Annals of Telecommunications (2023) 78:551–560
https://doi.org/10.1007/s12243-022-00941-8

DLT architectures for trust anchors in 6G

Dennis Krummacker[1] · Benedikt Veith[1] · Daniel Lindenschmitt[2] · Hans D. Schotten[1,2]

Received: 2 September 2022 / Accepted: 15 December 2022 / Published online: 17 January 2023

Abstract

This manuscript investigates viable Distributed Ledger Technology (DLT) architecture approaches to be used as basis for the distribution of integrity verification data. We discuss what can be a Trust Anchor and how the property of trust can be enabled as a service for mobile communications infrastructures. This follows up on a preceding publication, in the course of which a service was developed that can be utilized to create trust and traceability in transactions between other services. Crucial for the integrity of such an audit trail is proof for which side was committing, in case a tampering was detected. For such verification in the aftermath, mechanisms for the distribution of meta data are necessary. Where our ultimate goal is to develop a versatile framework for Trust as a Service (TaaS), the work at hand contributes the investigation on header distribution. We put a major focus on providing Trust as a Service (TaaS) especially in the mobile communications domain since a reliable concept for trustworthiness is indispensable for the vision of organic infrastructures beyond 5G, which means that such networks are flexible regarding their composition and open for stakeholders.

Keywords Trust Anchor · DLT · TaaS · Mobile Communications · 5G · 6G

1 Introduction

Some of the major enhancements of mobile communications beyond 5G do grant worthwhile capabilities but do also inflict new challenges as they impose risks to certain areas previously not an issue there. The idea of Organic Infrastructures [1–5] can present a great deal regarding flexibility, universal applicability and future-oriented evolution.

Part of this vision is that the functional diversity of 6G infrastructures is not limited to the collection provided right with commissioning or restricted to shipment by the dedicated manufacturer of one comprehensive overall system. Instead, expansions can come at any time from anyone; at least technically. But with granting the technical capability for everyone to contribute to a system, the question arises who shall be allowed to do so; or put differently, who can be trusted. Because then intentionally malicious intrusion is easily possible. To avoid this, tools for permission management are vital in order to control which stakeholder is allowed to perform which actions.

Similar for the availability of data throughout a network. Re-thinking the architecture of future mobile communications infrastructures and its dedicated parts (Core, Radio Access Network (RAN)) can aid measuring and exchanging information more efficiently [6]. But again, a certain degree of control over such extensive features is required. Special confidential information might be worth protecting. Or in general, inserting and fetching of data should not be randomly allowed to anyone.

The common issue is Trustworthiness. Who can be trusted — their intention, or even qualification. For that purpose, the authors of the present manuscript consider it to be convenient having trust provided through an infrastructure as a service for utilization by functionalities running within.

✉ Dennis Krummacker
dennis.krummacker@dfki.de

Benedikt Veith
benedikt.veith@dfki.de

Daniel Lindenschmitt
lindenschmitt@eit.uni-kl.de

Hans D. Schotten
schotten@eit.uni-kl.de

[1] Intelligent Networks Research Group, German Research Center for Artificial Intelligence (DFKI GmbH), Trippstadter Str. 122, Kaiserslautern, 67663, Rheinland-Pfalz, Germany

[2] Institute for Wireless Communication and Navigation, University of Kaiserslautern, Gottlieb-Daimler-Straße, Kaiserslautern, 67663, Rheinland-Pfalz, Germany

The work at hand follows up on the previous work published with the IEEE as [7]. This presented a twofold result as it proposed a solution for a trustworthy and verifiable coexistent spectrum allocation mechanism between disjoint infrastructures. A novel service (Spectrum Allocation & Sharing Function (SASF)) alongside additionally required tools were introduced that enable Core-to-Core (CtC) communication between independent 6G systems and that utilize this to perform a negotiation, which eventuates in sharing spectrum inside a common coverage area. Such a mechanism — that is, the negotiation's result as well as the subsequent operation — has to be unalterably traceable. Both parties have to trust each other that they proceed as agreed.

For this purpose, the second part introduced another service based on Distributed Ledger Technology (DLT) for granting trust. This so-called Trust & Traceability Function (TTF) was developed as a Core service that can be employed by other services to create trust in their operation. This was finally designed together so that the two SASFs interact over their respective TTFs through which the negotiation and its result is logged. Afterwards in the active operation phase, the factually utilized spectrum is logged as well. In the end, this enables a trustworthy analysis of the entire transaction, which can ultimately be used, for instance, to issue an invoice.

The current manuscript further elaborates on the topic of trustworthiness. The ultimate goal is to have an universal solution as a modular system component that is capable of granting trust in various types of operations. This can be for example regarding the trust in a stakeholder of a service, the trustworthiness of polled data, trust in a requested transaction or a component accessing others. To have common means for different objectives is of high significance in order to allow a proper standardization, which in turn is of paramount importance for such a crucial concern as trust.

This can be achieved by having an instance that is known to be trustworthy, from which trust in other elements can be granted through a proper coordination protocol. Trustworthiness is a delicate property that can only be adjudicated to some entity by an origin distinct from the target for trustworthiness. The genesis of initial trust within a collective, its maintenance and propagation are key. Trust can propagate hierarchically, where the root is some first entity that is assumed or certified to be trustworthy and that declares or denies trustworthiness for the lower entities. Or the initial trust arises from a consensus procedure within a collective. The abstraction of any entity, a device, software or mechanism, from which the trust originates initially before being propagated as a service, can be referred to as Trust Anchor. As one step towards a viable Trust Anchor for an universal trustworthiness framework, the work at hand investigates different architecture approaches for such a trust enabling service via DLT.

2 Related work

No concept for a service, which could enable a secured CtC communication between two or more cores of one or more operators is available in the 5G mobile communications standard. One of the reasons therefor is that solutions and systems in 5G are still very static, e.g., in respect to their geographical location. With the upcoming concept of private mobile networks in the 5G standard, new fields of application are introduced. One of these fields is the idea to have spatially dynamic networks, which are able to roam in order to fulfill their demands, e.g., in an agricultural surrounding. On the new path of scientific research to a standardized 6G also this category of private mobile networks, which can be referred to as nomadic, should be investigated. It is hence necessary to analyze new requirements for communication between distinct mobile networks.

To be able to establish a reliable and secure communication across all participating private mobile networks, one of the major tasks for CtC communication is that all aligned parties have to be unalterably traceable and trust each other to only behave as agreed. A first solution for that was introduced in [7]. The work at hand now focuses on different approaches to enable a trusted infrastructure by using DLT, e.g., blockchain concepts. As a concept for achieving trust in mobile 5G and Beyond 5G (B5G) networks, [8] identified different trust dimensions like applications or communication, which need to be considered by a trust management system. Security measures need to establish and maintain every of those dimensions. The authors sum up, that with blockchain or trusted platforms, new issues will come up, by which it is necessary to achieve a balance between a certain trust level in a network and the costs to achieve it. In [9], prominent trust approaches were analyzed. Additionally, a pre-standardization approach for trust and reputation models for 6G networks is suggested. Four modules for accomplishing key actions in trust models and fulfilling the requirements and KPIs by using new technologies are introduced.

The authors of [10] investigated into security and privacy issues of future 6G networks and uncovered issues related to new technologies. To establish TaaS in upcoming 6G mobile networks, challenges and opportunities are discussed in [11]. By reviewing the usage of blockchain in combination with future technologies, an outlined number of possible solutions was defined. It is assumed, that blockchain will support the growth of 6G and is able to face security threats. As in [8], the authors of [11] too addressed the necessary balance of security and performance in future 6G networks.

In [12], blockchain technologies are also explored in the setting of future 6G wireless networks. The authors investigated the increasing attention on data security especially for AI applications. Their simulation results have shown that blockchain could enable resistance against tampering of data in a mobile network. As in [12], the utilization of DLTs in 6G networks was also discussed in [13]. The authors identified emerging directions by using blockchain in 6G. One of those are Trust-based Secure Networks, which shall enable trust, security and privacy in future mobile networks through privacy compliance and access control. Also the application of AI methods like federated learning could be used for enabling trusted infrastructures.

3 Preceding work

The previously published work [7] discussed three areas: CtC communication, the newly introduced core service SASF for coexistent spectrum allocation and the newly introduced core service TTF for creating trust in the SASF's transactions. In the following a brief digest is provided.

Simplified, the scenario is that one full infrastructure (called *licensee*) has spectrum licensed in a coverage area. Another full infrastructure (called *applicant*) approaches spatially and initiates a negotiation about getting spectrum granted for operation.

3.1 Architecture

The SASF contains the logic behind controlling the whole procedure. This requests spectrum from the other core, resolves a decision for a request and collaborates with the local Radio Resource Management (RRM) to either acquire spectrum for sharing, restrict the own utilized range or configures to operate in the granted frequency area.

So the applicant's SASF would be configured to request a certain amount of radio resources from the connected SASF. The latter performs the negotiation and afterwards grants the RRM spectrum and capacity according to the result. During operation, requests for radio resources arrive at the licensee's SASF. In this system, a decision has to be made how much resources and what spectrum precisely can be dispensed.

The TTF is responsible for creating trustworthiness in the negotiation and the achieved agreement, but also the measurements during operation. It internally implements a DLT-based storage mechanic to immutably store data. During negotiation, the SASF sends its messages to or receives them from the TTF, which forwards them in respectively opposing direction while concurrently logging. During succeeding operation, DLT is utilized to store the measurement values.

As common, internally within one core, all functions communicate via a REST-API. As far as this level is concerned, it is sufficient for functioning to insert the two new core functions, including their specified API and they are able to exchange information according to a simple application level protocol representing the access methods — until the CtC communication becomes involved. How the integration of the SASF and TTF into given cores may look like is depicted in Fig. 1.

3.2 Novel functions and required capabilities

Looking deeper into the system, changes on several points are required to enable a mechanism as striven for. Mostly because dedicated CtC communication is not envisaged in current systems. Nonetheless is such an interaction

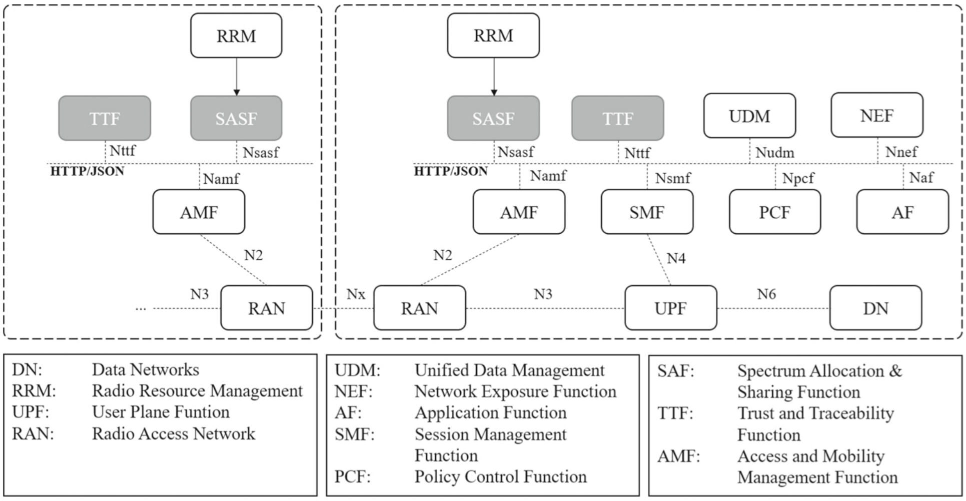
RRM
TTF
SASF
Nttf
Nsasf
HTTP/JSON
Namf
AMF
N2
N3
...
RAN
Nx
RRM
SASF
TTF
UDM
NEF
Nsasf
Nttf
Nudm
Nnef
HTTP/JSON
Namf
Nsmf
Npcf
Naf
AMF
SMF
PCF
AF
N2
N4
RAN
N3
UPF
N6
DN
DN: Data Networks
RRM: Radio Resource Management
UPF: User Plane Funtion
RAN: Radio Access Network
UDM: Unified Data Management
NEF: Network Exposure Function
AF: Application Function
SMF: Session Management Function
PCF: Policy Control Function
SAF: Spectrum Allocation & Sharing Function
TTF: Trust and Traceability Function
AMF: Access and Mobility Management Function

Fig. 1 5G Service Based Architecture (SBA) Functions. The SASFs interact across the distinct cores. All such communication flows through the related TTF indispensable, since involved operations for proposed functionalities are clearly a management plane task and are thus proposed to be integrated as core function, whereas the designated usage scenario tackles to have fully independent infrastructures coexisting (depicted in Fig. 2).

This leads to the areas that need to be touched in addition to solely implementing the novel core functions. The first obstacle is that a nomadic network in the very first instance requires a physical interconnection with the stationary network to enable a fundamental reachability. For this, evidently a radio link is desired. Already the physical conditions for this link — like establishing first contact — are not trivial to solve and on top a protocol is necessary that allows to exchange messages dedicated to CtC communication.

Means for such a communication is yet to be elaborated in more detail in future work, as the previously published work focused on the DLT technology to create the trustworthiness. Hence, this manuscript refers to it as "Nx interface" as a placeholder. Depending on how it will exactly be implemented, this *Nx* can have logical contact points with N1, N2, Next Generation Application Protocol (NGAP) and of course how the 5G gNodeB Basestation (gNB) operates the radio link. Possible solution strategies are also discussed in a later Section of the preceding publication.

Candidates are Protocol-regarding implications on *Xn Application Protocol (XnAP)* or extending the Self-Backhauling mechanism that is already in use for direct over the air communication between base stations albeit being actually only applicable within one network operator and with one core. The Xn interface describes a logical point-to-point link between two NG-RAN nodes in 3GPP 5G networks (3GPP TS 38.420). It provides the infrastructure for the XnAP (3GPP TS 38.423) and enables Control Plane functions for User Equipment (UE) mobility management, dual connectivity or resource coordination, as well as User Plane functions for data transfer or fast retransmission.

The second obstacle is then the logical communication, which goes further than only introducing a protocol to use over the Nx interface. The key topic here is that messages have to be transported from one core function to another inside a distinct core, which consequently transit over the RAN. When sticking to 5G's system architecture, various elements in sequence would be involved in such a communication. Hence, the protocols involved between such elements (i.e., NGAP) and the interface specifications (i.e., N2) have to be extended for carrying required information. Equally the elements themselves (AMF, gNB) require modifications to their logic to parse respective messages correctly.

3.3 Logical and physical communication

Communication from inside the core to outside is not part of the 5G design. The only other component a core talks to is the RAN. And for this, one singular message gate exists: The AMF, which connects to the base station over the N2 interface. We expect that this is the appropriate point to add the novel CtC capability in. CtC messages would then tunnel through the AMF, go over the N2 and are transmitted over the air between base stations. As said, transmitting messages over such sequence is not part of the current core design. Actually, the AMF is responsible for (simplified spoken) handling tasks regarding UE connection and similarly, over N2, configuration data is handed to the RAN regarding operating the radio communication.

But the idea to use the AMF as a forwarding gateway is not totally alien. On N2 as the reference point between the base station and the core, the NGAP is in use to support both UE and non UE related services and to create a decoupling between AMF and other services talking beyond it. For this purpose, NGAP supports information that the AMF is just responsible to relay (which is for instance done between the 5G-AN and the SMF).

One major challenge is to first establish a physical connection between two independent mobile communications networks. Though private networks are defined in 5G, the communication between two or more cores respectively two

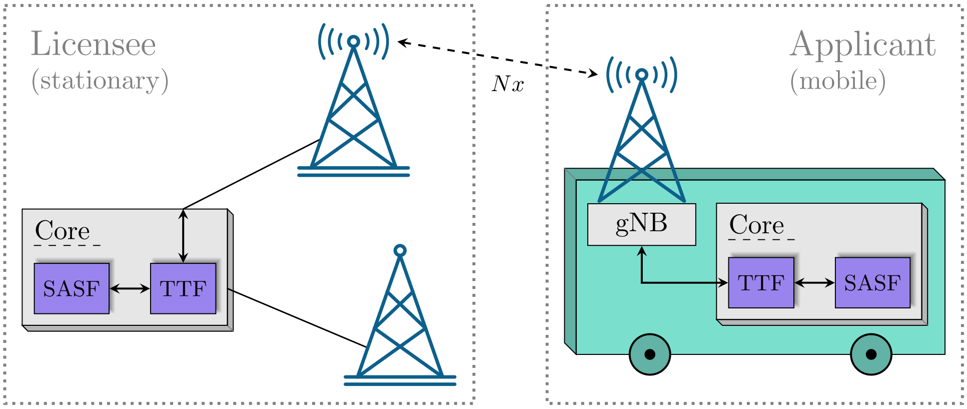

Fig. 2 Direct radio connection between independent networks for Core-to-Core communication

or more RAN of distinct full infrastructures is not part of the standard. 6G needs to provide a solution for this existing gap to enable the functionality of spectrum sharing or spectrum licensing between different operators within the coverage of a local network. The most crucial obstacle here is that an initial radio connection means already utilizing spectrum, which will only be the object of the following negotiation about whether the applicant is granted permission to use it.

More detail on the issue as well as 5G Private Networks is provide within the preceding publication [7].

4 Trust & Traceability Function (TTF)

The TTF provides the service of establishing traceable and verifiable communication links to the core of a peer network, as well as a trusted logging facility. Its current design is intended to integrate into the context of the SBA in B5G cores. The endpoints of a link within a session share a common view of a ledger instance, where the trail of exchanged data is stored in a tamper-proof manner. The majority of the described mechanisms originate from existing DLTs, whereas a set of adaptions need to be considered in order to apply DLT specific features to the application field of CtC communication in 6G networks. The most influential adaption comes from the fact that the communication link usually is set up between two points instead of distributing the ledger to an arbitrarily sized set of network participants. For this case, no consensus algorithm for decentralized peer-to-peer networks is deployed, but rather a protocol for establishing a measurement of trust between two entities is discussed. The architecture discussed in this section is primarily aimed to support a CtC scenario, which maps to a collaborative negotiation process as described in [7]. Core internal logging applications as in Phase 2) depict a special case, where no peer is present.

We describe a ledger as a state machine, where the updated state depends on newly issued transactions and the previous state. The following terms are based on the notation used in the Ethereum Yellow Paper [14]:

$$\sigma_{i+1} = \Pi(\sigma_i, B_{i+1})$$

A state is denoted by σ. B depicts a Block, which contains one or several transactions and Π describes the state transition function. This kind of describing a ledger allows for defining complex business logic, which can be executed on top of the block storage, as it is used for example on the Ethereum Blockchain. However, the application of using the ledger as an audit trail can be seen as a special case with a simple, transparent state transition function, i.e., the ledger state directly describes the accumulation of all submitted transactions. While the requirements of such kinds of applications already can be met by transaction based DLT without a dedicated state defined by additional application logic, the concept is nonetheless introduced at this point in order to ensure compatibility for further application fields. For example to enable the implementation of application logic within the TTF for the execution of Smart Contracts, like they are described for example in [15]. This way, for instance, it would be possible to likewise record and attest the logic of a service as opposed to merely its result.

4.1 Architecture

The internal architecture of the TTF, which is discussed in this section, is depicted in Fig. 3. The TTF requires two kinds of externally accessible interfaces. The Service API exposes the services necessary for the submission of messages to an active session and retrieving the current state of the ledger. Core functions consuming the service submit a session by sending a payload along with metadata necessary for addressing the session and the respective service within a session. The TTF then initiates a procedure to append the payload to the ledger. The Transport API provides the logical link to the peer network via the respective forwarding mechanisms of the AMF and the RAN.

The highest management layer within the TTF, the Ledger Management, is responsible for the instantiation and addressing of different Audit Sessions. Active Audit Sessions are described by Ledger Instances, whereas the message trail of terminated sessions is stored and retrievable on demand. When a new session is established, the Ledger Management communicates with the peer TTF to build

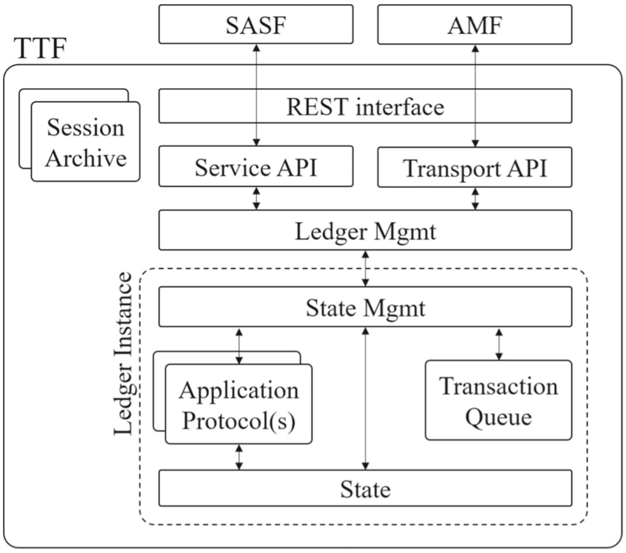

Fig. 3 Functional elements of the Trust & Traceability Function

a Genesis Block, which describes the initial state of the session.

A particular Ledger Instance is administrated by a State Management Layer. It maintains a logical link to the State Management of the respective session within the peer TTF and orchestrates state transitions resulting from newly issued transactions on the ledger. It furthermore checks the signatures included in message payloads for validity and applies read and write permissions.

Within a Ledger Instance, the State Management accesses modules describing the current state of the session, in particular a TX queue, a state instance and application protocols. The TX queue holds newly issued transactions to the ledger with pending consensus status, which means the TTF is waiting for a response from the peer with the consent to append the transaction to the ledger. The application protocols process submitted transactions and compute a resulting ledger state. This means, the application protocols define the state transition function of the ledger. The state instance depicts the storage of all previously appended blocks as well as the current version of the ledger state.

Since the information written to the ledger does not originate from the TTF itself, but rather comes from arbitrary core functions consuming the service (in our case the SASF), the framework requires a measure to verify the identity of transaction authors. This can be provided by using digital signatures, i.e., via ECDSA [16]. By requiring every transaction author to add a digital signature to the payload, the system ensures non-repudiation (an author can not deny having created a signature before) and data integrity on transaction level (manipulations of the payload result in an invalid signature).

4.2 Procedures

On establishing a new session, both peers need to exchange information to build a Genesis Block. It depicts the first entry on the common ledger and is the starting point of the consensus between the peers. It contains information necessary for maintaining consensus on future ledger entries. Since the block needs to match on both sides of the communication link, a protocol for its format is required beforehand, for example the following order:

- Metadata from the requesting peer
- Metadata from the responding peer
- Settings and protocol metadata for the ledger

Peer related metadata include public keys from authors on the respective side of the link, which can also be used as account identifiers of the authors on the ledger, and the session ID corresponding to the ledger instance on each peer, respectively. The settings section also may include information on which application protocols are used in the session.

After the setup of a ledger instance, new blocks are appended to the ledger when a core function sends a respective request containing the signed payload to the TTF (here: P1, for *on Peer-1*). Via the session ID, the ledger management forwards the payload to the ledger instance, where the state transition is calculated by the application protocols, resulting in a block to be appended to the ledger. The block contains the signed transaction(s) and a header with at least the following information:

- The hash over the previous block on the ledger (Block Hash): This builds a hash chain over all submitted blocks, so any manipulation on a block would require to manipulate all succeeding blocks on the ledger.
- A hash over each transaction (TX Hash): Can be used to verify the inclusion of a transaction into a block, without having to access the whole block, i.e., other transactions in the same block.
- A value identifying the state resulting from the application of transactions (State Root): In DLT, this can be implemented by using a Merkle Patricia Tree for storing the state, where a hash tree structure is built upon all storage items, resulting in a single root hash, which completely describes the sum of all storage items [14]. In the TTF, the application protocols would set storage items in the state according to newly issued transactions and the root hash of the Merkle Patricia Tree would be included in the block header.

After the state transition is calculated, the transactions are stored at the TX queue and sent to the peer TTF (here: P2, for *on Peer-2*) in a Block Inclusion Request, along with the metadata from the new block header. Using the Block Hash, P2 validates that P1 is building on top of the same view of the ledger, i.e., the previous block matches on each side. By computing the state transition via the local application protocols and comparing the State Root, P2 validates that the ledger state matches the new state, calculated at P1. On success, P2 appends the block to its ledger instance and sends a Successful Outcome Response to P1, which then also appends the block to the ledger.

This procedure also allows to handle the special case of concurrent Inclusion Requests, where both sides nearly simultaneously send Block Inclusion Requests to each other. This situation is detected at each side when it receives a Block Inclusion Request before receiving a response to its own request. For this case, the order in which transactions from both peers are applied to the ledger state must be defined by a protocol beforehand. The procedure is then terminated by an additional message roundtrip to ensure the accumulative state transition of all new transactions is valid to both peers.

The application built on top of the services of the TTF not only relies on submitting data to the ledger, but also on reading information from it, especially in the use case of building a traceable communication medium, i.e., for radio resource allocation negotiations. In the proposed framework, we identify three ways of retrieving information from a ledger instance, provided that the respective read access is granted:

- *Block Query*: The client retrieves one or several blocks from the ledger instance
- *State Query*: The client queries storage items from the current state of the ledger instance
- *Application Protocol API*: Depending on the read access to the state, application protocols also might expose an API to retrieve an application-specific pre-processed view on the ledger state

4.3 Communication audit trail as main application protocol

As stated earlier, the application of using the ledger as a common storage for the exchange of information depicts a special case for the use of the TTF. The use case of spectrum allocation discussed in this work is an example hereof since the application logic mainly takes place at the level of SASF instances, using the TTF only as a transport layer for exchanged information. The application protocol used in the ledger instance therefore fulfills the task of storing the list of transactions directly in the state, which can then be queried by the SASF, where they are interpreted within the spectrum allocation context.

5 Architecture approaches for trust anchors

The archiving of terminated sessions in the TTFs is crucial for establishing trust, because it allows peers to compare both views on the ledger in the aftermath and detect tampering of the audit trail. However, this procedure alone does not provide a proof of which side has manipulated the data. For this purpose, a framework for the distribution of block headers to third parties is required. The distribution or publishing needs to take place at a point in time where both peers maintain consensus on the ledger, i.e., at nominal operation. This provides mechanisms for future verification to detect on which side the data has been manipulated, without having to share confidential data written to the ledger. The third party should depict a facility which all actors trust to be neutral like a regulative authority or alternatively a distributed ledger maintained by a sufficiently high number of independent actors.

5.1 Ledger manipulations

For an intentional collusion, involved parties would indeed be able to manipulate the Ledger. If this manipulation is done after the headers have been published, the manipulation can be detected. But there is no guarantee anymore that it can be rolled back since both peers have modified their Ledger instance. A mutual manipulation happening before the headers are published cannot be detected. However, both parties making agreements on the Ledger would bypass its actual purpose of building trust, where in this case the trust has already been established.

5.2 Header distribution architecture

In the following section, the third party, collecting the header data from TTFs, is abstracted by the term Trust Anchor, independent from the technology implementing it. This discussion provides an expansion of the preceding work in [7].

The Trust Anchor is responsible for maintaining only data necessary for the verification of the consensus between TTFs on a Ledger. It is expected to hold a verifiable consistency over time, while the private data (transactions, state) is still stored at the TTFs locally. This separation of the Trust Anchor architecture from the private domain of TTFs allows it to remain open and transparent. This makes it less complex to include mechanisms for verification and resilience, since the access to the data stored at the Trust Anchor does not need to be restricted or regulated.

The verification data that is uploaded to the Trust Anchor describes a checkpoint for the latest verifiable consensus between TTFs, so it should also be possible to update it over time. When the Trust Anchor receives data from TTFs, it needs to verify that the data indeed describes a consensus between peers. This can be done either by both peers submitting the same set of data to the Trust Anchor independently, or by uploading a version of the data which has been signed by both TTFs. The former approach does not rely on a digital signature algorithm, as the second approach does, but it requires the Trust Anchor to buffer the data received by one peer and wait for the submission of the exact same data from the other peer, doubling the network load.

Multiple approaches can be compared regarding the interface between the TTFs and the Trust Anchor. In all of the cases, the information transmitted to the Trust Anchor should be usable for verifying the consistency of a Ledger Instance on both peers. In a first option, all header data from every block of a Ledger can be transmitted to the Trust Anchor. This procedure could be integrated into the consensus procedure between TTFs, when a new block is appended, but leads to a high network load and increased storage demands for the Trust Anchor. On the other hand, the existence of a particular block in the Ledger can be verified directly in any case. A second approach would be to only transmit header data of specific blocks. This procedure needs to be initiated asynchronously by one of the peers and reduces network load and storage requirements alike. The blocks of which header data are transmitted can be considered checkpoints of the Ledger, since the header data of a particular block includes dependencies on all previous blocks. To verify the existence of a block in the Ledger with this approach, information from all blocks between the one to be verified and the next checkpoint is necessary.

Regarding the architectural design of the Trust Anchor itself, several approaches are possible and may be combined in some ways. The approaches discussed in the following consider two different branches of technology, i.e., DLT and Verifiable Databases.

DLT represents a set of technologies, where multiple independently acting entities maintain consensus on a common set of data, with its most prominent form being the Blockchain. This property in combination with the various required technologies like hashing, cryptography or distributed consensus algorithms ensures resiliency and non-repudiation by design. A survey on several DLT directions and implementations is carried out in [17].

A Verifiable Database or Verifiable Ledger, according to [18], maintains an append-only storage, where data records can be added but not modified or deleted on the lowest level. It also provides the means for clients to computationally verify that a queried record has not been tampered with and that the overall storage of the database remains consistent.

Standard Centralized Database For the perspective of a single country, a standard database (SQL, NoSQL) might be deployed by a regulative authority. In this case, the trust purely originates from societal acknowledgment and legislation. By strongly regulating the external access to data, privacy is preserved by the architecture itself, which might allow to associate further metadata with pure verification data and open up new possibilities of centralized data processing. This option requires the least resources regarding the overall storage footprint. On the other hand, there are only limited ways to validate the correct behavior of the Trust Anchor. Limited trust in regulative authorities of other countries in the geopolitical context might affect the trustworthiness of the overall system in the case of roaming.

Verifiable Centralized Ledgers By deploying a Verifiable Ledger on a regional basis instead of a private database, trust is detached from an authority since it relies on the transparency of the Trust Anchor and the exposed verification mechanisms. Any institution with the necessary resources may provide such a Ledger. An approach to automate and globalize the consistency verification of all Verifiable Ledgers can be to deploy a Distributed Ledger with all Logs and additional monitoring entities (Auditors) as nodes. Logs may submit their root hashes to this Ledger periodically, which are then used by Auditors to validate the consistency of the Logs. This option and the first one, i.e., all in essence centralized approaches, have in common that additional measures for the rollback of detected manipulations of the Ledgers have to be taken, e.g., continuous independent backups need to be maintained for resiliency.

Global Distributed Ledger In a fully decentralized approach, all TTFs may additionally run nodes of a single Distributed Ledger, which provides the global Trust Anchor storage for header validation. This approach is architecturally simple and a new TTF may easily be added to the public Trust Anchor network. This Global Distributed Ledger also can provide reputation mechanisms between TTFs, e.g., via the provision of Smart Contracts. One disadvantage of such a global solution is clearly the large amount of data that must be processed and stored at each node participating in the network.

Cascaded Distributed Ledgers A possibility to tackle the scalability issues of big public DLT networks as described in the previous part can be the cascading of smaller Distributed Ledgers by linking them via a global relay chain. This means that a local group of TTFs runs a Distributed Ledger as local Trust Anchor. The set of local Trust Anchor networks is then connected to an additional Blockchain network via specific nodes, which provides a globally validated overall state. A prominent example of such a Blockchain of Blockchains is the Polkadot network [19]. This architecture approach requires a complex consensus procedure to track the correct behavior of the nodes, which link the local Ledgers to the relay chain. The separation into multiple local Distributed Ledgers enables the local networks for independent implementations and updates, as long as the interface towards the relay chain matches the standard protocol.

5.3 Trusted infrastructure

The discussions on the infrastructure carried out in this work focused on the establishment of a trustworthy consensus between two independent network infrastructures. The verifiable operation of the trust building framework however can also be exposed to third party applications, thereby propagating the trustworthiness of the underlying architecture to external modules, e.g., for verifiable accountability. This leads to the concept of TaaS provided by the infrastructure operator to third party applications running on top of the 6G network. Here not the infrastructure is the entity seeking trustworthiness but the established basis for others, i.e., acts as Trust Anchor for devices and applications that utilize it.

The fundamental idea, i.e., the hierarchical concept behind the feature from [7] and a Trusted Infrastructure are very similar, whereas the major difference is that the root of the tree is represented by a different element. Or differently put, when combining both scenarios, the initial trustworthiness originates on a different level. The authors believe that a proper framework can serve both scenarios equally, which is object for upcoming development.

6 Conclusion and outlook

The preceding publication [7] introduced a mechanism to enable multiple 6G infrastructures to coexist in a shared frequency range with respect to licensing issues as it proposed core functions for negotiating and logging the coexistent operation. Major focus was put on being able to trust in historic logging data because the frequency occupation aspect is a question of licensing and hence a legal issue. The most in-depth focus was afforded to a core function, which enables trustworthiness using DLT. Mainly missing was a procedure for the distribution of the raised meta data. The manuscript at hand thus investigated on viable architecture approaches for Trust Anchors for the purpose of verification of audit trail integrity.

The ultimate goal of our endeavors is a versatile framework that can be utilized in various use-cases, even to enable slightly distinct scenarios in conjunction as briefly addressed in Section 5.3. Intended future work hence covers to further elaborate a header distribution mechanism, a more thorough concept for a Trust Anchor and finally a comprehensive implementation of the developed tools into a framework that can be rolled-out as a modular service into networks.

Acknowledgements The authors acknowledge the financial support by the German *Federal Ministry for Education and Research (BMBF)* within the project ≫Open6GHub≪ {16KISK003K & 16KISK004}. A related manuscript was priorly published with the IEEE as [7].

Author contribution Conceptualization: D.K. and B.V. Methodology: D.K. and B.V. Investigation: D.K., B.V. and D.L. Writing—original draft preparation: D.K., B.V. and D.L. Writing—review and editing: D.K., B.V. and D.L. Visualization: D.K. and B.V. Supervision: D.K. and H.D.S. Project administration: D.K. Funding and project acquisition: D.K. and H.D.S. All authors have read and agreed to the published version of the manuscript.

Funding Open Access funding enabled and organized by Projekt DEAL.

Availability of data and materials Not applicable

Code availability Not applicable

Declarations

Ethics approval Not applicable

Consent to participate Not applicable

Consent for publication Not applicable

Competing interests The authors declare no competing interests.

References

1. Bless R, Bloessl B, Hollick M, Corici M, Karl H, Krummacker D, Lindenschmitt D, Schotten H, Wimmer L (2022) Dynamic network (re-)configuration across time, scope, and structure. In: 2022 Joint european conference on networks and communications & 6G Summit (euCNC/6g Summit). IEEE Xplore, IEEE, pp 547–552
2. Krummacker D, Fischer C, Munoz Y, Schotten H (2022) Organic & dynamic infrastructure: getting ready for 6G. In: Mobile communication-technologies and applications; 26th ITG-symposium. ITG, IEEE Xplore, VDE, vol 304
3. Corici M, Troudt E, Chakraborty P, Magedanz T (2021) An ultra-flexible software architecture concept for 6g core networks. In: 2021 IEEE 4th 5g world forum (5GWF), pp 400–405. https://doi.org/10.1109/5GWF52925.2021.00077
4. Krummacker D, Schotten H (2022) Status-preserving, seamless relocation of processes in orchestrated networks such as organic 6G. In: 2022 IEEE 5th international conference on industrial cyber-physical systems (ICPS), pp 1–8. IEEE Xplore, IEEE. https://doi.org/10.1109/ICPS51978.2022.9816860
5. Corici M, Troudt E, Magedanz T, Schotten H (2022) Organic 6g networks: decomplexification of software-based core networks. In: 2022 Joint european conference on networks and communications & 6G Summit (EuCNC/6G Summit). IEEE, pp 541–546
6. Krummacker D, Fischer C, Veith B, Schotten HD (2022) 6G core-architecture – approaches for enhancing flexibility across control and user plane
7. Krummacker D, Veith B, Lindenschmitt D, Schotten HD (2022) Radio resource sharing in 6G private networks: trustworthy spectrum allocation for coexistence through DLT as core function. In: 2022 1st International conference on 6G networking (6GNet), pp 1–8. IEEE Xplore, IEEE. https://doi.org/10.1109/6GNet54646.2022.9830407

8. Benzaïd C, Taleb T, Farooqi MZ (2021) Trust in 5g and beyond networks. IEEE Netw 35(3):212–222. https://doi.org/10.1109/MNET.011.2000508
9. Jorquera Valero JM, Sánchez Sánchez PM, Gil Pérez M., Huertas Celdrán A, Martínez Pérez G (2022) Toward pre-standardization of reputation-based trust models beyond 5g. Comput Standards Inter, vol 81. https://doi.org/10.1016/j.csi.2021.103596
10. Wang M, Zhu T, Zhang T, Zhang J, Yu S, Zhou W (2020) Security and privacy in 6g networks: new areas and new challenges. Digital Commun Netw 6(3):281–291. https://doi.org/10.1016/j.dcan.2020.07.003
11. Nguyen T, Tran N, Loven L, Partala J, Kechadi M-T, Pirttikangas S (2020) Privacy-aware blockchain innovation for 6g: challenges and opportunities. In: 2020 2nd 6G wireless summit (6G SUMMIT), pp 1–5. https://doi.org/10.1109/6GSUMMIT49458.2020.9083832
12. Li W, Su Z, Li R, Zhang K, Wang Y (2020) Blockchain-based data security for artificial intelligence applications in 6g networks. IEEE Netw 34(6):31–37. https://doi.org/10.1109/MNET.021.1900629
13. Kalla A, De Alwis C, Gur G, Gochhayat SP, Liyanage M, Porambage P (2022) Emerging directions for blockchainized 6g. IEEE Consumer Electr Mag. https://doi.org/10.1109/MCE.2022.3164530
14. Wood G (2014) Ethereum: a secure decentralised generalised transaction ledger, Berlin version b8ffc51 - 2022-02-21. Accessed 25 Feb 2022
15. Alharby M, Van Moorsel A (2017) Blockchain based smart contracts : a systematic mapping study, pp 125–140. https://doi.org/10.5121/csit.2017.71011
16. Al-Zubaidie M, Zhang Z, Zhang J (2019) Efficient and secure ecdsa algorithm and its applications: a survey. Int J Commun Netw Inf Secur, vol 11
17. Antal C, Cioara T, Anghel I, Antal M, Salomie I (2021) Distributed ledger technology review and decentralized applications development guidelines. Future Internet, vol 13(3). https://doi.org/10.3390/fi13030062
18. Zhang M, Xie Z, Yue C, Zhong Z (2020) Spitz, vol 13. https://doi.org/10.14778/3415478.3415567
19. Wood G Technology — Polkadot. Online. https://polkadot.network/technology/. Accessed 18 Aug 2022