a permissionless public blockchain. *Permissioned* networks consist of a fixed set of participants that each operate a node of the private network. We experimented with the permissioned blockchain Hyperledger Fabric, but found many obstacles during our research that made it unsuitable for the DEALER platform. These include missing native token support, no means to exchange tokens for fiat currency, and the increased barrier to entry caused by the need to deploy and operate a private Hyperledger Fabric node. The latter results in high initial costs and maintenance costs for updates and monitoring, while availability is less certain due to the limited number of blockchain nodes. *Permissionless* blockchains are operated by independent miners that are incentivized through mining rewards distributed by the consensus protocol (i.e. Proof of Work or Proof of Stake). The blockchain infrastructure is thus already available, but transaction fees must be paid to the maintainers of the platform. Public blockchains also provide a high number of distributed nodes that guarantee high availability, while token distribution can be handled transparently using existing exchanges. Since high availability and incentives for participants are essential aspects of our concept, we choose a *permissionless* blockchain approach for our concept.

Commonly, researchers use Ethereum for permissionless blockchain application prototypes due to its good tool support and large developer community [26]. Unfortunately, the intermittently high transaction costs[6] represent a barrier to entry and reduce the ability to provide incentives for participants. The low maximum transaction throughput of around 15 transactions/second [27] amplifies this issue, as transaction fees rise when the network is congested. This problem is exacerbated when transaction demand increases to extreme levels [28]. Therefore, after evaluating both permissionless and permissioned blockchains, we settle on the EOS blockchain[7] for our implementation. We utilize EOS as opposed to other permissionless blockchain platforms like Ethereum for several reasons. First and foremost, EOS does not charge users transaction costs. Transaction allowances are determined based on staked EOS tokens, thus lowering the long-term cost of using the platform. In addition, EOS provides more scalability regarding transaction throughput (up to 8,000 transactions/second [29]). The EOS network itself is maintained by hundreds of nodes around the world using delegated Proof of Stake (dPoS) consensus. 21 active block producers are selected from a list of candidates[8] based on the votes of EOS token holders. The block producers themselves are encouraged to participate in the network through block rewards (EOS token), which they receive for creating new blocks. Other nodes serve as standby nodes and store a copy of the blockchain, ready to assist if an active producers goes offline or no longer has enough votes. Since the 21 active producers run a deterministic byzantine fault-tolerant protocol among each other, at least 15 colluding producers are required to take over the blockchain.

Data storage. Due to high costs associated with smart contract data storage, larger data items are commonly stored off-chain in blockchain applications [30]. One way of trading data using blockchain is settling the trade on-chain and trading the actual data off-chain [18]. This avoids the need for another storage platform besides the blockchain. However, it also requires the seller to re-upload data to every buyer, which means that both seller and buyer need to be online at the same time. A decentralized off-chain storage platform avoids this issue. To ensure an integrity link between the blockchain network and the off-chain store, the database should be content-addressable. Since only encrypted information is stored off-chain, access control is not required. Distributed Hash Tables (DHTs) provide these properties: they offer public, distributed and content-addressable key-value data storage. We opted for IPFS[9] as the DHT implementation in the prototype. IPFS is widely used in research as an off-chain storage solution, and it provides the features needed for sharing CTI data and encryption keys.

In the DEALER prototype, each participant operates a IPFS node. IPFS nodes are simple to set up; after installation only a single command is required to start the daemon. We use these IPFS nodes to obtain fixed address for each peer for sharing dynamic content, referred to as its *IPNS address*. The node's IPNS address is based on the hash of the peer's public key and can only be updated with a signed update from that peer. We exploit this functionality to statically address each user's shared incidents and decryption keys. We leverage the

[6] https://bitinfocharts.com/comparison/ethereum-transactionfees.html.

[7] https://eos.io.

[8] https://bloks.io.

[9] https://ipfs.io.

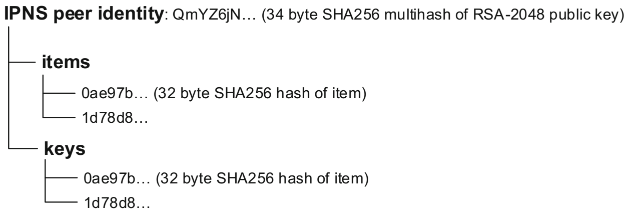

IPNS peer identity: QmYZ6jN… (34 byte SHA256 multihash of RSA-2048 public key)
items
0ae97b… (32 byte SHA256 hash of item)
1d78d8…
keys
0ae97b… (32 byte SHA256 hash of item)
1d78d8…

Fig. 4 IPFS off-chain storage folder hierarchy (for each user)

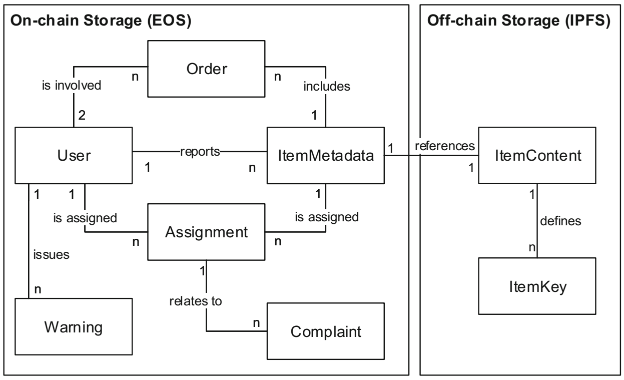


Fig. 5 Simplified entity relationship model of data stored on the DEALER platform

IPFS Mutable File System to create a local folder hierarchy corresponding to the files we intend to share (Fig. 4). The root hash of this folder hierarchy changes every time an item or key is added to a folder. Each time that happens, the updated hash is published to the peer's IPNS address. Other peers can resolve this address to retrieve the latest incidents and keys shared by other users. By *pinning* content hashes, verifiers permanently replicate the encrypted incident shared by the seller to ensure its availability. Verifiers are incentivized to replicate seller content, since they potentially profit from each sale in case of a dispute (Sect. 4.5).

5.2 Architecture and data model

As shown in Fig. 1, the prototype architecture consists of a smart contract on the EOS blockchain platform and IPFS-based decentralized storage. The blockchain platform provides executable smart contracts that implement the *Platform* role in the processes described in Sect. 4. IPFS provides storage capabilities for reported incident data and encryption keys. It also provides pseudonymous identity: Participants sign up with blockchain accounts, which are authorized through public-private key pairs and represented by unique addresses. Figure 5 gives an overview of the platform's data model.

The model shows a distinction between *on-chain* and *off-chain* storage. The on-chain storage manages transaction information and metadata including assignments of users (*User*), reports (*ItemMetadata*), votes *Assignment* and purchases (*Order*) of incident data. The off-chain storage holds the actual incident data (*ItemContent*) as well as the encrypted decryption keys for the information (*ItemKey*). The *ItemMetadata* table contains the reported incidents' metadata, including a short description, the originating industry, the price and a reference to the reporting user. *ItemMetadata* also contains the CTI item's hash, which links the metadata to the full incident data *ItemContent* stored off-chain. Using the hash reference, data can be retrieved from IPFS through a DHT lookup and verification of the retrieved file against its hash reference. The assignment of randomly selected verifiers is done using the *Assignment* table by establishing a link between the verifying user and the respective item. This table also stores the results of item verification, while cumulative results of verification and rating processes are stored in the *ItemMetadata* table. The assignment table is additionally linked to the *Complaint* table, which stores complaints about inaccurate verifications. The *Order* table finally contains the transactions associated to an order, where a transaction establishes the relationship between the buyer and the seller, as well as the item concerned. Besides storing report items, the application also allows the issuance of warnings. These can be inserted by authorities as a specific type of user and stored in the table *Warning* on chain.

5.3 Application prototype

The prototypical implementation of the platform consists of three major components: the smart contract on the EOS blockchain based on EOS C++ code, the IPFS data storage and a DApp (Decentralized Application) front end based on Node.JS. Since Smart Contract and data storage were already described previously, this section focuses on the implementation of the DApp.

Figure 6 shows the implemented components (node.JS server and EOS smart contract) and their interactions with the distributed system. Each DEALER participant runs a node.JS server which manages the encryption keys and blockchain wallet for the organization. It also serves the web interface to internal users. On user requests, the node.JS server interacts with the IPFS network and the EOS test network. CTI data and file keys are stored at the local IPFS node and managed through its IFPS identity. Requests for new CTI data are resolved through the IPFS network. Blockchain transactions are sent to the smart contract on the EOS test network, and data are read back through the EOS node's HTTP API.

Figure 7 shows the user interface of the DApp. The application's user interface offers four fundamental areas tailored to each participant type. The area *BUY* allows potential buyers to get an overview of offers on the platform and to buy and download available incident information. The overview contains a short description of the incident information as well

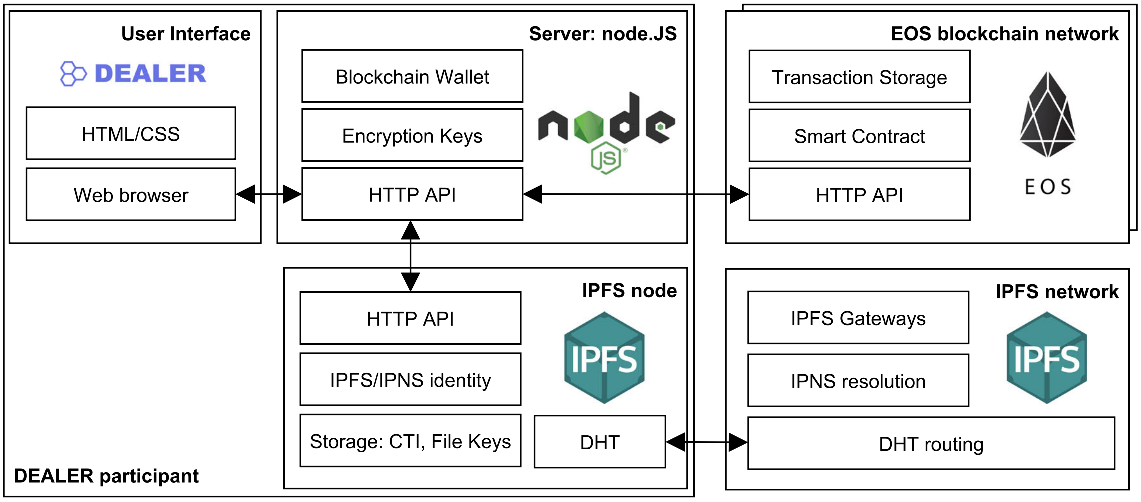
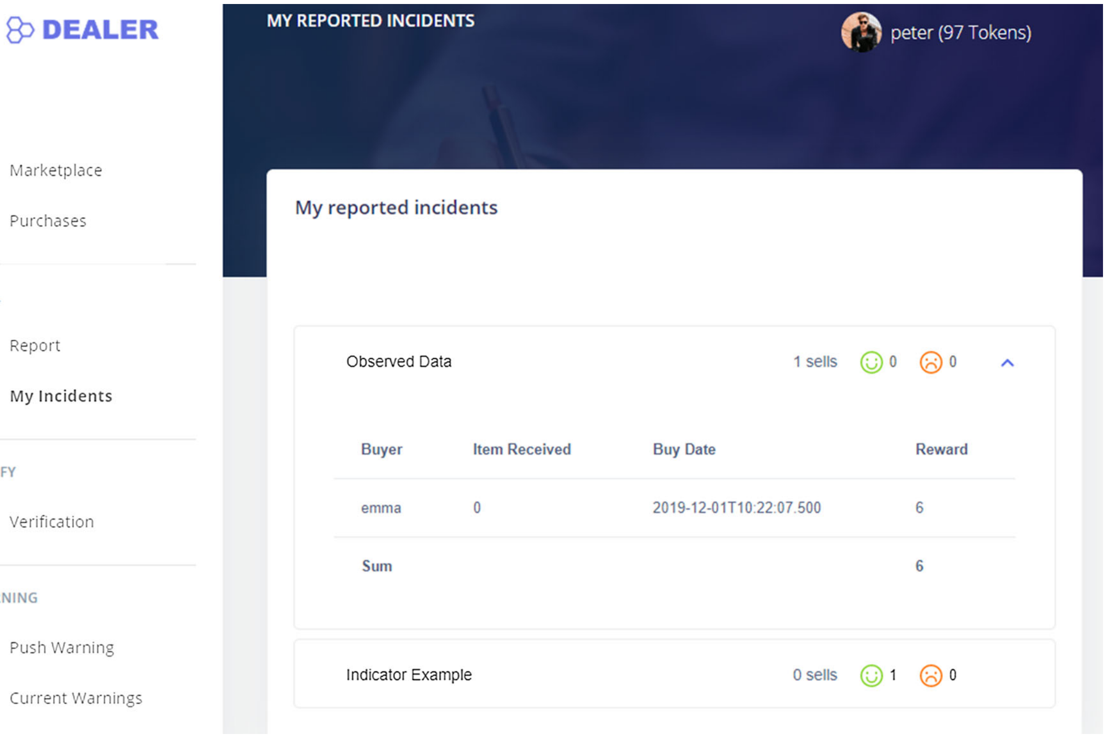


Fig. 6 Prototypical implementation of the DEALER platform



My reported incidents

Observed Data 1 sells ☺ 0 ☹ 0

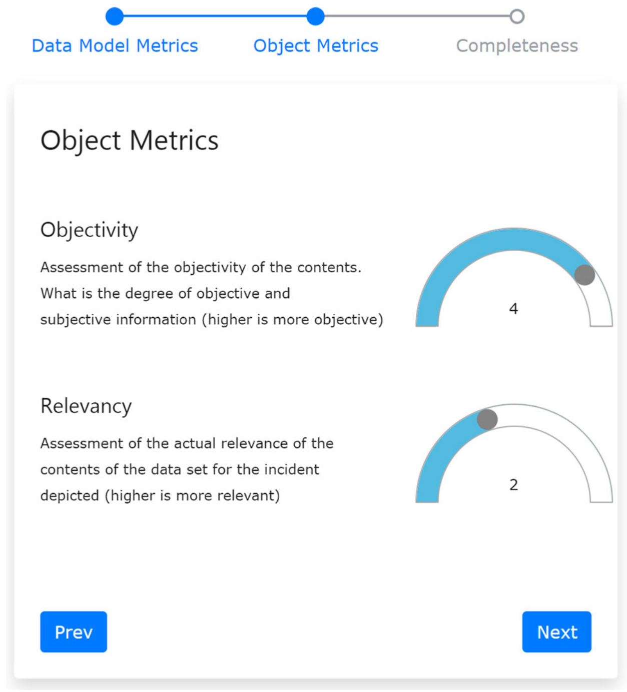
Buyer	Item Received	Buy Date	Reward
emma	0	2019-12-01T10:22:07.500	6
Sum			6

Indicator Example 0 sells ☺ 1 ☹ 0

Fig. 7 User interface of the DEALER platform

as its current verification status and price. Buyers can also manage past purchases and re-download previously bought information at any time. The area *SELL*, allows sellers to report an incident to the blockchain. Such a report can contain a title, a short description, the corresponding industry sector, the actual incident data and a sale price. Incidents are encrypted using AES-256-CBC before being uploaded to IPFS. After the DHT upload, the hash reference and metadata are submitted to the smart contract. If the incident was intended for sale, RSA-encrypted copies of the AES symmetric keys are shared with the verifiers using their public keys stored on the blockchain. Besides reporting, sellers can manage past reports and view the verification status and number of their successful sales.

The *VERIFY* section allows the user to act as a verifier for an incident. The verifier is presented with a list of all incidents assigned for verification. For each individual incident, the verifier is presented with a wizard as shown in Fig. 8. The wizard sequentially requests input for the quality criteria defined in Sect. 4.3. The verification results are arithmeti-



Object Metrics

Objectivity

Assessment of the objectivity of the contents. What is the degree of objective and subjective information (higher is more objective)

Relevancy

Assessment of the actual relevance of the contents of the data set for the incident depicted (higher is more relevant)

Fig. 8 DEALER verification wizard

cally averaged after submission and sent to the platform in a blockchain transaction. Although the prototypical application allows a weighting of the individual quality criteria, this was not implemented within the demonstration prototype for reasons of clarity.

Finally, the area *WARNING* allows authorities to issue warnings on current threats to platform participants. Warnings contain informational text and structured incident information for particularly dangerous threats.

The source code for the prototype can be downloaded at the project repository.[10] A live version of the DApp is available online,[11] and the deployed EOS contract can be inspected on the EOS Kylin testnet.[12]

5.4 Implementation challenges

EOS. EOS developer tools posed some challenges, as the development environment EOS Studio crashed frequently during our tests. Some features did not work as advertised or did not work at all. Another sticking point is that debugging is not possible within the environment and even console outputs are only accessible in a cumbersome way. However, many of these issues were improved with subsequent updates during our research.

Furthermore, achieving scalability of the smart contract is not trivial. EOS allows a maximum of 150ms CPU time per transaction, so performance must be kept in mind while developing the smart contract. For example, loops over table entries must be avoided, since they will lead to exceeded transaction CPU time as tables grow larger. Instead, indexes should be added on the required columns using the `multi_index` table feature. Additionally, page load times increased with an increasing number of incidents. This issue can be resolved by setting appropriate limits on `get_table_rows` queries to the EOS node and paginating results.

IPFS. IPFS is based on a content-addressed DHT data structure. This means that the address of data changes when the data are mutated by an update. It should be kept in mind that the DEALER platform needs to provide a single address for buyers and verifiers to retrieve decryption keys from a seller. With IPNS, IPFS provides a way to get a single fixed address, whose link target (i.e. a folder with keys) can be updated dynamically. Unfortunately, this address is tied to the IPFS node, which means that each user has to operate their own IPFS node. While this may be seen as a limitation of our DEALER implementation, it also comes with the advantage of user data sovereignty. Even if other IPFS nodes go offline, data will remain stored locally once it has been retrieved from the IPFS DHT.

6 Evaluation

After presenting the prototype design, we now evaluate whether the chosen blockchain platform fits the needs of threat intelligence reporting. Since EOS supports > 1000 transactions per second [29], we do not expect throughput to become a bottleneck. However, there are costs associated with transacting on a public blockchain, which we evaluate in Sect. 6.1. Additionally, we consider computation times, network latency and storage requirements in Sect. 6.2.

6.1 Transaction costs

Smart contracts on EOS require CPU, NET and RAM to execute. CPU and NET represent the processing and network utilization of transactions and are acquired by staking EOS for a fixed time. RAM is needed to store data in the smart contract state and is purchased at a fixed price. To calculate the required stake per user to run the contract sustainably, we evaluate the resources consumed by our smart contract in Table 2. Transactions were run multiple times with differing parameters on the EOS Kylin testnet. For CPU/NET, the values represent locked currency, i.e. to share one incident per day, EOS worth 0.20€ must be staked permanently. For RAM, the costs cumulate with each executed action and are

[10] https://github.com/Dealer-Platform/.

[11] https://dingfest.ur.de/dealer/.

[12] https://kylin.bloks.io/account/eosdealeradm.

Table 2 Resources consumed by the EOS smart contract.

Action	CPU (stake)	NET (stake)	RAM (purchase)
Sharing	1.76 ms, 0.201€	0.256 kb, 0.0005€	0.755 kb, 0.088€
Verification	0.58 ms, 0.067€	0.120 kb, 0.0002€	0.000 kb, 0.000€
Purchase	1.07 ms, 0.065€	0.112 kb, 0.0002€	0.153 kb, 0.018€
Warning	0.74 ms, 0.084€	1.71 kb, 0.0034€	1.896 kb, 0.221€

EOS price: 2.00€, RAM price: 0.058 EOS/kb, CPU cost: 0.05 EOS/ms, NET cost: 0.001 EOS/kb

thus much higher. For this reason we now focus on RAM costs.

In the following we estimate the costs of the platform based on a real-world example. Therefore, we assume that the platform will be used for the reporting obligations of critical infrastructures in Germany. According to the Federal Office for Information Security (BSI), it is estimated that around 250 reports are carried out annually in 9 industry sectors [31]. The EOS RAM needed to store 250 incidents costs 22€ per year at the current conversion rate. The verifications do not cost any RAM since they only modify storage entries and don't add data.

We assume that participating companies are particularly interested in information from their sector (on average 28 reports per sector). According to the BSI, 1648 institutions in Germany are currently affected by the reporting obligation [31]. We thus estimate about 1648 * 28 = 46,144 purchases to be made in ongoing operations (823€). Additionally, we assume that authorities may issue warnings about once a month (3€). In summary, we expect a total RAM cost of 848€ to store all platform interactions occurring in one year. This is quite a feasible amount, considering that it covers more than a thousand institutions.

6.2 Performance

We evaluate the performance of our prototype with regard to user request latency (network latency and computation time) as well as storage requirements.

Request latency. We evaluate the performance of the server component locally on a machine with an i7-8550U CPU and 16GB RAM, running node.JS v10.17. The ping latency from the local machine to the go-ipfs v0.6.0 node running on a Raspberry Pi 3B is $\mu = 0.9\,\text{ms}, \sigma = 1.3\,\text{ms}$, while the ping latency to the EOS Kylin network node is $\mu = 12.0\,\text{ms}, \sigma = 0.9\,\text{ms}$ (100 pings). Request latency consists of transmission latency and server-side computation time. We measure the full request latency by timing `curl` requests with a bash script. Computation time is measured by tracking execution time of routes within the node.JS express instance for these requests. Therefore, transmission latency and client rendering speed are not included in measurements. However, these delays are negligible after the initial download of JS, CSS and image assets.

Table 3 Request Latency (RL) and Computation Time (CT) in ms

Action	RL μ	RL σ	CT μ	CT σ
W - Sharing	2185	327	2112	328
W - Verification	2453	137	2379	136
W - Purchase	2432	204	2355	202
W - Warning	1813	287	1738	286
R - Marketplace	1106	138	1041	138
R - Purchases	924	118	856	117
R - Report	65	6	1	1
R - My Incidents	1534	175	1464	168
R - Verification	1059	128	992	128
R - Dispute	930	132	863	132
R - Push Warning	404	34	338	34
R - Current Warnings	66	2	1	1
R - User Profiles	1050	95	983	95

As for the testing setup, there are 211 existing incidents in the smart contract, 108 of which are assigned to our test user for verification. We test each operation 100 times, including item upload, verification and purchase operations. We executed the tests in the order shown in Table 3.

The results show reasonable latencies of 1–2 s. Generally, the loading time for POST requests is higher, since the server first parses the request and then also prepares the webpage for the returned page. For example, to obtain the effective computation time of *W - Sharing*, the latency of *R - My Incidents* must be subtracted. In practice, the GET request latency can be removed by offering a POST-only endpoint for automated reporting.

In summary, the latencies should be appropriate for normal usage. Additional front-end optimizations such as pagination can further optimize page load times once a large number of incidents is stored on the platform.

Storage requirements. Storage needs of the EOS platform are covered in Sect. 6.1, so we now focus on off-chain storage of incident data and file keys on IPFS. We uploaded a small fixed-size (38 bytes) incident *m* times. During our initial experiments we found that storage consumption grew exponentially, but was significantly reduced after running IPFS garbage collection. Garbage collection deletes local copies for old versions of data no longer in use, for example for updated file key entries. Therefore, we run the garbage collector before taking each measurement. This ensures that storage consumption is measured correctly and does not

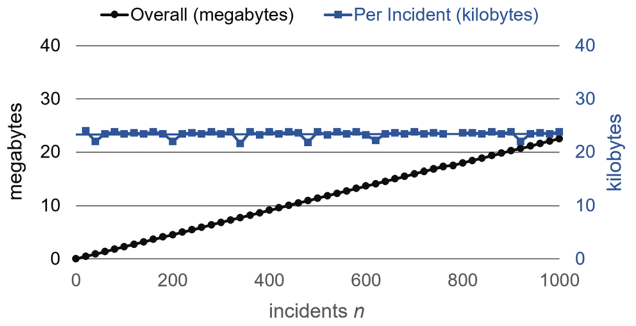


Fig. 9 IPFS storage consumption with increasing number of uploaded incidents

include duplicate entries from prior versions of the user's shared IPNS folder.

Figure 9 shows storage consumption of a single IPFS node with an increasing number of uploaded incidents. Storage consumption increases linearly with each uploaded incident. The total overhead is about 22 KB for each additional uploaded incident. 8 KB are added through preprocessing, which consists of the AES encryption of the incident, storing the ciphertext as base64 encoded string, and uploading file keys for the verifiers. The remaining 14 KB are due to IPFS internal data organization and tracking. We also tested a larger 1,032 KB incident and observed 380 KB total overhead. 340 KB are due to preprocessing, and another 40 KB are added by IPFS. The amount of overhead increases linearly with larger files, since the overhead originates largely from encryption and encoding (i.e. 2 MB incident ∼ 2.76 MB ciphertext).

6.3 Expert interviews

In addition to the evaluation of the transaction costs and the performance of the platform, we conducted several expert interviews to demonstrate the overall validity of our approach. The goal was to show that the intended implementation of the exchange platform offers real benefits for the industry. In this context, two specific questions were addressed. On the one hand, it was investigated to what extent the planned incentive system offers actual stimuli for companies to use it. On the other hand, it was investigated to what extent the integrity assurance measures can offer added value for companies in the reporting process. In addition to questions regarding the efficiency of the built-in incentive system, a further goal of the interviews was to get an impression of the usability of the platform as a whole in order to explore possible optimization opportunities. Accordingly, the interviews also covered the exchange process, the usability of the user interface and perceived security of the platform.

The interviewees are four security experts from different industry sectors. We conducted interviews with a Project Manager of a SME operating in the area of secure cloud services, with a security expert of a large corporate data center, with an academic researcher in the Field of Cyber Threat Intelligence as well as with a Security Consultant of a SME operating in the field of security consulting. We have designed the selection in such a way that all persons interviewed have extensive knowledge in the field of IT security and can therefore adequately assess the security benefits of the platform for their companies.

The expert interviews were designed according to the semi-structured approach of Lazar et al. [32] and are subdivided into the following 5 phases.

Phase 1—Introduction. At the beginning of the interviews, each interviewee was first asked about his or her knowledge as well as the extent of experience in the field of IT security and their knowledge of currently existing reporting obligations. The participants were also asked about their current position in the company and their budget responsibility in the area of IT security. The participants were also encouraged to indicate problems with the interview process at an early stage.

Phase 2—Incentive structure. The objective of the first thematic interview phase was to examine the benefits of financial incentives for the exchange process. To this end, the DEALER platform was first presented to the participants and the underlying idea was explained in detail. Subsequently, the participants were asked whether such an incentive system would be suitable for may be of interest to companies in principle. In this context, the participants were also asked what basic conditions would have to be fulfilled for their active participation. Finally, we asked if the participants can think of ways to abuse the system, or if they had concerns that they could be cheated by other participants.

Phase 3—Integrity features. The goal of the interview's second thematic phase was to assess the usefulness of the platform's integrity assurance and non-repudiation mechanisms. In order to achieve this, the participants were asked whether they had already been confronted with reporting obligations and whether their company is subject to reporting requirements. Subsequently, the participants were asked whether they saw a concrete benefit in the provision of integrity assurance and non-repudiation mechanisms and how this would be useful for them.

Phase 4—Platform usability. After evaluating the basic benefits of the concept in the previous interview phases, this phase deals with the actual implementation of the platform. The goal was to evaluate the usability and the benefit of the user interface as well as the exchange and reporting process. In order to obtain meaningful results, the participants were given access to the platform and only a brief explanation of the basic features of the platform was given. The participants were then given two tasks. First, they had to post a fictional security incident for sale on the platform and at the same time report it to an authority. In the second step, the participants were then to get an overview of the market situation and buy information about a security incident. In this phase we pay special attention to how well the participants understand the platform and how they handle it. In addition to mere observation, the participants are also asked about their experience using the platform.

Phase 5—Wrap-up. In this last phase of the interview, a summarizing discussion is conducted. Finally, the participants are asked again about their overall impression of the platform and whether they could imagine using such a concept in an operational context. In addition, the participants are asked about further points of criticism and possible suggestions for improvement.

6.4 Interview results

The interviews lasted between 30 and 80 min. Longer interviews were mainly due to extensive discussions with the participants about the platform and possible application scenarios of the approach. At the same time also the large interest of the participants in the presented beginning showed up. All in all, the interviews led to a whole range of additional insights regarding the incentive structure, integrity features and platform usability.

Incentive structure. In the first part of the interview, the participants were asked whether the proposed incentives were interesting, whether participation in the platform was conceivable for them and whether they had any concerns about using it. Generally, the paid exchange of incident information was met with great interest. However, it also became clear that the platform would essentially be used for the exchange of non-critical incidents. In this section of the interview, most of the interviewees placed a very high value on automation and low personnel costs. Specifically, platform participation was considered attractive if the platform would save time and personnel expenses. From the interviewees' point of view, this can be achieved especially by providing high-quality reports, as this can save a lot of time in the evaluation and use of information. It also became clear that for companies, the verifiers and quality assurance play the central role on the platform. To make quality assurance transparent, interviewees suggested introducing certification for the verifiers, which could, for example, be performed by authorities. An essential participation prerequisite was the availability of an API for automated incident processing, in order to increase efficiency and avoid expensive manual labor. Another central factor for the use of the platform is the legal security of its use. On the one hand, it was pointed out that incident reporting can only be carried out if legal certainty is established. Another criterion was the possible use of SLAs and general terms and conditions.

Integrity features. In the second part of the interview, the participants were asked about the mechanisms of integrity assurance and non-repudiation on the platform. Overall, the interviewees see a significant value benefit from these functionalities, which is particularly evident in the context of reporting obligations or insurance-related claims. They see clear potential for automation and reduction of bureaucracy. Especially the possibility to report on time, based on facts, irrevocably verifiable and tamper-proof were considered important features by the interviewees. It was emphasized that this feature is particularly interesting in cases where very high penalties are imposed for failure to report. However, the interviewees also pointed out various pitfalls and problems in implementing these features. It was shown that integrity assurance could also be carried out by public authorities and that for a real world implementation, various funded projects involving the authorities concerned would certainly be necessary.

Platform usability. Finally, the participants were asked about usability aspects of the platform. Overall, it can be stated that all participants understood the platform in principle and were able to use it completely after a short time. The interviews also consistently provided positive feedback on the proof of concept presented. The verifier user interface was particularly positively highlighted. At the same time, many suggestions for improvement were also made, especially with regard to productive use of the application. For example, it was suggested to integrate various additional information on legal implications of actions on the platform as well as the possibility to provide data sets with SLAs or terms and conditions. Furthermore, it was pointed out that in production use, extensive tools for the presentation of data set metadata are necessary in order to make clear and efficient purchase decisions. In this context, it was also suggested to introduce a subscription function for relevant sellers and to provide API access to increase the efficiency of the platform.

7 Discussion

In this section we discuss the results of this work. For this purpose, the previously defined requirements are reviewed in Sect. 7.1 and compared to the actual results achieved in the prototype. Subsequently, we discuss security concerns for the platform in Sect. 7.3.

7.1 Requirements

Reporting requirements. At the beginning of this work, Sect. 3 defined various requirements for a platform that simultaneously complies with legal requirements and offers incentives for the exchange of CTI information. Specifically, we defined *integrity* and *availability* of data as well as the *non-repudiation* of reports as target values for compliance with legal requirements. The decentralized blockchain technology used provides the necessary basic conditions to build a platform that is compliant with these requirements. One of the most important features of a blockchain is the assurance of data integrity using the decentralized ledger technology. Our solution assures *integrity* by including a hash of the data on-chain. Due to the EOS blockchain's immutability, this hash can be traced back to the original upload transaction and authenticated with the sender's signature.

Availability. The presented concept has, as previously stated, increased demands on the availability of the platform. In general, blockchains also offer a very high availability of the network nodes as pointed out by Weber et al. [2]. This results from the fact that the blockchain nodes are geographically distributed and run in a highly redundant manner. At the same time, one of the main restrictions of blockchain systems is that write-access is often limited, which may result in availability drawbacks. This is mainly the result of the low number of possible transactions per second of the considered blockchains Ethereum and Bitcoin. Since the EOS blockchain exceeds the possible transactions per second of these networks by orders of magnitude [29], restrictions of write availability are unlikely. It should also be emphasized that the EOS network is distributed over the entire globe,[13] which makes the availability of the network relatively independent of local events. This problem can be tackled in various ways with the EOS chain. On the one hand, it is possible to increase the available resources for the current project by increasing the share contributed. If you have even higher availability requirements, the block chain can also be set up with your own block producers. An example for such a split with own block producers is the Ultra/UOS[14] project. In this example, an own EOS blockchain was created to meet the high demands on throughput and availability within online games.

In the presented prototype, we store metadata of each reported security incident on the EOS blockchain in a publicly accessible manner. In order to establish a reference for *non-repudiation*, a timestamp is included in the incident metadata proving the report's existence. A reference to the reporting EOS wallet is included to link the report to the reporter's EOS wallet. The full incident data are stored on the IPFS DHT and replicated by the incident's seller and verifiers, ensuring *availability* of off-chain data through sufficient redundancy.

In addition to this, the prototype also provides the necessary tools to protect personal data within reports according to legislations such as the GDPR. To achieve this, the exchanged information is processed in an encrypted form on the platform. Each data flow is addressed to an explicit recipient and protected with the corresponding public key. This ensures that only the receiving authority can view the reported information. In the case of an exchange on the marketplace, the data are also encrypted and assigned to a buyer and verifier as specific recipients. However, since the data are transferred to different recipients, the mere assignment to the recipient is not sufficient for information and privacy protection. According to this, the offering company must decide here which data may be passed on to recipients. Both the interests of the company and the legal situation must be taken into account.

Incentives. As shown above, incentives represent are a necessary condition for an active exchange between the parties involved. In order to be able to implement such incentive procedures, we created marketplace within the platform for the mutual exchange of CTI information. Participants can offer their incident information at the marketplace in return for payment. This gives them the ability to compensate costs incurred in the detection and recording process and thus provides a financial incentive to participate in the platform. Another focus of the platform is to ensure sustainability of the implemented incentive structure. Verifiers ensure the data quality of the traded CTI information as well as functions that guarantee transactional fairness for both buyer and seller. Verifiers and sellers have an incentive to host incident data on IPFS since they profit from incident sales.

7.2 Comparison to other platforms

Overall, it can be concluded that the platform for the exchange of CTI information presented in this work offers several specific advantages over existing CTI sharing platforms. Traditional systems usually rely on trust in a Trusted Third Party (TTP) to implement the data protection goals. In contrast to this, the decentralized DEALER system guarantees these protection goals without the need for a specific trust relationship. The availability of the platform is distributed among different independent actors and no central actor is required for integrity proofs. Moreover, the implemented marketplace for the exchange of information is likewise not dependent on the trustworthiness of actors. Within the implemented smart contract, the sales process as well as the selection of verifiers is predefined and transparent for all participants.

7.3 Security

Free-riding verifiers. An important consideration is prevention of free-riding verifiers. Every verifier periodically receives free access to a randomly selected incident. As a result, verifiers must be punished if they do not perform verification as requested. If a verifier repeatedly fails to verify

[13] https://glass.cypherglass.com/map/main/top50.

[14] https://www.ultra.io.

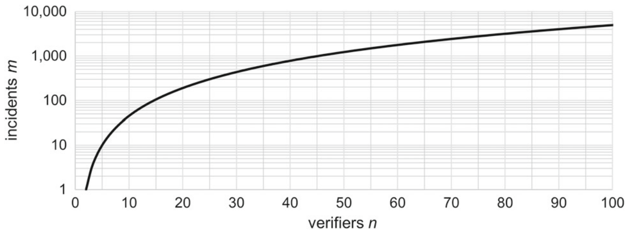


Fig. 10 A pair of verifiers is assigned to the same incident every m incidents, given n verifiers (y-axis is log scale)

assigned incidents in active status, other verifiers may start a multisignature vote for verifier removal. This encourages verifiers to only remain active when they intend to verify, to avoid losing their verifier status.

Content reselling. Reselling information is a common concern for data marketplaces [18]. As with all digital marketplaces, content reselling *outside the platform* cannot be fully prevented. Both buyers and verifiers may attempt to resell incidents they obtained through DEALER. In practice, this is discouraged by the difficulty of selling digital goods without trusted intermediaries [22]. The smart contract of the DEALER platform replaces intermediaries and provides certainty for buyers that they will receive the incident. Therefore, it is more difficult for illegal re-sellers to find buyers outside the platform without the market-making aspects of DEALER. To prevent reselling content *on DEALER itself*, the verifier system is in place to prevent it. The hash of the shared incident data is stored in the smart contract, allowing the identity of the original author to be clearly established through the timestamp and the signing public key of the transaction. Uploading duplicate incidents with the same hash is prevented by the smart contract, but resellers can slightly modify the incident to change its hash. Still, in the long run the similarity checks introduced in Sect. 4.3 reveal duplicates. If a duplicate is recognized, verifiers may submit a low rating. Similarly, buyers are likely to notice that they received a duplicate and rate the incident poorly, leading to a decreasing rating. This discourages potential buyers and lead to decreasing profits from reselling attempts.

Sybil attacks. Sybil attacks involve attackers being able to create new identities cheaply to manipulate the application. They can be mitigated by introducing nontrivial barriers to entry. On the DEALER platform, this threat mainly applies to sellers and verifiers. Sybil sellers could flood the platform with incidents to overwhelm verifiers. Sybil verifiers could dilute the quality assurance verifiers are supposed to provide. Therefore, as established in Sect. 4.1, both sellers and verifiers need to deposit cryptocurrency to create an account. Verifiers additionally need to prove their physical identity on registration. These measures present a significant obstacle for creating Sybil users.

Verifier collusion. The platform requires a minimum number of verifiers to ensure their assignment is sufficiently random to deter collusion. If assignment is not random, sellers may collude with verifiers to ensure incident verification. Alternatively, a pair of verifiers may collude during dispute resolution. The binomial coefficient determines the probability of assigning two verifiers to the same incident (n is the number of verifiers, and $k = 2$). As shown in Fig. 10, with 15 verifiers the probability is $< 1\%$, while with 50 verifiers it is $< 0.1\%$. Hereby we determine 15 verifiers as a safe minimum number of verifiers to safely operate the platform. Since verifiers may be temporarily inactive, a higher number is preferable in practice. With an expected amount of 250 reports annually (Sect. 6), each pair of verifiers shares only 2–3 incidents per year, which provides little incentive for collusion.

Even if an attacker is able to guess the pseudorandom number, the potential impact of such an attack is low. The background for this is the corresponding attacker model. At best, the attacker could assume the seller role and choose which verifiers are assigned to an uploaded incident. If these verifiers are controlled by the attacker, he may generate false ratings. By making fake incidents seem attractive, this could trick potential buyers into purchasing the fake incident. However, this would quickly become apparent, since buyers would rate such incidents low. If buyer ratings significantly diverge from verifier ratings, such incidents can be marked as potentially fraudulent in the DApp. Colluding sellers and verifiers are also registered by name on the platform and can be banned through majority consensus (Sect. 4.1).

Incident confidentiality. A compromise of the RSA or AES encryption scheme might compromise the confidentiality of the incidents stored on IPFS. Since IPFS data are stored on publicly available nodes, confidentiality is an inherent problem that can only be counteracted by encryption. This is especially the case as it is not possible to prevent an attacker from downloading the entire history for later decryption. However, we consider this scenario to be less problematic for various reasons. On the one hand, the procedures are state-of-the-art encryption technology and it can be assumed that they will be considered secure for many years to come, while the benefit of decrypted information on security incidents will decrease significantly over time. On the other hand, it can be assumed that the participants of the platform do not trade highly confidential data via the platform, since it is known that at least the validators must be given insight into the data and a large part of the data are available for sale on the platform anyway. Accordingly, the confidentiality of the data essentially relates to the protection of participation incentives. A possible compromise of the encryption schemes can additionally be counteracted by re-encrypting the data with a secure procedure, at least partially. If it is possible to decrypt incidents without purchase, participation for sellers and verifiers would be eliminated. Accordingly, such a change to newer procedures would be necessary at an early stage.

8 Conclusion

In this work we presented a fully decentralized model for sharing CTI. It is designed with legal and privacy requirements in mind and ensures sustainable sharing using cryptocurrency-based incentives. We implemented the DEALER platform based on the EOS blockchain and IPFS DHT and demonstrated its practical feasibility. On the platform, structured incident information is exchanged pseudonymously. Randomly selected verifiers use a set of objective CTI quality indicators to bootstrap incident reputation and help buyers select fitting incidents. Buyers and sellers are protected through dispute resolution mechanisms and exchange items based on cryptocurrency incentives.

Beyond our model and prototypical implementation, an integration with existing incident discovery, reporting and visualization systems is essential to the platform's practical viability. For example, the incident information currently available in plaintext could be enriched by a visualization system such as the one presented by Böhm et al. [33]. Based on such integrations, the platform can be deployed on the public EOS blockchain and tested with a larger number of users. In this scenario, price discovery mechanisms and their relationship to incident data quality can be analyzed. While our infrastructure is developed with privacy in mind, future work should ensure privacy and compliance with legal requirements (i.e. GDPR) in practice.

Acknowledgements We would like to thank our reviewers for their in-depth feedback and comments, which contributed significantly towards the final manuscript.

Funding Open Access funding enabled and organized by Projekt DEAL. This research was supported by the Federal Ministry of Education and Research, Germany, as part of the BMBF DINGfest Project (Grant No. 16KIS0501K) (https://dingfest.ur.de).

Compliance with ethical standards

Conflict of interest All authors declare that they have no conflict of interest.

Ethical approval This article does not contain any studies with human participants or animals performed by any of the authors.

References

1. Kannengießer, N., Lins, S., Dehling, T., Sunyaev, A.: What does not fit can be made to fit! trade-offs in distributed ledger technology designs. In: Bui, T. (ed.) 52nd Hawaii International Conference on System Sciences, HICSS 2019, Grand Wailea, Maui, Hawaii, USA, January 8–11, 2019, pp. 1–10, ScholarSpace (2019). http://hdl.handle.net/10125/60143
2. Weber, I., Gramoli, V., Ponomarev, A., Staples, M., Holz, R., Tran, A.B., Rimba, P.: On availability for blockchain-based systems. In: 36th IEEE Symposium on Reliable Distributed Systems, SRDS 2017, Hong Kong, Hong Kong, September 26–29, 2017, pp. 64–73 (2017). IEEE Computer Society. https://doi.org/10.1109/SRDS.2017.15
3. Schwartz, A., Shah, S.C., MacKenzie, M.H., Thomas, S., Potashnik, T.S., Law, B.: Automatic threat sharing: how companies can best ensure liability protection when sharing cyber threat information with other companies or organizations. Univ. Mich. J. Law Reform **50**, 887 (2016)
4. Laube, S., Böhme, R.: Mandatory security information sharing with authorities: implications on investments in internal controls. In: Ray, I., Sander, T., Yung, M. (eds.) Proceedings of the 2nd ACM Workshop on Information Sharing and Collaborative Security, WISCS 2015, Denver, Colorado, USA, October 12, 2015, ACM, pp. 31–42 (2015). https://doi.org/10.1145/2808128.2808132
5. Bauer, S., Fischer, D., Sauerwein, C., Latzel, S., Stelzer, D., Breu, R.: Towards an evaluation framework for threat intelligence sharing platforms. In: 53rd Hawaii International Conference on System Sciences, HICSS 2020, Maui, Hawaii, USA, January 7–10, 2020, pp. 1–10, ScholarSpace (2020). http://hdl.handle.net/10125/63978
6. IBM Corporation: X-Force Exchange. https://exchange.xforce.ibmcloud.com/
7. Facebook Corporation: Facebook Threat Exchange (2019). https://developers.facebook.com/programs/threatexchange/
8. Wagner, C., Dulaunoy, A., Iklody, A.: MISP—the design and implementation of a collaborative threat intelligence sharing platform. In: Proceedings of the 2016 ACM on Workshop on Information Sharing and Collaborative Security, pp. 49–56 (2016)
9. Luatics: OPENCTI. https://www.opencti.io/en/
10. Liu, C.Z., Zafar, H., Au, Y.A.: Rethinking FS-ISAC: an IT security information sharing network model for the financial services sector. CAIS **34**, 2 (2014)
11. Wagner, T.D., Mahbub, K., Palomar, E., Abdallah, A.E.: Cyber threat intelligence sharing: survey and research directions. Comput. Secur. **87**, 101589 (2019). https://doi.org/10.1016/j.cose.2019.101589
12. Serrano, O., Dandurand, L., Brown, S.: On the design of a cyber security data sharing system. In: Proceedings of the 2014 ACM Workshop on Information Sharing 38; Collaborative Security. ACM, New York, USA (2014), WISCS '14, pp. 61–69
13. Dandurand, L., Kaplan, A., Kácha, P., Kadobayashi, Y., Kompanek, A., Lima, T.: Standards and tools for exchange and processing of actionable information. November (2014)
14. Brown, S., Gommers, J., Serrano, O.: From cyber security information sharing to threat management. In: Proceedings of the 2nd ACM

Workshop on Information Sharing and Collaborative Security, pp. 43–49 (2015)
15. Mohaisen, A., Al-Ibrahim, O., Kamhoua, C., Kwiat, K., Njilla, L.: Rethinking information sharing for threat intelligence. In: HotWeb 2017—Proceedings of the 5th ACM/IEEE Workshop on Hot Topics in Web Systems and Technologies (2017)
16. Sauerwein, C., Sillaber, C., Mussmann, A., Breu, R.: Threat Intelligence Sharing Platforms : An Exploratory Study of Software Vendors and Research Perspectives, 13. Internationale Tagung Wirtschaftsinformatik, WI 2017, St. Gallen (2017)
17. Sillaber, C., Sauerwein, C., Mussmann, A., Breu, R.: Data quality challenges and future research directions in threat intelligence sharing practice. In: Proceedings of the 2016 ACM on Workshop on Information Sharing and Collaborative Security, pp. 65–70 (2016)
18. Alexopoulos, N., Vasilomanolakis, E., Roux, S.L., Rowe, S., Mühlhäuser, M.: TRIDEnT: Building Decentralized Incentives for Collaborative Security (2019). arxiv:1905.03571
19. Gong, S., Lee, C.: Blocis: blockchain-based cyber threat intelligence sharing framework for sybil-resistance. Electronics **9**, 521 (2020)
20. Homan, D., Shiel, I., Thorpe, C.: A new network model for cyber threat intelligence sharing using blockchain technology. In: 10th IFIP International Conference on New Technologies, Mobility and Security, NTMS 2019, Canary Islands, Spain, June 24–26, 2019, pp. 1–6. IEEE (2019). https://doi.org/10.1109/NTMS.2019.8763853
21. Shafagh, H., Burkhalter, L., Hithnawi, A., Duquennoy, S.: Towards blockchain-based auditable storage and sharing of iot data. In: Thuraisingham, B.M., Karame, G., Stavrou, A. (eds.) Proceedings of the 9th Cloud Computing Security Workshop, CCSW@CCS 2017, Dallas, TX, USA, November 3, 2017, pp. 45–50. ACM (2017). https://doi.org/10.1145/3140649.3140656
22. Wagner, E., Völker, A., Fuhrmann, F., Matzutt, R., Wehrle, K.: Dispute resolution for smart contract-based two-party protocols. In: IEEE International Conference on Blockchain and Cryptocurrency, ICBC 2019, Seoul, Korea (South), May 14–17, 2019, pp. 422–430. IEEE (2019). https://doi.org/10.1109/BLOC.2019.8751312
23. Bundestag, D.: Gesetz zur Erhöhung der Sicherheit informationstechnischer Systeme. Drucksache des Deutschen Bundestages **18**(31), 273 (2015)
24. Schlette, D., Böhm, F., Caselli, M., Pernul, G.: Measuring and visualizing cyber threat intelligence quality. Int. J. Inform. Secur. (2020). https://doi.org/10.1007/s10207-020-00490-y
25. Gascon, H., Grobauer, B., Schreck, T., Rist, L., Arp, D., Rieck, K.: Mining attributed graphs for threat intelligence. In: Proceedings of the Seventh ACM on Conference on Data and Application Security and Privacy (Association for Computing Machinery, New York, NY, USA, 2017), CODASPY ’17, pp. 15–22 (2017). https://doi.org/10.1145/3029806.3029811
26. Ayman, A., Aziz, A., Alipour, A., Laszka, A.: Smart Contract Development in Practice: Trends, Issues, and Discussions on Stack Overflow, CoRR abs/1905.0 (2019). arxiv:1905.08833
27. Bach, L.M., Mihaljevic, B., Zagar, M.: Comparative analysis of blockchain consensus algorithms. In: 41st International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO), pp. 1545–1550 (2018)
28. Zmudzinski, A.: ETH Transaction Fees Hit All-Time High Second Day in a Row (2020). https://cointelegraph.com/news/eth-transaction-fees-hit-all-time-high-second-day-in-a-row
29. Larimer, D.: EOSIO Dawn 3.0 Now Available (2018). https://medium.com/eosio/eosio-dawn-3-0-now-available-49a3b99242d7
30. Xu, X., Weber, I., Staples, M.: Architecture for Blockchain Applications. Springer, Berlin (2019)
31. Bundesamt fuer Sicherheit in der Informationstechnik. Die Lage der IT-Sicherheit (2019). https://www.bmi.bund.de/SharedDocs/downloads/DE/publikationen/themen/it-digitalpolitik/bsi-lagebericht-2019.pdf?__blob=publicationFile&v=4
32. Lazar, J., Feng, J.H., Hochheiser, H.: Research Methods in Human-Computer Interaction. Morgan Kaufmann, Burlington (2010)
33. Böhm, F., Menges, F., Pernul, G.: Graph-based visual analytics for cyber threat intelligence. Cybersecurity **1**(1), 16 (2018)

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.