Journal of
Marine Science and Engineering

MDPI

Article

Decentralized Documentation of Maritime Traffic Incidents to Support Conflict Resolution

Dennis Jankowski [1,*], Julius Möller [2], Hilko Wiards [1] and Axel Hahn [1,2]

[1] German Aerospace Center (DLR), Institute of Systems Engineering for Future Mobility, Escherweg 2, 26121 Oldenburg, Germany
[2] Department of Computing Science, Carl von Ossietzky University Oldenburg, Ammerländer Heerstraße 114-118, 26129 Oldenburg, Germany
* Correspondence: dennis.jankowski@dlr.de

Abstract: For the investigation of major traffic accidents, larger vessels are obliged to install a voyage data recorder (VDR). However, not every vessel is equipped with a VDR, and the readout is often a manual process that is costly. In addition, not only ship-related information can be relevant for reconstructing traffic accidents, but also information from other entities such as meteorological services or port operators. Moreover, another major challenge is that entities tend to trust only their records, and not those of others as these could be manipulated in favor of the particular recording entity (e.g., to disguise any damage caused). This paper presents an approach to documenting arbitrary data from different entities in a trustworthy, decentralized, and tamper-proof manner to support the conflict resolution process. For this purpose, all involved entities in a traffic situation can contribute to the documentation by persisting their available data. Since maritime stakeholders are equipped with various sensors, a diverse and meaningful data foundation can be aggregated. The data is then signed by a mutually agreed upon timestamping authority (TSA). In this way, everyone can cryptographically verify whether the data has been subsequently changed. This approach was successfully applied in practice by documenting a vessel's mooring maneuver.

Keywords: decentralized documentation; traffic incidents; conflict resolution; timestamping authority; maritime data

check for updates

Citation: Jankowski, D.; Möller, J.; Wiards, H.; Hahn, A. Decentralized Documentation of Maritime Traffic Incidents to Support Conflict Resolution. *J. Mar. Sci. Eng.* **2022**, *10*, 2011. https://doi.org/10.3390/jmse10122011

Academic Editors: Sebastian Feuerstack, Marko Perkovic and Lucjan Gucma

Received: 15 November 2022
Accepted: 14 December 2022
Published: 16 December 2022

1. Introduction

In the 1960s, the first regulations to make flight data recorders (black boxes) mandatory were adopted in several countries. Back then, primitive analog systems had been used to record primary flight data such as altitude, airspeed, heading, and acceleration [1]. Several decades later, more convenient digital flight data recorders are used that can precisely record over 100 variables simultaneously [2]. Compared to the aviation industry, continuous recording of data for marine casualty investigations was introduced several years later, in 1997, by IMO Resolution A.861(20), which was also included in the International Convention for the Safety of Life at Sea (SOLAS). According to SOLAS chapter V [3], passenger ships and ships of 3000 gross tonnages and upwards, which are constructed after a specific date, shall be fitted with a voyage data recorder (VDR) to assist in casualty investigations. Since then, VDRs have been used to analyze shipping accidents in official investigations; for example, in the investigation of the grounding and partial sinking of the Costa Concordia in 2012 [4].

However, VDRs are tightly secured devices, and their disassembly, and especially data extraction and analysis can be a complex and time-consuming manual process [4]. Also, according to the IMO casualty investigation code, investigations are regulated by the national authorities, which often only initiate official investigations for serious accidents that lead to the total loss of a vessel, loss of human lives, or significant environmental pollution (e.g., as regulated in the Maritime Safety Investigation Act of Germany or the

Merchant Shipping (Accident Reporting and Investigation) Regulations of the UK). This leads to several problems in investigating less severe accidents or critical traffic situations, especially from the perspective of data acquisition. Data from the vessel's sensor systems or surveillance data is one of the essential puzzle pieces for tracing the chain of events that lead to such a situation. As today's vessels often have sizes of 300 m in length or larger, even smaller accidents can lead to major financial losses for the involved participants in an accident. Additionally, law enforcement can become a problem as ships often sail under the flag of micro-states, and international enforcement of the law is extremely costly. In the best case, data acquisition must happen immediately after an incident. Still, in busy port areas, damages to port infrastructure, e.g., quay walls, may not always be recognized directly after the incident, and it may not be clear how the damage occurred. In these cases, there often exists multiple data sources that could contain data that is related to such an event. Nevertheless, if no official authority is coordinating the process of data acquisition and analysis, trust issues may arise between the involved parties regarding the authenticity of the data.

From this, it can be seen that there is a need to collect data in potentially critical maritime traffic situations that may lead to accidents while maintaining trust between all different involved parties in the case that the recorded data may be used in a later analysis of such a situation. Apart from that, the provision of a guarantee between the traffic participants that the data has been recorded at the point of the event, and was not modified, is also important to maintain trust. Therefore, in this paper, we propose a new method for recording vessel traffic data from multiple parties, with the intention of using this data in an incident investigation while guaranteeing non-repudiation of the recordings.

In Section 2, we initially provide relevant background information on current standards for Maritime Accident Investigations, Data Protection, and existing procedures for the secure storage of distributed data. On this basis, we subsequently derive requirements for a system for the documentation of critical traffic situations. Section 3 takes a closer look at the related work. We present existing techniques that address tamper-proof recording of data. In addition, blockchain-based approaches and concepts from the automotive and maritime sectors will also be considered. In Section 4, our approach to decentralized tamper-proof documentation of maritime traffic situations is presented in detail. Afterwards, in Section 5, the approach is tested and evaluated on a mooring maneuver in practice. Finally, Section 6 summarizes the obtained results and provides insight into limitations and future work.

2. Background

To derive requirements for a system for the documentation of maritime incidents, considering the above-mentioned challenges, it is first analyzed how accident investigations are currently carried out. Then concerns are discussed that may arise for different parties regarding a self-organized solution for investigating more minor incidents. Considering these approaches, state-of-the-art methods to securely document incident data are analyzed. Finally, we derive requirements for the conceptualization of a system for the documentation of these incidents.

2.1. Maritime Accident Investigations

Accidents in the maritime industry have been present since the invention of shipping. Countries typically have their own regulations and processes on when and how accident investigations should be carried out, adapting codes and regulations from the IMO (cf. [5]). Results from the analyses of accidents and incidents are then published in reports. As a basis for relevant accident data, we analyzed accident investigation reports from the authorities of Germany [6] and the United Kingdom [7]. The following information is normally included in investigation reports dealing with accidents that are related to traffic (this does not include accidents on board a single ship):

- Data related to the involved vessels (name, type, flag state, size, special features, etc.).
- Planned routes of the involved vessels.

- Type, date, and place of the accident.
- A detailed description of the sequence of events that led to the accident, often including textual descriptions of personnel behavior, navigation decisions, vessel movements, pictures of the involved ships or damage caused, and other information that is specific to the type of accident.
- If available—data dumps from the involved vessels.
- Meta-information on how the investigation was carried out.
- Analysis of the causes of the accident.
- Summary and recommendations.

Especially the description of the sequence of events that led to the accident is a critical part of a report, as it is the main basis for analyzing the causes of the accident. It often includes eyewitness testimony or may be based on data recorded on board the vessel or other maritime surveillance equipment (e.g., VTS centers).

Hence, in the discussed scenario of a smaller incident that is not being investigated by a state authority, the involved parties should aim at preserving data that can represent navigation decisions and vessel movements at the time of the accident as a minimum objective reference for its cause. Other information (such as the data related to the vessels, meta-information on the investigation, and further analysis) can still be derived at a later stage of the incident analysis.

2.2. Data Protection and Sovereignty

In the case of an accident investigation, authorities have the executive power to confiscate VDRs or request data from external surveillance systems. However, this often happens a certain amount of time after the event. For more minor incidents without an official investigation, the procedure of data recording and preservation must be organized differently. Also, at the time of the incident itself, it may not be apparent to all parties that an incident is currently happening. Therefore, it is essential that data in potentially critical situations is continuously recorded and available for later analysis. In the case of the quay wall, it can also occur that the damage is only noticed in a general inspection or by other incoming vessels, and it is not even clear which parties were involved in the incident. In these cases, it is not an option of the port owner to force all ship owners to share their VDR data; thus, other means are needed for more efficient data sharing. On the other hand, it might also not be a possibility for all the involved parties to continuously exchange data with each other just in case something happens. Depending on the shared data, this may also violate data protection regulations (such as the GDPR in the EU, cf. [8]), for example, if they include personal data. Another aspect to consider is the value of the data to a company since, for example, business secrets could be revealed if data about a ship is constantly shared (preferred routes, etc.). In general, the idea of being able to meaningfully control and govern one's own data is a recent trend and is generally referred to as data sovereignty [9]. These factors lead to the conclusion that the recorded incident data cannot be stored centrally in a cloud environment (or similar) but must remain with the involved parties until a conflict arises and needs to be resolved. Even then, the participants should have full decision-making power over their data.

Finally, keeping local records of data with proof that it has not been modified and was recorded at the time of an incident only makes sense if it can be ensured that the recorded data evidence will stand up in court. Only in this way can a serious conflict resolution be performed between the involved parties and other relevant stakeholders (such as insurance companies). Similar to an analog paper document, a digital datum acquires its validity through a digital signature. In the setup of a Public-Key Infrastructure, users possess a public- and a private-key, which can be used to encrypt or sign data. The public key is also typically signed by a certification authority (CA) in the form of a certificate, thus binding it to the physical identity of a user. In order for the user to be able to use the key pair to sign its data in a trustworthy way, a certification authority must be used to certify the public key and apply special verification procedures (in accordance with applicable law) to ensure that the user's identity is correct. Furthermore, key-holders must take measures to prevent their private key from being exposed to other unauthorized parties in such a way that it can be assumed that data was actually signed by the key-holder and the key was not compromised [10].

2.3. Approaches to Immutable and Decentralized Data Storage

Establishing trust between different parties that have no direct trust relationship but aim to exchange information with each other is not a new problem in communications engineering. Typical setups for the exchange of digital information start with the authentication of the involved parties with the help of an identity provider. Therefore, a relying party can identify other participants by trusting the identity provider [11]. However, authentication of a party does not imply full trust in the actions of that party. In the maritime industry, it may be easy to identify a vessel or infrastructure with the help of its registration data at the IMO via an MMSI or even digitally via Identity Providers that connect vessel identities to digital certificates such as the Maritime Connectivity Platform (MCP) [12]. However, when it comes to trusting the correctness of a data asset that was recorded by a specific entity, additional frameworks need to be introduced. As discussed in Section 1, for accident investigation data, it is crucial to ensure that the data were recorded at the time of the accident and not subsequently modified. In similar situations, where the different participants do not trust a common central entity that could guarantee the correctness of the exchanged data, blockchain technology has often been applied.

A blockchain is a specific type of distributed database that only allows the addition of new information if a consensus in the network of involved participants is reached. Furthermore, data is only being added to the blockchain in the form of atomic transactions, which are organized in a chained data structure referencing its respective predecessor. After a transaction is issued, it is impossible to modify it, which is referred to as immutability [13].

However, the inflationary usage of blockchain technology in all possible use cases has been questioned more and more recently [14,15]. Also, there are other less complex solutions to the problem of providing proof that data existed at a certain point in time and was not modified. For instance, a possibility of this kind in regard to timestamps is represented by so-called timestamping authorities [16,17]. A TSA is a service that establishes that some data existed at the specified time based on reliable time sources and cryptographic signatures. Depending on the application, such a TSA can be operated in-house or externally, either as a commercial service or as a service for third parties.

The biggest problem in using these services is that all parties must trust the selected TSA [18]. Suppose one party can obtain the private key or manipulate the TSA's internal system time. In that case, the entire chain of evidence is invalidated since the data can then be signed with different timestamps. To circumvent this, an approach based on "distributed trust" is presented in [19] by randomly selecting subsets from a set of TSAs via a pseudorandom number generator (PRNG). However, even in this case, users of TSAs must always rely on the availability and proper functioning of the TSA they are using to sign (similar to trusting a certificate authority in a PKI).

2.4. Requirements for the Secure Documentation of Critical Maritime Traffic Situations

From the above considerations, the following requirements were derived for the development of a system to securely document critical maritime traffic situations for usage in smaller incident investigations and insurance cases:

(R1) The system must be able to record arbitrary sources of data.
(R2) The system must provide evidence that the data was recorded during the incident and was not modified.
(R3) The system must support decentralized usage such that every involved party can protect the sovereignty of their data.

(R4) It must be possible for a participant to dynamically join a network with other participants to agree on the common data recording of a potentially critical situation (e.g., berthing of a ship or an evasive maneuver).

(R5) If not all the involved participants are equipped with the system, a fallback mechanism must exist such that data is still being recorded by the remaining parties with a minimal loss of trust.

3. Related Work

In the past decade, the field of data security has seen a strong increase in interest with a particular focus on methods to prevent data tampering. This chapter presents some of the most promising approaches to tamper-proof data recording. First, we examine blockchain-based approaches, which are often associated with cryptocurrencies, but their possible applications extend far beyond that domain. Furthermore, we explore other techniques for tamper-proof data recording based on cryptographic techniques, and finally close with approaches from the automotive and maritime domains.

3.1. Blockchain-Based Approaches

Various works have already considered the use of blockchain-based approaches for documenting events. These can be roughly divided into approaches that use public blockchains (e.g., Bitcoin, Ethereum) and those that use permissioned blockchains. The former are particularly attractive for their transparency, as everybody can view the blockchain data, while permissioned blockchains are usually used in private settings.

Approaches based on the public Bitcoin blockchain were pursued in [20,21]. Due to the transaction costs associated with public blockchains, these approaches are based on writing summary hashes or anchor points of KSIs (Keyless Signatures Infrastructure) [22] to the blockchain at regular intervals. The data itself is not published. In this way, it is possible to verify whether data has been tampered with by inspecting the summary hashes. The drawback of these approaches is that it is not possible to directly verify the data that was written to the blockchain. Instead, the blockchain user needs to trust the KSI infrastructure that was used to write the summary hashes to the blockchain. Furthermore, it is also apparent that the benefit of the Bitcoin blockchain lies primarily in the permanent publication of the data, as the timestamps themselves are not trustworthy (see [23,24]) and require additional external time verification such as KSI [22].

To achieve better scalability and a higher data rate, logging infrastructures based on permissioned blockchain were presented in [25,26] and achieved data rates of 100–3500 transactions per second. In [27], an approach is described where cars serve as live witnesses to situations and decisions as participants in a blockchain. Based on vehicle-to-vehicle and vehicle-to-infrastructure communication, a shared truth is formed in this way. However, trust requires that as many independent parties as possible are part of the permissioned blockchain network. Depending on the use case with alternating or short-term stakeholders, this could result in a large overhead and is the reason why permissioned blockchain networks are usually based on a network of companies and organizations [27].

3.2. Tamper-Proof Data Recording

In addition to the approaches presented based on blockchains, there are other ways to check data records based on cryptographic properties, to a certain extent, for subsequent changes. For example, the documents can be electronically signed, stored in a data structure, and subsequently checked for validity. In [28], a log server is presented for this purpose, which provides a structure for tamper-evident data logging for a larger number of clients based on Merkle trees by feeding back smaller commitments to the clients. In [29], an approach based on hash-chains is presented. This approach enables logs to be protected on compromised machines without the need to publish anchors of the log by linking each entry with the previous one based on one-way hash functions. Therefore, a modification of the data would render the complete chain compromised. Several approaches based on the trusted hardware features of newer processors are shown in [30,31]. These works use the Trusted Platform Module (TPM) to ensure a tamper-proof log. The TPM creates a chain of trust from the bootloader to the operating system and the log system. The work in [31] uses TPM 2.0 to secure the log even between power cycles.

3.3. Further Approaches

In the automotive sector, the Event Data Recorder (EDR) is currently used to record incident sensor data [32]. Triggered by a crash, these record low-bandwidth data including car and engine speeds, brake status, and accelerations [33]. The data is stored in non-volatile memory within the control unit. These systems are permanently integrated into cars and operate on proprietary interfaces [34], but are themselves probably not adequately protected against manipulation of the collected sensor values, and measures against subsequent overwriting are currently not described [35].

Some works deal with fishery logbooks, i.e., logbooks in which the catches of fishermen can be documented. However, there is no focus on the manipulation of the data, as these are concerned with the architecture of the data exchange [36] or the analysis of these data [37].

A low-threshold approach to analyzing maritime sailing behavior and traffic situations is to look at AIS data. This data is publicly receivable and can be purchased afterwards from service providers but lacks a fine-grained resolution. Vessel Traffic Services [38] use this data (along with other sensors such as radar, cameras, etc.) to provide live assistance and instructions [39]. Since this data is sent publicly at regular intervals, the data received in this way can be used as additional anchor points for a possible conflict resolution.

In summary, the maritime domain still lacks a method for trusted, automatic data recording of multiple sensors without compromising data sovereignty and privacy. The blockchain-based approaches shown are promising but still miss some practicality regarding transaction speed, scaling, and compartmentalization. Some of these shortcomings are solved by using classical approaches based on cryptographic signatures, but these lack the distributed characteristics of the blockchain. As a result, we are not aware of any approach that sufficiently addresses the properties of non-repudiation, decentralization, and performance simultaneously.

4. Concept

As already described in the requirements from Section 2.4, it is crucial for trustworthy documentation that it can be verified that the recorded data has not been changed intentionally or unintentionally. This assurance can be provided by timestamping authorities which is fast and efficient compared to alternative approaches such as blockchain technologies for the certification of data [13,40]. Therefore, this paper presents a TSA-based concept to document critical traffic incidents in a decentralized scenario.

The certification process of TSAs is analogous to the certification processes of a PKI. However, in this case, an authority is used to certify the existence of a date at a specific time instead of the identity of an entity (c.f. Figure 1).

For this purpose, the requestor sends its data that should be certified in hashed form to the TSA. This is done to reduce the payload and to avoid exposing the actual data to the TSA. The TSA adds the current timestamp to the transmitted data hash and hashes the data again. The TSA then sends the signed document back to the requester. The requester needs to persist the original data and the TSA's digital signature. If the requester wants to prove that their data has existed at a certain point in time and has not been changed, they can verify this cryptographically in two steps (c.f. Figure 2).

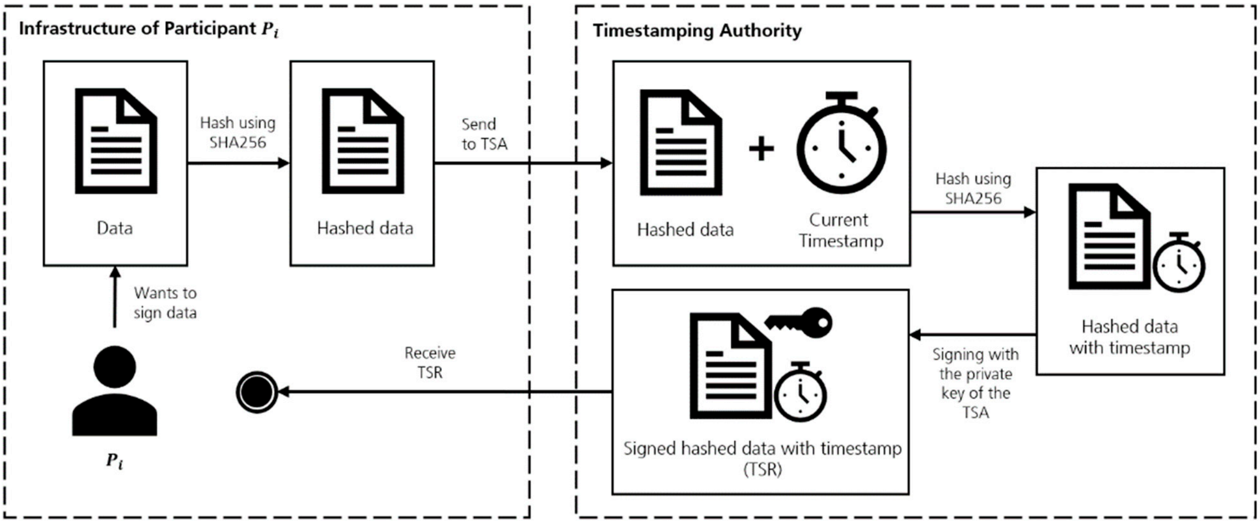


Figure 1. Timestamping process between a requester and a TSA [41].

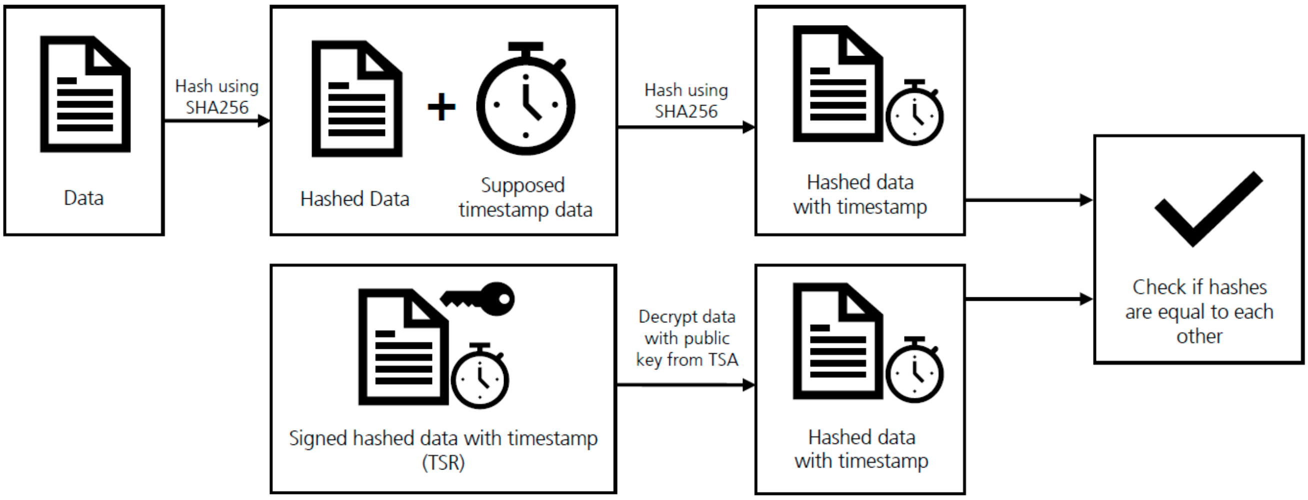

Data
Hash using SHA256
Hashed Data
+
Supposed timestamp data
Hash using SHA256
Hashed data with timestamp
Check if hashes are equal to each other
Signed hashed data with timestamp (TSR)
Decrypt data with public key from TSA
Hashed data with timestamp

Figure 2. Verification process of timestamped data [41].

In the first step, a hash is created based on the data using the same hash method that was applied by the requester during the timestamping process. Subsequently, the exact timestamp at which the TSA has signed the data is added to this hash. This document is hashed again with the hash procedure used by the certifying TSA. In the second step, the cryptographic signature of the document is decrypted using the TSA's public key. The hash that was derived in this process is then compared with the hash value from step 1. If the two values match, the verification was successful, and it can be assumed that the data already existed at the specified time. If the two hash values differ, there can be many reasons for this. For example, it is possible that the data was changed or that it was signed at a different time. Regardless of the reason, the verification failed in this case.

In this way, it is possible to verify whether data was available at a certain point in time by utilizing a TSA. However, the main issue with using an authority to establish trust is that all the trust depends on the authority itself. Thus, the signature of a TSA has only some value if the parties, to whom something should be proven, also trust this single TSA. Consequently, if an architecture is built based on only one authority, this directly leads to a problem if only one of the involved parties does not trust this central TSA. Especially in the global maritime domain, in which many different stakeholders from various fields and nations meet, an agreement on a worldwide central authority is not feasible. The utilized architecture must therefore ideally support the use of arbitrary TSAs, so that the parties involved can individually agree on a single or a set of trusted TSAs in the specific situation. In this way, trust is maximized as each participant can communicate their trusted TSAs instead of choosing from a predefined set of authorities. Since the participants in a traffic event usually do not know each other beforehand, the agreement on the set of jointly trusted TSAs must happen before a critical traffic situation.

Figure 3 shows the proposed architecture that enables the involved stakeholders to document their data in a trustworthy and decentralized scenario without using a central instance for coordinating the agreement on a TSA.

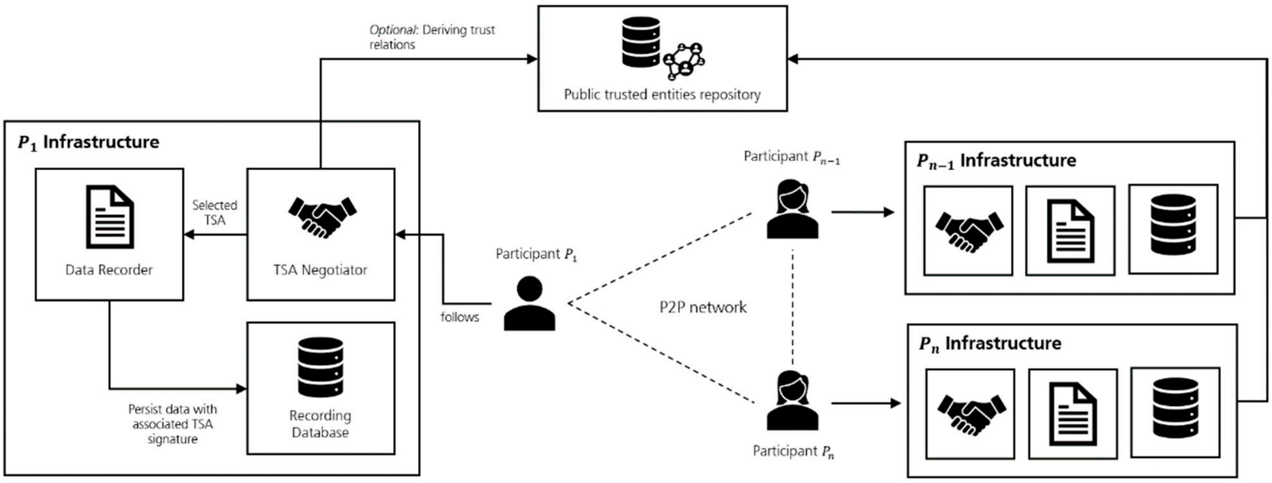


Figure 3. Approach for Decentralized Documentation of Maritime Traffic Incidents.

The entire architecture consists of five components:

- P2P-Network—Basis for communication between the involved participants of an incident that should be recorded.
- TSA Negotiator—Responsible for the derivation of a common set of trusted TSAs between all participants.
- Data Recorder—Coordinates the recording and signing of data streams by a trusted TSA derived from the TSA Negotiator.
- Recording Database—Persists the original data with the associated signature from the used TSA locally.
- Public Trusted Entities Repository—Can optionally be used by participants for publishing which other entities (e.g., organizations, or companies) and TSAs they trust. The repository is centrally managed and supports the participants in deriving a larger set of trusted TSAs based on their trusted relationships.

Each participant deploys the *TSA Negotiator*, the *Data Recorder,* and *the Recording Database* on its local infrastructure so that it has complete control over these components. This reduces the amount of data that leaves the infrastructure of the participant, which in turn protects the participants' sovereignty and minimizes the system's required bandwidth. In the following, the overall process of a to-be-documented situation is described in more detail:

(1) Initiate the documentation: The entire documentation process is initiated by activating a predefined trigger. This trigger can be chosen according to the use case. For instance, the trigger can be the entry of a ship into a certain geographical area (such as the port) or an under- or exceeded speed. Thus, the trigger defines whether an incident is a document worthy incident or not. The definition of a trigger is the responsibility of the participant with the primary interest in documenting a critical incident, such as the port operator, when a vessel enters the harbor.

(2) Establish a communication channel: The basis for communication between the participants is a P2P-Network that is set up dynamically between the participants in

the event of a potential incident. By using a P2P-Network, a central communication channel can be avoided so that the participants can communicate with each other in a completely decentralized way. The network is used to coordinate the communication between the participants to agree on a common set of trusted TSAs that can then be used to sign the record. Since the negotiation already takes place during the initiation of a critical traffic situation, the involved parties must have already established a P2P network between themselves.

(3) Derive a common trusted set of TSAs: After establishing a communication channel, the participants need to find a common trusted set of timestamping authorities with which they will sign their data recordings. For negotiation, each participant sends its own trusted TSAs to every other participant over the P2P network so that each participant knows the trusted TSAs from every other participant. The derivation of the trusted TSAs is performed by a deterministic negotiation protocol of the TSA Negotiator.

(4) Record the data: The Data Recorder is responsible for recording and signing the data that should be documented. Therefore, the Data Recorder periodically divides the data stream from a participant into discrete chunks and sends the hashed chunks to the TSA derived by the TSA Negotiator. Afterwards, the TSA's original data and signature are stored locally in the Recording Database.

In the following, we will discuss in detail how the *P2P Network*, the *TSA Negotiator*, the *Data Recorder,* and the *Public Trusted Entities Repository* work.

4.1. P2P Network

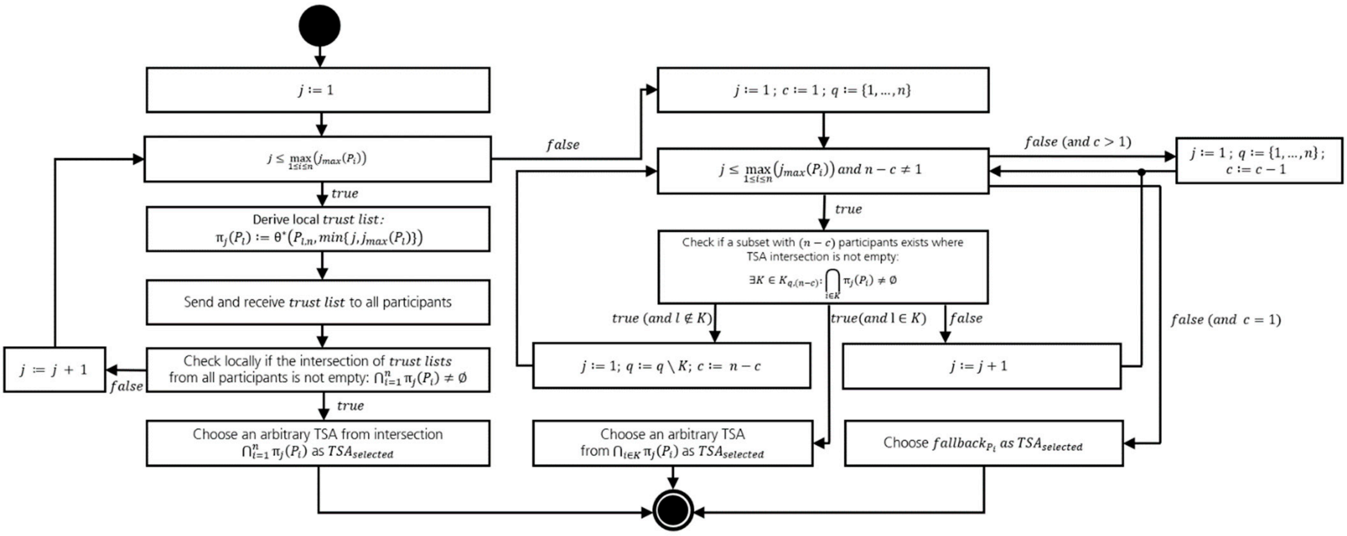
To agree on a common set of TSAs, a communication channel is needed through which all participants can share their trusted TSAs. Since there might not be a central instance for the communication that is trusted by all, a P2P network is used allowing the participants to interact directly with each other without an intermediary. As already described, the P2P network is established dynamically when a critical traffic incident occurs between the involved parties as soon as the defined trigger is released. To be able to set up such a network dynamically, all participants' IPs must be known by everyone. Depending on the application, this may not be the case, so the IP-addresses must first be exchanged with each other. In practice, this can be done in every conceivable way. In the following, we have outlined an exemplary possibility in more detail.

Let P be the set of all participants, so one of the participants $a \in P$ publishes its IP address publicly so that the other participants from the set $P \setminus a$ can obtain it. Subsequently, each participant from $P \setminus a$ sends its own IP-address to participant a. In this way, participant a knows the IP addresses of all participants, enabling him to forward the addresses of all participants to everyone. In this way, each participant receives the IP-addresses of all other participants, which means that each participant is now able to communicate with one another.

Usually, the initiating participant should be responsible for publishing the primary IP address. The IP address can be transmitted via various channels, via a website or AIS messages. It is also conceivable to publish the address as a service in a public Maritime Service Registry (MSR) of the MCP. The service registry acts as a directory where services of maritime stakeholders can be found by entering various parameters such as keywords or geographical regions. For example, a port operator could publish its IP address as a service in the Maritime Service Registry. Vessels wishing to enter the port would then have the opportunity to search the MSR for a corresponding service, in order to find the stored IP-address that is required for the initial contact. After all, the parties have already contacted the port operator; therefore, the latter can set up the P2P network as described above. However, in some cases, it might not be possible for every participant to directly expose an IP-port for communication (e.g., due to some ship-related IT security regulations). Also, as IP connectivity is not always available, it is conceivable to use other technologies (such as VDES) as a backup solution for communication. Another possible solution is an architecture, as proposed by IEC 63173-2 [42], where a service at the shore-side exposes a publicly available interface via IP and the communication to the ship ("last mile") is realized with a more secure communication channel.

4.2. TSA Negotiator

As mentioned in Section 4, the main task of the TSA Negotiator is to ensure that the participants can agree on a common set of timestamping authorities when entering a potentially critical situation. This is realized with a negotiation protocol as is shown in Figure 4. The negotiation protocol is a deterministic protocol so that each participant can run the protocol locally after gathering all information and the results of all participants will be consistent with each other. Therefore, a central instance that coordinates the negotiation can also be avoided here.



$j := 1$
$j \leq \max_{1\leq i\leq n}(j_{max}(P_i))$
true
Derive local *trust list*: $\pi_j(P_l) := \theta^*(P_{l,n}, min\{j, j_{max}(P_l)\})$
Send and receive *trust list* to all participants
Check locally if the intersection of *trust lists* from all participants is not empty: $\cap_{i=1}^{n}\pi_j(P_i) \neq \emptyset$
false
$j := j + 1$
true
Choose an arbitrary TSA from intersection $\cap_{i=1}^{n}\pi_j(P_i)$ as $TSA_{selected}$
false
$j := 1$; $c := 1$; $q := \{1, \ldots, n\}$
$j \leq \max_{1\leq i\leq n}(j_{max}(P_i))$ and $n - c \neq 1$
false (and $c > 1$)
$j := 1$; $q := \{1, \ldots, n\}$; $c := c - 1$
true
Check if a subset with $(n - c)$ participants exists where TSA intersection is not empty: $\exists K \in K_{q,(n-c)}: \bigcap_{i\in K}\pi_j(P_i) \neq \emptyset$
true (and $l \notin K$)
true(and $l \in K$)
false
false (and $c = 1$)
$j := 1$; $q := q \setminus K$; $c := n - c$
$j := j + 1$
Choose an arbitrary TSA from $\cap_{i\in K}\pi_j(P_i)$ as $TSA_{selected}$
Choose $fallback_{P_l}$ as $TSA_{selected}$

Figure 4. Negotiation Protocol for the agreement on a trusted TSA between different participants.

In the following, we refer to the participants as P_i with $i = \{1, \ldots, n\}$. Furthermore, the local view of a participant is denoted by using P_l for self-references. Initially, each P_i provides a list of trusted TSAs $\theta(P_i)$ and a list of trusted organizations $\phi(P_i)$.

It is assumed that each participant P_i has an internal prioritization of TSAs and wants to maximize the trust by choosing the TSA ranked highest in its local priority list. To ensure that the TSAs with the highest trust from all of the participants are found, the potential TSAs are exchanged in several iterations such as j (a higher number of iterations, j means less trust as the lists of possible TSAs are extended with lower prioritized TSAs by each participant). We call a set of TSAs that is trusted by a participant P_i the *trust list* of P_i. Theoretically, any participant can arbitrarily choose trust lists in every iteration. However, we propose the following method for determining the trust list of a participant P_l in iteration j $\left(\pi_j(P_l)\right)$ of the protocol:

$$\pi_j(P_l) := \theta^*(P_l, \min\{j,\ j_{max}(P_l)\})$$

with

$$\theta^*(P_i, n) := \theta\left(\left\{P_k | P_k \in \cup_{i=1}^{n}\phi^{(l-1)}(P_i)\right\}\right)$$

With P being a set of participants, $\theta(P)$ is defined as $\theta(P) := \cup_{p\in P}\theta(p)$ and $\phi^l(P_i)$ is the repeated application of ϕ to get the trusted organizations of P_i with ϕ^1, the union of all trusted organizations of each $P_k \in \phi^1$ with ϕ^2, etc. Note that $\phi^0 = \varnothing$.

For example, in the first iteration $j = 1$, P_l will only add the TSAs it trusts directly to its trust list $\pi_1(P_l)$. For $j = 2$, more TSAs will be added by also including the trusted TSAs of organizations or companies that the participant P_l trusts (given by ϕ^1, e.g., by using the public trusted entities repository, see Section 4.4). Subsequently, this can be continued by also adding the TSAs of the trusted organizations of the organizations trusted by $P_l(j = 3)$. These transitive relations can thus be continued for the number of iterations in the protocol. In this way, in each iteration, the probability that a commonly trusted TSA is found is increased, as $\pi_j(P_i) \subseteq \pi_{j+1}(P_i)$. At the same time, the individual participant's trust in the added TSAs decreases. As this process will not carry on indefinitely, each participant P_i defines locally in how many iterations $j_{max}(P_i)$ the own trust list is extended, so that the number of total iterations in the protocol is defined as: $\max_{1<i<n}(j_{max}(P_i))$.

Now for each iteration, each participant derives its trust list for the j-th iteration $\pi_j(P_l)$ and broadcasts it to all participants over the P2P network. In this way, each participant receives $n - 1$ trust lists in total. Then, it is checked locally (per iteration) if there is an intersection of all trust lists:

$$\bigcap_{i=1}^{n} \pi_j(P_i) \overset{!}{\neq} \varnothing$$

If an intersection is found, any TSA from the identified intersection can be used by the participants as $TSA_{selected}$ and utilized to sign the recorded data. If after the last iteration (i.e., $j = \max_{1<i<n}(j_{max}(P_i)))$ still no common TSA can be found, it is not possible to identify a TSA that all participants trust. Therefore, the second stage of the protocol is entered and is aimed at finding the largest possible subsets of participants who can agree on a set of TSAs in such a way that:

$$\exists K \in K_{q,k} : \bigcap_{i \in K} \pi_j(P_i) \neq \varnothing$$

Here, we define $K_{q,k} := \{M \subseteq P(q) || M| = k\}$ with q being an index set for the considered participants in a single round of the second stage of the protocol. Hence, $K_{q,k}$ is the set of all subsets of q with size k. We introduce c as a pruning factor for the participant sets, such that the number of currently considered participants for finding a common TSA is $(n - c)$. Initially, we set $c = 1$ and $q = \{1, \ldots, n\}$, so that all subsets are considered that are missing only a single participant. For each set K in $K_{q,\ (n-c)}$ it is then checked whether a common set of TSAs exists for which $\cap_{i \in K} \pi_j(P_i) \neq \varnothing$ is true (similar to the first stage of the protocol). Generally, the search for a common TSA set proceeds identically to the search with all participants so that for each pruning factor, the trust lists are also successively extended in j iterations. If a set is found, then an arbitrary TSA from the set $\cap_{i \in K} \pi_j(P_i)$ is selected as $TSA_{selected}$ for signing the data for the participants from K. Since the trusted TSAs of the individual participants do not change during the negotiation, the trust lists do not need to be transmitted again over the P2P network. If a common TSA cannot be found for any K after j iterations, c is increased by 1 so that in the next iteration, all $K \subseteq q$ with one participant less are taken into account.

However, since a valid K is a proper subset of the index set of participants q, even after finding a TSA there will be a set of participants $q \backslash K$ for which no TSA has been found yet. Given that the set $q \backslash K$ refers to at least 2 participants if $c > 1$, it should still be aimed to create a common trust basis between these participants. This is done in the same way as for the first subset with $j := 1;\ q := q \backslash K;\ c := n - c$. This process is repeated until $n - c = 1$, so that no more subsets with at least 2 participants can be formed. If this case occurs, a common TSA cannot be found based on the trust lists of the respective participants under any circumstances. Here, each of these participants P_i should use self-selected TSA ($fallback_{P_i}$) as $TSA_{selected}$ to sign the records and to establish at least a minimum of trust and security in the documentation.

Note that the search for $\cap_{i \in K} \pi_j(P_i)$ for a subset K of q assumes that it is better to find a common TSA at least among the largest possible subset of participants, instead of finding no TSA at all. In this way, at least the largest possible trust is established between the participants. Also, if at any step there are multiple subsets of q of equal size for which $\bigcap_{i \in K} \pi_j(P_i) \neq \varnothing$ holds, a choice must be made for one subset K deterministically since the participants in the two sets might overlap. To deterministically select one of the sets, they are hashed and then sorted alphabetically. The alphabetically first set is then defined as the selected set K.

4.3. Data Recorder

After each participant has found a $TSA_{selected}$, the local documentation of the incident can begin. The Data Recorder is responsible for managing the data's recording, signing and persistence. Figure 5 shows the functionality of the Data Recorder.

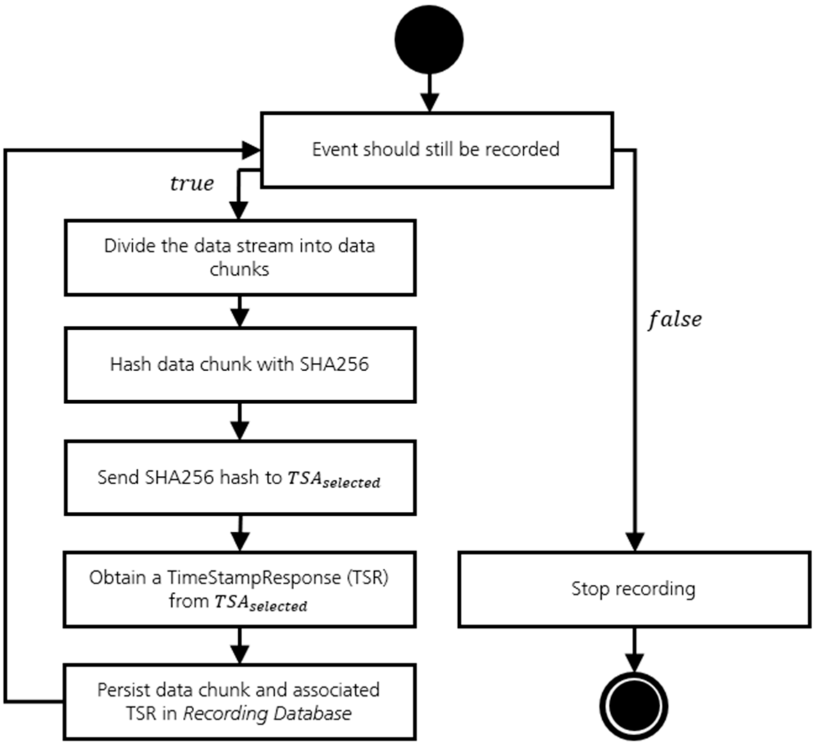

Event should still be recorded
true
Divide the data stream into data chunks
Hash data chunk with SHA256
Send SHA256 hash to $TSA_{selected}$
Obtain a TimeStampResponse (TSR) from $TSA_{selected}$
Persist data chunk and associated TSR in *Recording Database*
false
Stop recording

Figure 5. Procedure for local data recording and signing.

Initially, participants decide for themselves whether they want to continue the current recording of the event. Every participant can choose when to end the recording independently from the other participants since the end of an incident can be interpreted differently. As the recorded data is usually live data, the data must first be split into regular chunks so that the data can be signed by the $TSA_{selected}$ at all. Chunks can be formed in different ways (fixed size, fixed time span, etc.). Depending on the use case and individual preferences of the participants, a different way may be favored. Therefore, the participants can also decide by themselves under which conditions they subdivide their data. The chunk is then hashed using SHA-256 and transmitted to the $TSA_{selected}$. Analogous to the signing process in Figure 1, the hashed chunk is hashed again by the $TSA_{selected}$ and signed with the private key of the $TSA_{selected}$. Afterwards, the resulting time stamp response (TSR) is sent back to the appropriate participants so they can persist the TSR together with the original data chunk locally on its own infrastructure. This process is repeated until the participant decides that the hazardous situation has ended, and proof of the process is not required anymore.

4.4. Public Trusted Entities Repository

The Public Trusted Entities Repository is a component that can be used *optionally* by participants for publishing publicly which companies, organizations and TSAs they trust. A participant's trusted entities and TSAs are expressed as a separate set of entities and TSAs. Thus, for example, $\phi(\mathrm{P_A}) = \{\mathrm{P_B},\ \mathrm{P_C}\ ,\ \ldots\}$ represents the trusted entities and $\phi(\mathrm{P_A}) = \{\mathrm{P_B},\ \mathrm{P_C}\ ,\ \ldots\}$ the trusted TSAs of participant $\mathrm{P_A}$. Using these two sets, it becomes possible for participants, analogous to the Web of Trust, to derive transitive trust relations by adding the TSAs of trusted entities to their own trusted TSAs or to look at which other entities their trusted entities rely on in order to add their trusted TSAs to their own *trust list* (c.f. Section 4.2).

Example: If participant A trusts organization B, then participant A can look up in the Public Trusted Entities Repository whether organization B has published which TSAs and organizations it trusts. If participant A finds an entry for organization B, it can view its TSAs and add them to its own set of trusted TSAs.

In this way, the process of finding trusted TSAs is simplified by allowing the participants to derive them via their relationships instead of having to check each TSA individually. At the same time, the repository should increase the number of trusted TSAs per participant to maximize the probability of a common intersection between participants. Nevertheless, the use of the repository is only optional, and participants are free to derive their list of trusted TSAs according to their own preferences.

5. Application and Evaluation

In the following, we present a proof-of-concept prototype to show the applicability of the architectural framework and the negotiation algorithm that was introduced in Section 4. For this purpose, we first describe how the approach was prototypically implemented. Then, an exemplary use-case is presented in which the approach is practically applied under real conditions in a maritime testbed infrastructure. Finally, the results of the application are measured and discussed with respect to the system's performance.

5.1. Implementation

In the following, we introduce the technologies we used for the proof-of-concept implementation of the presented approach. The prototypical implementation will serve as the basis for the following application and evaluation. The full implementation has been published and can be used for research purposes under the CC BY 4.0 license (DOI: 10.5281/zenodo.7323786).

General: The prototype is mainly implemented in Java. The TSA Negotiator and the Data Recorder were implemented analogously to the outlined processes in Figures 4 and 5. For the handling of any cryptographic operations and security-related protocols, such as the local creation of a timestamping request, the library Bouncy Castle was used. Bouncy Castle is a collection of open-source cryptographic programming interfaces [43]. It provides many methods for handling, e.g., X.509 certificates, key pairs, and timestamping authorities.

Timestamping Authority: For the certification of the data, an open timestamping authority of the "German Research Network (DFN)" is used. For research purposes, the TSA can be used free of charge. The exchange of data complies with RFC 3161. The TSA used can easily be replaced by any other RFC 3161-compliant TSA via the adjustment of a single parameter in the implementation [44].

Data Recording Database: For the persistence of the recorded data and the TSA signatures, a MongoDB instance is utilized. MongoDB is a No-SQL database that is well suited for storing semi-structured and unstructured data. Therefore, the database is ideally situated to persist the original records data, which can occur in a wide variety of different file formats, along with the associated signature of the TSA [45].

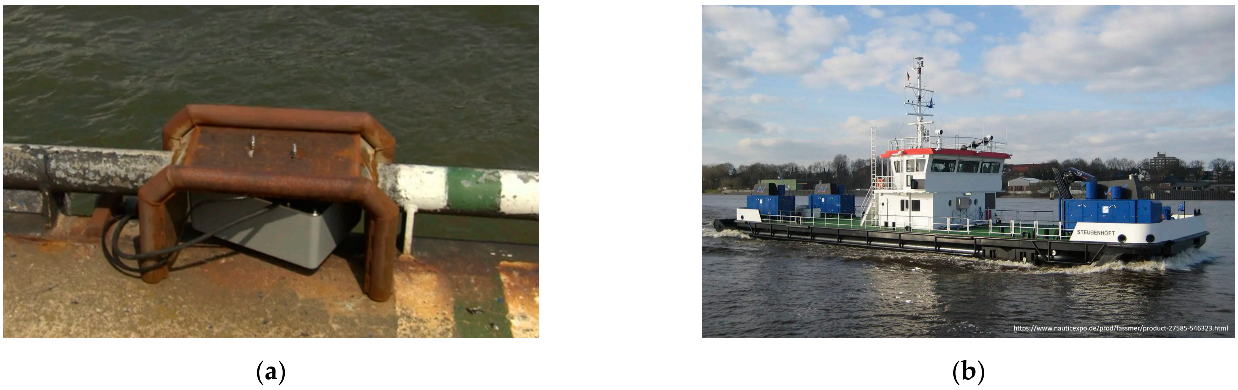
Communication: For communication between the participants, a simple peer-2-peer was implemented using the client-server framework Netty. The entire TSA negotiation is performed over the P2P network. The data that needs to be transmitted is exchanged with each other via TCP/IP.

Deployment: For a user-friendly deployment on multiple machines, the implementation was fully containerized using Docker [46]. For this purpose, two Docker containers were created containing all components, methods, and interfaces needed.

5.2. Evaluation Scenario

During mooring maneuvers, collisions between vessels and the quay walls occur regularly. Often, minor damages remain undetected at first and are only noticed by the port operator after some time. A subsequent clarification of which vessel caused the damage is no longer possible so that the port remains on the incurred costs. To prevent this, port operators are interested in documenting the berthing maneuvers of vessels in a trustworthy way. To evaluate the presented approach for the decentralized documentation of maritime incidents, we applied our system to a real-world problem of port operators and assessed it with respect to its functionality and performance.

For this purpose, the system was embedded in the *SmartKai* testbed in Cuxhaven, Germany (c.f. Figure 6a). The *SmartKai* testbed was set up for the development of a port-side assistance system to support pilots and captains during the berthing of vessels. This is realized by measuring the distance of a vessel to the quay wall using LiDAR sensors and making it available to the stakeholders in real-time. In contrast to AIS, the measurements are not limited to a single reference point but cover the entire contour of a vessel and are available to the stakeholders at a much higher frequency (5 Hz). In total, the *SmartKai* testbed provides 8 LiDAR sensors, radar, and AIS data over the entire port area, and other environmental data such as tidal, weather, and visibility information. Due to the high number of available sensors and the realistic environmental conditions such as the used hardware and network, the testbed is ideally suited to evaluate the presented approach properly. The functionality of the presented approach is demonstrated by a mooring maneuver of the vessel *Steubenhoeft* at the quay wall in the harbor of Cuxhaven. The *Steubenhoeft* is a dredger that is 40 m long and 10 m wide (c.f. Figure 6b). In principle, the proposed approach could also be used to document any other traffic situation.



https://www.nauticexpo.de/prod/fassmer/product-27585-546323.html

(**a**) (**b**)

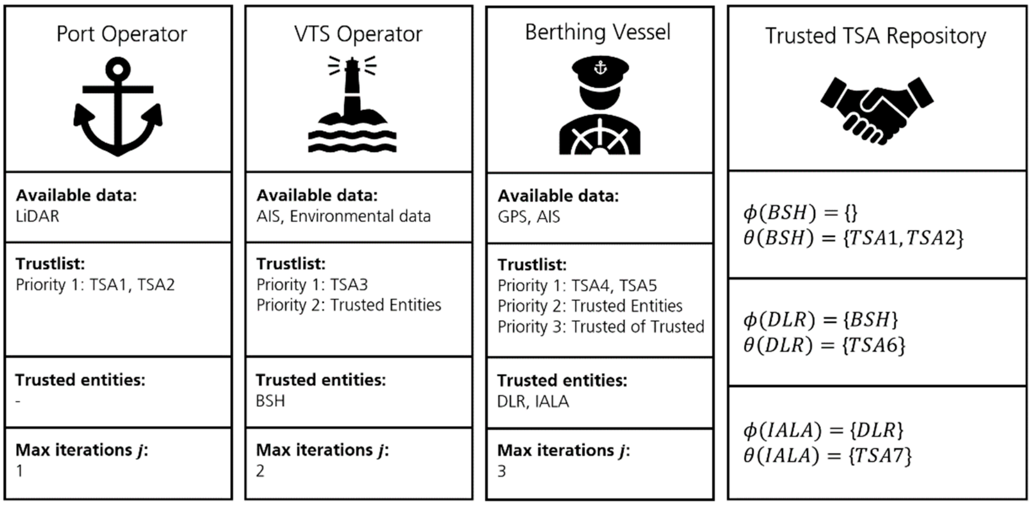
Figure 6. (**a**) LiDAR sensor at the quay wall in Cuxhaven, Germany, to measure the distance between the quay wall and vessels (part of the SmartKai infrastructure) (**b**) Dredger vessel *Steubenhoeft* with which the mooring maneuver that has to be documented was performed.

Within the evaluation mooring scenario, a total of three participants take part (c.f. Figure 7):

(1) The Port Operator who operates the port is interested in ensuring that the quay wall of the port is not damaged and has installed 8 LiDAR sensors at the quay wall with an update frequency of 5 Hz. The sensors' measurements are synchronized so that they can be collected and signed together. Furthermore, the Port Operator trusts TSA1 and TSA2 but does not rely on any other external entity.

(2) The Vessel Traffic Service Operator knows the Port Operator and supports the berthing documentation by providing valuable environmental data about the port area. The VTS operator collects AIS, visibility, tidal, and wind data. The AIS data is received several times per second from surrounding vessels at irregular intervals. The installed wind sensors update with a frequency of 10 Hz, and the visibility sensor every minute. In contrast, the tidal information is retrieved only once per hour.The VTS Operator mainly trusts TSA3. If it is not possible to agree directly on the first priority, the VTS Operator also trusts the TSAs of its trusted entity—the BSH.

(3) The Berthing Vessel is the vessel that wants to moor in the port. The Berthing Vessel stores its received AIS messages and own GPS positions. The GPS data is encoded in the NMEA0183 format and is updated with 1 Hz. The vessel relies primarily on TSA4 and TSA5. However, the vessel is willing to expand its list in two additional iterations: Once by the TSAs of its trusted entities—the DLR and the IALA and another time by the TSAs of the trusted entities of the DLR and IALA.



Port Operator	VTS Operator	Berthing Vessel	Trusted TSA Repository
Available data: LiDAR	**Available data:** AIS, Environmental data	**Available data:** GPS, AIS	$\phi(BSH) = \{\}$ $\theta(BSH) = \{TSA1, TSA2\}$
Trustlist: Priority 1: TSA1, TSA2	**Trustlist:** Priority 1: TSA3 Priority 2: Trusted Entities	**Trustlist:** Priority 1: TSA4, TSA5 Priority 2: Trusted Entities Priority 3: Trusted of Trusted	$\phi(DLR) = \{BSH\}$ $\theta(DLR) = \{TSA6\}$
Trusted entities: -	**Trusted entities:** BSH	**Trusted entities:** DLR, IALA	$\phi(IALA) = \{DLR\}$ $\theta(IALA) = \{TSA7\}$
Max iterations *j*: 1	**Max iterations *j*:** 2	**Max iterations *j*:** 3	

Figure 7. Initial situation of the evaluation scenario.

The hardware and software of the three participants is deployed on three separate machines. The Port Operator and VTS Operator instances are located at the quay and are connected to each other via a local network. These machines have an Intel Core i7-8700T processor with 16GB DDR4 RAM. The machine of the Berthing Vessel is located on board. It is based on an Intel Core i7-1185G7 processor with 16GB DDR4 RAM. All machines have an LTE connection.

The Public Trusted Entities Repository is available for all the participants. In our implementation, any entity (such as an organization, an authority, or a company) is allowed to publish their trusted organizations ϕ and trusted TSAs θ. However, in reality, it is primarily organizations, authorities, and institutes that publish trust lists, as they have the resources to verify the trustworthiness of individual TSAs. In addition, these also serve as an anchor of trust so that many participants could also trust their trust lists. In the specific test scenario, the repository has three entries from BSH, DLR, and IALA (c.f. Figure 7). The parties are the public authority "Federal Maritime and Hydrographic Agency of Germany (BSH)," the intergovernmental organization "International Association of Marine Aids to Navigation and Lighthouse Authorities (IALA)," and the research institute "German Aerospace Center (DLR)." Note that the entries in the Trusted TSA Repository are only an example and do not necessarily correspond to reality.

5.3. Application and Results

To demonstrate the presented approach, the evaluation scenario was applied to the four phases presented in Section 4. In addition, it is described how the generated documentation of the mooring maneuver can be used in case of a conflict.

(1) Initiate the documentation: Since the Port Operator wants to protect its port infrastructure, they are also responsible for defining when a traffic situation becomes critical and worthy of documentation. As already described in Section 4, for this purpose, a trigger needs to be defined that determines when a critical traffic event begins and thus the documentation is initiated. In the case of a mooring maneuver, a geographical region in front of the quay wall was utilized (c.f. Figure 8).

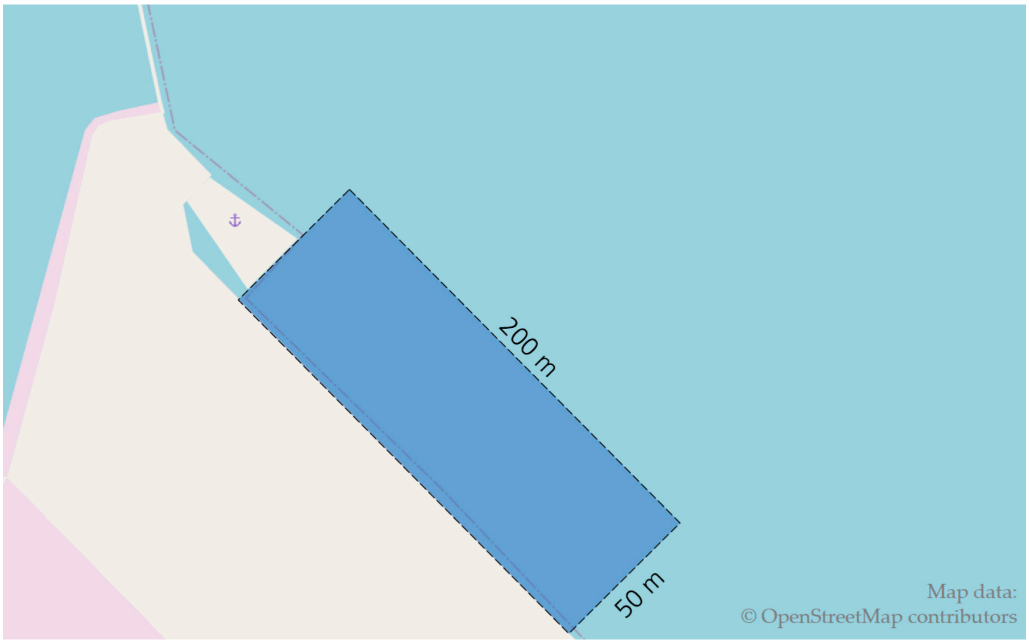


Figure 8. Geographical region in front of the quay wall where a documentation process is initiated when a vessel enters (Cuxhaven, Germany).

As a trigger, we defined a 50 m × 200 m box in front of the Amerikahafen in Cuxhaven. If a Berthing Vessel enters this region, the documentation's initialization trigger is released. All vessels that request to berth in the port are informed that they have to participate in the documentation process when entering the port. Additionally, the IP-address of the port is also published at which a vessel has to report as soon as it enters the critical region.

(2) Establish a communication channel: For establishing a communication channel between all participants, the Berthing Vessel is obligated to contact the IP of the Port Operator by using the prototype. Since the Port Operator and the VTS Operator know each other and the VTS Operator always supports the berthing documentation, the two participants are permanently connected via a P2P network. In this way, the Port Operator is connected to the VTS Operator and the Berthing Vessel. To enable the Port Operator and the VTS Operator to communicate with each other as well, the Port Operator broadcasts each other's IP addresses to all participants. In this way, each participant knows the IP of every other participant and can thus establish a P2P connection.

(3) Derive a common trusted set of TSAs: Once a communication channel is established, participants use it to negotiate a mutually trusted set of TSAs. For this purpose, the participants share their preferred familiar TSAs in j iterations to all other participants via the P2P network. The negotiation proceeds according to the Negotiation Protocol (c.f. Figure 4), as outlined in detail in Table 1.

Table 1. Familiar TSAs from each participant for each negotiation iteration j and common TSA intersection according to the Negotiation Protocol.

Iteration j	Port Operator	VTS Operator	Berthing Vessel	TSA Inters.
1	TSA 1, 2	TSA 3	TSA 4, 5	∅
2	TSA 1, 2	TSA 1, 2, 3	TSA 4, 5, 6, 7	∅
3	TSA 1, 2	TSA 1, 2, 3	TSA 1, 2, 4, 5, 6, 7	TSA 1, 2

- Iteration $j = 1$: According to the evaluation scenario (c.f. Figure 7), in the first iteration, each participant shares their first priority of trusted TSAs. Therefore, the Port Operator broadcasts TSA1 and TSA2, the VTS Operator broadcasts TSA3, and the Berthing

Vessel TSA4 and TSA5. After receiving messages from all other participants, the participant can determine the intersection of the trusted TSAs locally. In iteration $j = 1$, the intersection is an empty set, so another iteration is initiated along the negotiation protocol.

- Iteration $j = 2$: In the second iteration, participants are requested to expand their list of trusted TSAs. Therefore, the VTS Operator and the Berthing Vessel include the TSAs of their trusted entities. The VTS Operator trusts the BSH. Based on private relationships or the use of the Public Trusted Entities Repository, the VTS Operator knows which TSAs are trusted by the BSH. In this case, the BSH trusts TSA1 and TSA2, so they are included into the VTS Operator's TSA list. The Berthing Vessel relies on the DLR and the IALA. The DLR trusts TSA6 and the IALA trusts TSA7. Therefore, both TSAs are also included in the list of the Berthing Vessel. The Port Operator follows a very strict strategy and trusts only his own two TSAs. There is no extension of the list at all. Afterwards, every participant broadcasted their trust list and it is checked again whether a common intersection exists between the participants. After iteration $j = 2$, the intersection at TSA is still empty, so the next iteration is initiated.
- Iteration $j = 3$: In the third iteration, only the Berthing Vessel is further expanding its list of trusted TSAs. For this purpose, the Berthing Vessel relies on the trusted TSAs of the trusted entities from the DLR and IALA. Therefore, TSA1 and TSA2 are also added to the trust list of the Berthing Vessel. The VTS and Port Operators are not expanding their trust list. Again, the intersection of all trust lists is determined locally. In the iteration, the participants find a common basis with TSA1 and TSA2.

This fulfills the condition $\cap_{\mathrm{i}=1}^{\mathrm{n}}\pi_{\mathrm{j}}(\mathrm{P_i}) \neq \varnothing$, so the negotiation protocol terminates at this point. The participants can use either TSA1 or TSA2 to sign their data. If no common TSA has been found at this point, it would not be possible to find a common TSA for all participants, since all participants have already extended their lists to the maximum ($\mathrm{j_{max}} = 3$). In this case, an attempt would be made to find a common TSA for as large a subset of participants as possible (c.f. Figure 4, left part).

Record the data: After agreeing on a common set of TSAs, no further communication takes place over the P2P network. Instead, each participant records their data analogously to the process of Figure 5 by chunking the data in 10-s intervals and hashing it with SHA256. Each participant then sends the hashes directly to one of the determined TSAs. In our application, the Port Operator and Berthing Vessel used the TSA1 and the VTS Operator utilized TSA2. Finally, the data chunks of the data stream are persisted in the local recording database together with the signature of the TSA.

Only in case of an actual conflict, e.g., the Berthing Vessel damages the quay wall, the signed records are retrieved from the databases as proof of the actual course of events. In this case, the Port Operator and the VTS Operator would disclose their data to show that the Berthing Vessel damaged the quay wall. In addition, the participants can verify, with the process described in Figure 2, that the data already existed during the berthing maneuver. Since the Berthing Vessel also trusts the signing TSA, it cannot deny that the data has been manipulated afterwards by the Port or VTS Operators. Of course, the Berthing Vessel can also provide its own signed data in order to exonerate itself. In general, a large and heterogeneous data foundation helps to reconstruct the real course of a berthing maneuver. In this way, the data of several participants can be compared with each other in order to check whether they provide a coherent overall picture. For example, the vessel's track recorded by the LiDAR sensor could be compared with the AIS recordings of the vessel itself. Even in the case of real-time manipulation during a critical traffic event, data recordings that do not fit into the overall picture could be detected in this way so that they can still be identified as being manipulated.

Since it is not always foreseeable when a critical traffic situation will occur, it is also essential to evaluate the required time from the initiation of the documentation process to the final signed data. Therefore, as part of the practical application of the approach, we collected data for the duration of creating the communication channel, deriving the common TSA basis, and signing and storing a data chunk (c.f. Table 2). In addition, we also ran the scenario 1000 times simulatively to see if the durations also matched the values from the field test. The simulation was executed on a computer with an Intel Core i9-9900K processor and 32 GB of RAM.

Table 2. Time required by the approach in the evaluation scenario depending on the respective phase.

Phase	Duration in SmartKai Testbed	Average Duration in Simulation
Establishing communication channel	636 ms	30 ms
Derive a common trusted set of TSAs	313 ms	100 ms
Signing and storing of a data chunk	118 ms	106 ms
Total	1067 ms	236 ms

As can be seen, the presented approach requires a total of 1067 milliseconds under real conditions from initiating the communication channel to storing the first signed chunk in the database. This time is composed of the three phases of establishing the communication channel with 636 ms, the TSA negotiation of the participants with 313 ms, the signing of the data with a TSA, and subsequent storage in the local database with a duration of 118 ms. Even faster results were obtained within our 1000 simulative runs. On average, the total duration was 236 ms. The creation of the communication channel with 30 ms was significantly faster than in the testbed. This can be explained by the faster and more stable internet connection in the simulation environment. Furthermore, the derivation of a TSA set with 100 ms and signing and saving of the chunks with 106 ms was also faster than in the testbed.

5.4. Discussion

The functionality of the approach for a decentralized documentation of critical traffic situations was demonstrated based on a realistic evaluation scenario within a maritime SmartKai testbed. Furthermore, the presented approach fulfills the requirements of Section 2.4:

(R1) Firstly, if a participant wishes to document its data recording, it must be sent to the identified TSA in hashed form. This makes the documentation process completely independent of the format and content of the underlying data. The only prerequisite is that it must be possible to hash the documented data. This is no issue for any data stored in files, and even streamed data can be hashed by breaking it down into individual packets.

(R2) Secondly, as proof that the data has not been manipulated after recording, a hash of the data is sent to a TSA approved by all participants. This TSA provides the hash with a timestamp and signs it with its digital signature. In this way, it can be cryptographically verified at any time whether the data has been manipulated after it has been recorded.

(R3) The presented approach puts a strong focus on protecting the sovereignty of individual data providers. Each participant is able to record and persist his data independently. As long as there is no conflict, the data is also not shared with the other participants. Even in the event of a conflict, the data provider has the option not to share the data. Additionally, only a hash of the original data is transmitted to the TSA itself, so even the TSA cannot draw any conclusions about the actual data.

(R4) Furthermore, anyone can join the P2P network used for communication as long as the participant knows the IPs of the other participants. The best way to distribute the IPs among the participants depends on the use case for which the documentation has to be created.

(R5) Lastly, if not all participants are equipped with the system for documentation, it is still possible for the remaining participants to document a situation among themselves in a tamper-proof manner. If no participant is equipped with the system or it is not possible to reach an agreement during TSA negotiation, the fallback TSA mechanism takes effect (c.f. Section 4.2). In this case, the participant documents his data with a TSA selected by himself in order to establish at least a basic trust in the documentation.

In order for the approach to be used for documenting critical traffic situations, it is important that the entire process from establishing contact to documenting remains within a reasonable time frame. Unlike in other domains such as the automotive domain, critical traffic events in the maritime domain can often be determined several minutes in advance, so the required time of 1068 ms should not have a negative impact in most cases.

In addition, the developed approach was also compared with similar existing approaches to discuss the strengths and weaknesses of each approach (cf. Table 3).

Table 3. Comparison of different approaches for the reconstruction of maritime situations (+: fulfilled; o: partially fulfilled; -: not fulfilled).

Approach	Variety of Data	Tamper-Proof	Data Sovereignty	Complexity	Connectivity
Voyage Data Recorder	o	+	+	-	+
Logbook	-	-	+	+	+
Blockchain-based approach	+	+	+	-	-
TSA-based approach	+	+	+	o	-

We compared the existing methods to reconstruct traffic events, such as VDR and logbook, the blockchain-based approach and the TSA-based approach presented in this paper. The evaluation criteria include the possibility of documenting any data (Variety of data), the tamper-proofness of the documentation (Tamper-proof), the local persistence of the data (Data Sovereignty), the complexity of the overall system and the readout process (Complexity), and the dependence on an Internet connection (Connectivity).

The Voyage Data Recorder can record various information in an automated way but is mainly limited to information about its own ship. However, due to its robustness and certification, the data is kept very secure, making it difficult to manipulate it [47]. In addition, the data is stored directly on the ship, so it does not need to be migrated to an external infrastructure. However, as mentioned above, retrieving the data turns out to be time-consuming, which is why the approach is not used in practice to clarify minor incidents [4]. A connection to the Internet is not required.

The most important events on board are documented in a logbook [3]. In general, this is a manual process that is still carried out by hand. Accordingly, a logbook does not record detailed data. In addition, it is only expected that the logbook will be filled out truthfully [48]. Therefore, it can be comparatively easily manipulated. However, on the positive side, the data records are only stored on the vessel itself, and the effort for the documentation and readout is low. The logbook also does not require an internet connection.

The blockchain and TSA-based approaches differ from the classic approaches. Both approaches are, in principle, data-independent so that any data can be documented. By storing a hash on a blockchain or signing the hash with a TSA, subsequent changes to the data can be easily identified. In both cases, the data is communicated externally, but since the data is only transmitted as a hash, no conclusions can be drawn about the original data (cf. Section 4). The complexity of the blockchain solution depends very much on how the blockchain itself is designed. Accordingly, the effort required to enter and verify a hash can be very high. However, since each TSA follows a standardized process and the data is cryptographically signed, the integrity of the data can be verified by examining the signature in a computationally efficient way and without a network of multiple blockchain nodes. Therefore, the overall complexity of the approach can be considered lower than the blockchain-based solution. However, both approaches require an internet connection in order to document the data in a tamper-proof way. In the near future, it is expected that ships will also have a permanent connection via satellites [49–51]. In regions close to land, most ships already have an LTE connection. Therefore, this criterion should not be a major limitation in the future.

Overall, the existing approaches for documenting traffic situations focus on data from the own vessel. Other external data sources are not part of the documentation. Nevertheless, the data from other sources could contribute to a better understanding of the cause of critical traffic situations. Therefore, establishing such a system in the maritime domain would make sense. In principle, the blockchain-based and TSA-based approaches are suitable for tamper-proof documentation. However, the complexity of a TSA-based approach will usually be lower, so it should be preferred.

6. Summary and Conclusions

In this paper, we have presented an approach for the creation of a decentralized and trustworthy documentation of maritime traffic situations. Our research literature reveals that the clarification of minor traffic incidents presents a particular difficulty, since the VDR on board a vessel is often only read by the authorities in the case of major incidents. A forgery-proof data basis is generally not available for clarification of smaller incidents. Furthermore, apart from ship-related data, e.g., the VDR, and a lot of contextual information such as weather information, is recorded by other entities. This contextual information can be very important for a proper documentation of an incident but is currently not addressed by existing approaches.

Therefore, we have taken up this challenge and by first analyzing which methods already exist to persist data in a tamper-proof way. Besides central authorities that perform the documentation and blockchain-based approaches, timestamping authorities that have the task of adding a timestamp to data in a cryptographically verifiable way can also be utilized. Compared to the other approaches, TSAs have the advantage of being able to certify the integrity of data records using standardized and computationally efficient operations. However, as the maritime domain is an international industry with many different participants from a large number of countries, it is unrealistic to assume that all participants will agree on a single central TSA. For the agreement on a TSA, the Negotiation Protocol was introduced, which ensures that a TSA with the maximum trust will be found among the participants. For communication, a P2P network is established when a document worthy situation is initiated. The presented approach also allows storing data records on the documenting party's infrastructure. In this way, the participants retain complete control over their own data. Only in the case of a conflict resolution, the participant needs to share his certified data with the other parties.

To demonstrate the functionality of the approach, we integrated our implementation into the infrastructure of the SmartKai Testbed in Cuxhaven. The approach was successfully tested based on a berthing maneuver involving three different parties. It was shown that even in real deployment environments, the parties could communicate over the dynamically created P2P network to agree along the Negotiation Protocol on a common TSA, to certify their own data using the authority and persist it in a local database.

Nevertheless, it should be noted that the presented approach also has some limitations. For example, a continuous internet connection or additional effort to establish non-IP-based communication to a gateway is required to agree on a TSA and to send the data recordings to the TSA for certification. Especially when the approach should be used on the open sea, where ships usually have limited or no internet connection, this could cause problems. Still, by using alternative communication channels such as the VHF Data Exchange System (VDES), or by caching the required information beforehand, it would be possible to exchange the required information even without a stable internet connection. A further limitation is that only parties that already have the required software installed on their infrastructure can participate in the documentation process. Parties that do not have the software or hardware can not be included in the documentation process. Lastly, the presented approach only grants that the data already existed on the date issued by the TSA. However, the data could have been manipulated before it was signed by the provider. Therefore, it is extremely important to look at data from multiple participants when resolving conflicts and see if they match.

In future work, we envision being able to ensure that a participant's data records have not been manipulated before the time of certification. In addition, we would like to investigate whether alternative communication channels can be used for the TSA negotiation, so that the parties can agree on a common basis of trust even if they do not have a stable internet connection in the initiation of a critical traffic incident. Furthermore, a legal classification of the approach would be interesting in order to be able to assess whether the decentralized documentation may also be used as evidence according to the current legal situation.

Author Contributions: Conceptualization, D.J., J.M., H.W. and A.H.; data curation, D.J., J.M. and H.W.; investigation, D.J., J.M. and H.W.; methodology, D.J., J.M. and H.W.; project administration, D.J.; software, D.J., J.M. and H.W.; supervision, A.H.; validation, D.J. and H.W.; visualization, D.J. and H.W.; writing—original draft, D.J., J.M. and H.W.; writing—review and editing, D.J., J.M., H.W. and A.H. All authors have read and agreed to the published version of the manuscript.

Funding: This work has been funded by the German Federal Ministry of Transport and Digital Infrastructure (BMVI) within the funding guideline "Innovative Hafentechnologien" (IHATEC) under the project SmartKai with the funding code 19H19008E and the FuturePorts project of the German Aerospace Center (DLR).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest. The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript; nor in the decision to publish the results.

References

1. Morcom, A.R. Flight data recording systems: A brief survey of the past developments, current status and future trends in flight recording for accident investigation and operational purposes. *Aircr. Eng. Aerosp. Technol.* **1970**, *42*, 12–16. [CrossRef]
2. Yoder, T.A. Development of Aircraft Fuel Burn Modeling Techniques with Applications to Global Emissions Modeling and Assessment of the Benefits of Reduced Vertical Separation Minimums. Master's Thesis, Massachusetts Institute of Technology, Cambridge, MA, USA, 2007. Available online: https://dspace.mit.edu/handle/1721.1/39713 (accessed on 28 July 2022).
3. International Maritime Organization (IMO). SOLAS Chapter V—Safety of Navigation. Available online: https://www.imorules.com/SOLAS_REGV.html (accessed on 5 October 2022).
4. Cantelli-Forti, A. Forensic Analysis of Industrial Critical Systems: The Costa Concordia's Voyage Data Recorder Case. In Proceedings of the 2018 IEEE International Conference on Smart Computing (SMARTCOMP), Taormina, Italy, 18–20 June 2018; pp. 458–463. [CrossRef]
5. IMO. Casualties. 2019. Available online: https://www.imo.org/en/OurWork/MSAS/Pages/Casualties.aspx (accessed on 31 July 2022).
6. Bundesstelle für Seeunfalluntersuchung. Bundesstelle für Seeunfalluntersuchung—Untersuchungsberichte. 2022. Available online: https://www.bsu-bund.de/DE/Publikationen/Unfallberichte/Unfallberichte_node.html (accessed on 31 July 2022).
7. UK Government. Marine Accident Investigation Branch Reports. *GOV.UK*. 2022. Available online: https://www.gov.uk/maib-reports (accessed on 31 July 2022).
8. Voigt, P.; Von dem Bussche, A. *The EU General Data Protection Regulation (GDPR)*; Springer International Publishing: Cham, Switzerland, 2017. [CrossRef]
9. Hummel, P.; Braun, M.; Tretter, M.; Dabrock, P. Data sovereignty: A review. *Big Data Soc.* **2021**, *8*, 2053951720982012. [CrossRef]
10. Mason, S. *Electronic Signatures in Law*; University of London Press: London, UK, 2016.
11. Stallings, W.; Brown, L. *Computer Security: Principles and Practice*, 5th ed.; Global ed.; Pearson: New York, NY, USA, 2018. Available online: https://elibrary.pearson.de/book/99.150005/9781292220635 (accessed on 16 August 2022).
12. Weinert, B.; Park, J.H.; Christensen, T.; Hahn, A. A Common Maritime Infrastructure for Communication and Information Exchange. In Proceedings of the 19th IALA Conference 2018, Incheon, Republic of Korea, 27 May–2 June 2018.
13. Sheth, H.; Dattani, J. Overview of blockchain technology. *Asian J. Converg. Technol.* **2019**, *5*, 1–4. [CrossRef]
14. Koens, T.; Poll, E. What blockchain alternative do you need? In *Data Privacy Management, Cryptocurrencies and Blockchain Technology*; Springer: Berlin/Heidelberg, Germany, 2018; pp. 113–129.
15. Andolfatto, D. Blockchain: What it is, what it does, and why you probably don't need one. *Fed. Reserve Bank Louis Rev.* **2018**, *100*, 87–95. [CrossRef]

16. *ANSI X9.95-2016*; Trusted Time Stamp Management and Security. American National Standards Institute (ANSI): Annapolis, MD, USA, 2016.
17. Zuccherato, R.; Cain, P.; Adams, D.C.; Pinkas, D. Internet X.509 Public Key Infrastructure Time-Stamp Protocol (TSP); No. 3161; RFC Ed. August 2001. Available online: https://www.rfc-editor.org/info/rfc3161 (accessed on 5 November 2022).
18. Buldas, A.; Laud, P.; Lipmaa, H.; Villemson, J. Time-stamping with binary linking schemes. In *Advances in Cryptology—CRYPTO '98*; Krawczyk, H., Ed.; Springer: Berlin/Heidelberg, Germany, 1998; Volume 1462, pp. 486–501. [CrossRef]
19. Haber, S.; Stornetta, W.S. How to time-stamp a digital document. *J. Cryptol.* **1991**, *3*, 99–111. [CrossRef]
20. Moller, J.; Jankowski, D.; Lamm, A.; Hahn, A. Data Management Architecture for Service-Oriented Maritime Testbeds. *IEEE Open J. Intell. Transp. Syst.* **2022**, *3*, 631–649. [CrossRef]
21. Nofer, M.; Gomber, P.; Hinz, O.; Schiereck, D. Blockchain. *Bus. Inf. Syst. Eng.* **2017**, *59*, 183–187. [CrossRef]
22. Jamthagen, C.; Hell, M. Blockchain-Based Publishing Layer for the Keyless Signing Infrastructure. In Proceedings of the 2016 Intl IEEE Conferences on Ubiquitous Intelligence & Computing, Advanced and Trusted Computing, Scalable Computing and Communications, Cloud and Big Data Computing, Internet of People, and Smart World Congress (UIC/ATC/ScalCom/CBDCom/IoP/SmartWorld), Toulouse, France, 18 July 2016; IEEE: New York, NY, USA, 2016; pp. 374–381. [CrossRef]
23. Cucurull, J.; Puiggalí, J. Distributed Immutabilization of Secure Logs. In *Security and Trust Management*; Barthe, G., Markatos, E., Samarati, P., Eds.; Springer International Publishing: Cham, Switzerland, 2016; Volume 9871, pp. 122–137. [CrossRef]
24. Buldas, A.; Kroonmaa, A.; Laanoja, R. Keyless Signatures' Infrastructure: How to Build Global Distributed Hash-Trees. In *Secure IT Systems*; Riis Nielson, H., Gollmann, D., Eds.; Springer: Berlin/Heidelberg, Germany, 2013; Volume 8208, pp. 313–320. [CrossRef]
25. Gervais, A.; Ritzdorf, H.; Karame, G.O.; Capkun, S. Tampering with the Delivery of Blocks and Transactions in Bitcoin. In Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security, Denver, CO, USA, 12 October 2015; pp. 692–705. [CrossRef]
26. Szalachowski, P. (Short Paper) Towards More Reliable Bitcoin Timestamps. In Proceedings of the 2018 Crypto Valley Conference on Blockchain Technology (CVCBT), Zug, Switzerland, 20 June 2018; pp. 101–104. [CrossRef]
27. Putz, B.; Menges, F.; Pernul, G. A secure and auditable logging infrastructure based on a permissioned blockchain. *Comput. Secur.* **2019**, *87*, 101602. [CrossRef]
28. Shekhtman, L.; Waisbard, E. EngraveChain: A Blockchain-Based Tamper-Proof Distributed Log System. *Futur. Internet* **2021**, *13*, 143. [CrossRef]
29. Wust, K.; Gervais, A. Do you Need a Blockchain? In Proceedings of the 2018 Crypto Valley Conference on Blockchain Technology (CVCBT), Zug, Switzerland, 2 June 2018; pp. 45–54. [CrossRef]
30. Crosby, S.A.; Wallach, D.S. Efficient Data Structures for Tamper-Evident Logging. In Proceedings of the 18th Conference on USENIX Security Symposium, Montreal, QC, Canada, 14–18 August 2009; pp. 317–334.
31. Schneier, B.; Kelsey, J. Cryptographic Support for Secure Logs on Untrusted Machines. In Proceedings of the 7th USENIX Security Symposium (USENIX Security 98), San Antonio, TX, USA, 26 January 1998. Available online: https://www.usenix.org/conference/7th-usenix-security-symposium/cryptographic-support-secure-logs-untrusted-machines (accessed on 5 November 2022).
32. Levin, D.; Douceur, J.R.; Lorch, J.R.; Moscibroda, T. TrInc: Small Trusted Hardware for Large Distributed Systems. In *NSDI*; Springer: Berlin/Heidelberg, Germany, 2009.
33. Sinha, A.; Jia, L.; England, P.; Lorch, J.R. Continuous Tamper-Proof Logging Using TPM 2.0. In *Trust and Trustworthy Computing*; Holz, T., Ioannidis, S., Eds.; Springer International Publishing: Cham, Switzerland, 2014; Volume 8564, pp. 19–36. [CrossRef]
34. MVEDR-EC-Motor Vehicle Event Data Recorder Brake and Electronic Control Working Group. IEEE 1616-2021-Standard for Motor Vehicle Event Data Recorder (MVEDR). IEEE. Available online: https://standards.ieee.org/ieee/1616/10329/ (accessed on 5 November 2022).
35. Yao, Y.; Atkins, E. The Smart Black Box: A Value-Driven High-Bandwidth Automotive Event Data Recorder. *IEEE Trans. Intell. Transp. Syst.* **2020**, *22*, 1484–1496. [CrossRef]
36. Singleton, N.; Daily, J.; Manes, G. Automobile Event Data Recorder Forensics. In *Advances in Digital Forensics IV*; Ray, I., Shenoi, S., Eds.; Springer: Boston, MA, USA, 2008; Volume 285, pp. 261–272. [CrossRef]
37. Vinzenz, N.; Eggendorfer, T. Forensic Investigations in Vehicle Data Stores. In Proceedings of the Third Central European Cybersecurity Conference, Munich, Germany, 14–15 November 2019; pp. 1–6. [CrossRef]
38. Wahab, A.; Waseso, B.; Pranoto, H. Synchronization of Catch Fish Data in Fisheries e-Logbook with a Vessel Monitoring System. *Int. J. Adv. Technol. Mech. Mechatron. Mater.* **2021**, *2*, 46–54. [CrossRef]
39. Mion, M.; Piras, C.; Fortibuoni, T.; Celić, I.; Franceschini, G.; Giovanardi, O.; Belardinelli, A.; Martinelli, M.; Raicevich, S. Collection and validation of self-sampled e-logbook data in a Mediterranean demersal trawl fishery. *Reg. Stud. Mar. Sci.* **2015**, *2*, 76–86. [CrossRef]
40. Young, W. What Are Vessel Traffic Services, and What Can They Really Do? *Navigation* **1994**, *41*, 31–56. [CrossRef]
41. Chang, S. Development and analysis of AIS applications as an efficient tool for vessel traffic service. In Proceedings of the Oceans '04 MTS/IEEE Techno-Ocean '04 (IEEE Cat. No.04CH37600), Kobe, Japan, 9–12 November 2004; Volume 4, pp. 2249–2253. [CrossRef]

42. Hepp, T.; Schoenhals, A.; Gondek, C.; Gipp, B. OriginStamp: A blockchain-backed system for decentralized trusted timestamping. *It-Inf. Technol.* **2018**, *60*, 273–281. [CrossRef]
43. Detho, W. Developing a system for securely time-stamping and visualizing the changes made to online news content. *arXiv* **2018**, arXiv:1802.07285. [CrossRef]
44. *IEC TC80*; IEC 63173-2:2022: Maritime Navigation and Radiocommunication Equipment and Systems-Data Interfaces-Part 2: Secure Communication between Ship and Shore (SECOM). IEC: Geneva, Switzerland, 2022. Available online: https://webstore.iec.ch/publication/64543 (accessed on 5 November 2022).
45. Legion of the Bouncy Castle Inc. Bouncy Castle–Documentation. Available online: https://web.archive.org/web/20221012164800/https://www.bouncycastle.org/documentation.html (accessed on 15 November 2022).
46. Deutsches Forschungsnetz (DFN). DFN Timestamping Authority. Available online: https://web.archive.org/web/20220000000000*/https://www.pki.dfn.de/faqpki/faq-zeitstempel (accessed on 15 November 2022).
47. MongoDB, Inc. MongoDB Documentation. Available online: https://web.archive.org/web/20221115034209/https://www.mongodb.com/docs/manual/ (accessed on 15 November 2022).
48. Docker, Inc. Docker Documentation. Available online: https://web.archive.org/web/20221104022709/https://docs.docker.com/ (accessed on 15 November 2022).
49. Piccinelli, M.; Gubian, P. Modern ships Voyage Data Recorders: A forensics perspective on the Costa Concordia shipwreck. *Digit. Investig.* **2013**, *10*, S41–S49. [CrossRef]
50. Schotte, M. Expert Records: Nautical Logbooks from Columbus to Cook. *Inf. Cult.* **2013**, *48*, 281–322. [CrossRef]
51. Wei, T.; Feng, W.; Chen, Y.; Wang, C.-X.; Ge, N.; Lu, J. Hybrid Satellite-Terrestrial Communication Networks for the Maritime Internet of Things: Key Technologies, Opportunities, and Challenges. *IEEE Internet Things J.* **2021**, *8*, 8910–8934. [CrossRef]