BRL Working Paper Series No. 18

The DibiChain protocol: Privacy-preserving discovery and exchange of supply chain information

Elias Strehle [1,*], Martin Maurer [2]

[1] Blockchain Research Lab, Max-Brauer-Allee 46, 22765 Hamburg, Germany
[2] CHAINSTEP, Max-Brauer-Allee 46, 22765 Hamburg, Germany
[*] Correspondence: strehle@blockchainresearchlab.org

Published: 10 Feb 2021

Abstract: Connecting and exchanging information across organizations becomes increasingly important as supply chains become more complex and expectations with regard to sustainability, transparency and resilience increase. At the same time, organizations are adamant about protecting any competitive advantage which derives from private information about, for example, supplier networks, available inventory or production processes. Technology aimed at enabling information exchange within and across supply chains must therefore ensure high degrees of privacy and control over private information. In light of this, we specify the DibiChain protocol for the discovery and exchange of supply chain information. The protocol prioritizes data minimization in shared data stores, avoidance of persistent user identifiers and anonymous communication with minimal intermediation. We further outline how the DibiChain protocol can serve as the foundation for privacy-preserving supply chain applications, including an anonymous discovery service for GS1 EPCIS event data.

Keywords: Privacy; Anonymity; Supply chain; Distributed ledger technology; Blockchain

1 Introduction

Supply chain relationships are characterized by the paradox of coopetition: Actors try to reap the benefits of cooperation and coordination without jeopardizing their competitive advantage (Wilhelm and Sydow, 2018). The fear of leaking valuable private information may cause actors to refuse to cooperate even if cooperation would benefit everyone (Zhang, 2009; Hackius et al., 2019). Addressing this source of inefficiency seems especially worthwhile in the context of transitioning to a more sustainable, circular economy (Geng et al., 2019) as lack of information sharing in supply chains and the resulting risks, e.g., when refurbishing or recycling products with unknown material composition, can be an obstacle to circular policies and business models (Winans et al., 2017).

We propose the DibiChain protocol as a potential solution for situations in which organizations (or humans) want to connect, discover and exchange information while preserving a high degree of anonymity and unlinkability. The protocol builds on a shared data store, for example a distributed ledger, and a public or private Tor network (Dingledine et al., 2004). The shared data store lets users

The authors gratefully acknowledge funding through the joint research project DIBICHAIN within the framework of the ReziProK program, which is funded by Germany's Federal Ministry of Education (funding reference number: 033R241). Apart from the provision of funding, the funding source had no involvement in this study.

publish anonymous pointers, called addresses, to reference private information without leaking it. It further lets users connect these address via links to allow other users to discover addresses connected to their own through a chain of links. Users can then send anonymous queries via the Tor network to request the private information referenced by an address. In this way, a direct information exchange becomes possible without giving up anonymity or channeling queries through a trusted third party; an implementation of the DibiChain protocol can provide the "missing links" between data silos within and across supply chains while ensuring a high degree of privacy.

The remainder of the paper is structured as follows. In Section 2 we discuss existing work dealing with the standardization and exchange of supply chain information. In Section 3 we describe the DibiChain protocol and some extensions. In Section 4 we outline an ecosystem for the discovery and exchange of supply chain information based on the DibiChain protocol. We further illustrate the practical implications by sketching a DibiChain-based discovery service for EPCIS visibility event data. In Section 5 we discuss the advantages and disadvantages of the DibiChain protocol in relation to alternative approaches. Section 6 concludes.

2 Related work

The non-profit organization GS1 publishes standards to promote the collection and exchange of information within and across supply chains, including a specification of standardized object identifiers (GS1, 2021), a core business vocabulary (GS1, 2017a), the EPCIS standard for visibility event data (GS1, 2016a) and a global traceability standard (GS1, 2017b). Together, these standards can serve as the foundation for instance-level traceability of the "what, when, where and why" of supply chain events such the manufacturing, packaging, shipping and receiving of goods (GS1, 2016b, Section 2.2). Traceability has many potential applications, including curbing illegal practices, improving sustainability performance, increasing operational efficiency, enhancing supply chain management and sensing market forces and trends (Hastig and Sodhi, 2020, Figure 1).

A key challenge for traceability is how to connect information across companies. The pragmatic approach of "one up, one down" tracing may not yield sufficient transparency and leave supply chains vulnerable to fraud and low quality of ingredients (Pearson et al., 2019). More comprehensive approaches, however, face the "data discovery problem" of finding out which companies are connected by a chain of events, establishing trust between companies without a direct business relationship and transferring data once connection and trust have been established (GS1, 2017b, p. 36).

In recent years, blockchain technology or—more generally—distributed ledger technology (DLT) has been discussed as a promising foundation to solve the data discovery problem within supply chains (Hewett et al., 2019). DLT is an umbrella term for technologies that enable a network of independent parties to operate an ordered, persistent and tamper-evident ledger of cryptographically-validated records (Rauchs et al., 2018).

Tröger et al. (2018) propose a blockchain-based discovery service for EPCIS visibility event data. The proposed solution lets companies store sanitized EPCIS event data and access policies pertaining to these data on a service provider's distributed ledger. During the sanitization process, the data are shortened and sensitive values are replaced with salted hashes. Connections between the sanitized event data on the ledger are stored by a discovery service provider in a graph database.

Among the various other blockchain-based solutions for exchanging supply chain information, many store business information in shared data stores (often the ledger itself) to create transparency and harness the power of consensus-based validity checks and smart contracts (see, e.g., Abeyratne and Monfared, 2016; Biswas et al., 2017; Banker, 2018; Bettín-Díaz et al., 2018; Caro et al., 2018; Casado-Vara et al., 2018; Leng et al., 2018; Casino et al., 2019b; Dasaklis et al., 2019). In a proposal

for food safety monitoring, Tian (2018) tackles privacy requirements by only storing "key information" on the distributed ledger and relegating storage of detailed information to trusted third parties. Ferdousi et al. (2020) propose a distributed ledger for cattle supply chains. Their approach ensures pseudonymity but remains susceptible to correlation through persistent user IDs and a one-to-one mapping of business processes to publicly visible transactions.

3 The DibiChain protocol

In this section we describe the abstract DibiChain protocol. It allows users to reference and connect private information in a shared data store, which in turn allows other users to discover these references and request the corresponding information from its owner. While our focus is on supply chain applications, the protocol itself is generic and makes no assumptions on the format or content of the referenced private information.

We presuppose a persistent data store that can be accessed by all users. This could be a shared database, a private and permisioned distributed ledger or a public blockchain. We further presuppose a Tor network (Dingledine et al., 2004) which is used for anonymous communication both from user to data store and from user to user. Note that this need not be the public Tor network.[1] Instead, one can choose to deploy a dedicated, non-public Tor network which is only accessible to the DibiChain users—similar to non-public deployments of public blockchain technologies like Ethereum (Buterin, 2013). Using a dedicated network ensures that nodes do not handle Tor traffic unrelated to the DibiChain application.

The DibiChain data store knows two types of data objects: addresses and links. An address is identified by a public key and provides information on how users can anonymously contact the address owner. A link connects addresses on the DibiChain, providing context and allowing users to represent relationships between addresses.

A DibiChain *address* is a hidden service descriptor as specified in the Tor protocol.[2] It has an ID which is derived from a public key and contains a list of introduction points. The introduction points reference Tor nodes through which users can establish an anonymous communication channel with the address owner. A user creates a DibiChain address in the following way:

1. Create a public/private key pair.
2. Set up introduction points for the corresponding address ID according to the Tor protocol.
3. Create the hidden service descriptor and sign it with the private key.
4. Publish the signed hidden service descriptor as an address on the DibiChain data store.

A DibiChain *link* connects addresses. The link data object contains the following fields:

- `fromAddressIds`: A list of address IDs.
- `toAddressIds`: A list of address IDs.
- `toTags`: A list of strings, each representing a (hash of a) topic or keyword.
- `linkId`: The hash of all fields above.

[1] https://www.torproject.org.

[2] For an illustration of how hidden services work, see https://community.torproject.org/onion-services/overview. For the full technical specification, see https://github.com/torproject/torspec/blob/master/rend-spec-v3.txt.

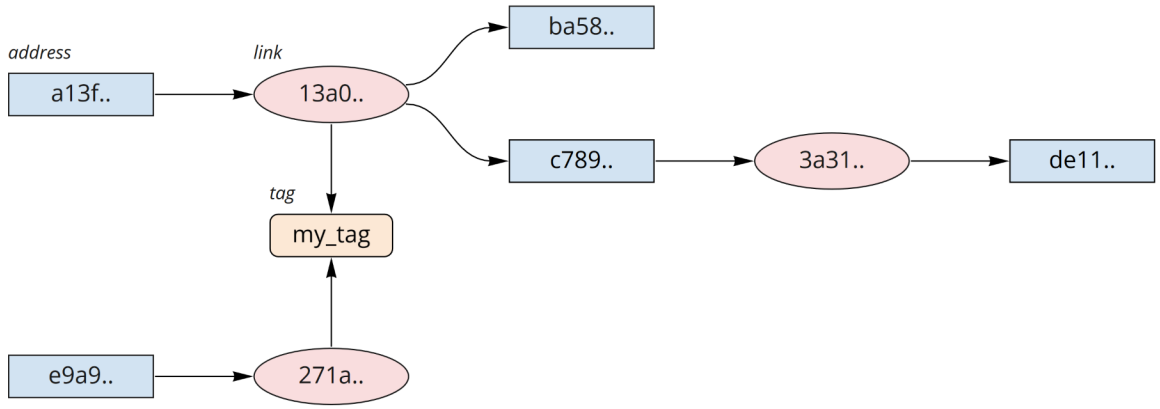

Figure 1: Illustration of connections on the DibiChain.

- `signatures`: A collection of cryptographic signatures of all fields above. Each signature corresponds to one address in `fromAddressIds` or `toAddressIds`. For example, a complete signature set for a link with two addresses in `fromAddressIds` and three addresses in `toAddressIds` contains five signatures.

A user creates a DibiChain link in the following way:

1. Create a partial link object with fields `fromAddressIds`, `toAddressIds` and `toTags`.
2. Compute the link ID by hashing the partial link object.
3. Add the link ID to the partial link object and sign it with the private keys corresponding to the referenced addresses.
4. Publish the signed link object on the DibiChain data store.

The purpose of `toAddressIds` and `toTags` is similar. Both represent a relationship with the addresses in `fromAddressIds`, allowing users to discover connected addresses by following a chain of links. The main difference is that the owner of an address in `toAddressIds` approves the link by signing it whereas tags have no concept of ownership and thus require no signature.

Figure 1 illustrates how addresses are connected on the DibiChain. We speak of a direct connection from address A to address B if there is a link where A is contained in `fromAddressIds` and B is contained in `toAddressIds`. We speak of an indirect connection between address A and address B if there is a tag T for which there are (1) a link where T is contained in `toTags` and A is contained in `fromAddressIds` and (2) a link where T is contained in `toTags` and B is contained in `fromAddressIds`. In Figure 1, for example, the addresses with IDs `a13f..` and `ba58..` have a direct connection and the addresses with IDs `a13f..` and `e9a9..` have an indirect connection. Direct references require coordination as every involved user must contribute one or more addresses and signatures to the link object prior to publication. Indirect references are easier to negotiate since users can pre-agree on a tag and then publish links independently from each other. At the same time, direct connections can foster additional trust by documenting the common intent of involved users. We discuss direct and indirect connections in the context of supply chain tracing in Section 4.1.

A downside of anonymity is lack of accountability, which can encourage misbehavior such as spamming or violation of use-case-specific rules. In Section 3.1 we describe how a trusted third party can provide accountability for activity on the DibiChain data store without sacrificing anonymity and unlinkability among users and between users and data store operators.

While a DibiChain data store could be operated by a single service provider, some use cases may

benefit from more sophisticated implementations with multiple operators and advanced security and trust guarantees, such as those provided by distributed ledger technology. In Section 3.2 we discuss how the DibiChain can benefit from the desirable features of distributed ledgers by adding capabilities for timestamping and notarization.

3.1 Achieving accountability

Anonymity can encourage misbehavior (Farkas et al., 2002). Malicious DibiChain users could spam addresses and links or even mount denial-of-service (DOS) attacks without the risk of being caught. Some user activity on the DibiChain may also violate contracts or laws. For example, a user could write personally identifying information into a link tag without the person's consent. For these reasons, it can be advantageous or even necessary to have a way of holding users accountable for undesired behavior. But how to establish accountability without giving up anonymity?

Balancing accountability and anonymity is especially challenging for networks which can be used without authorization, also called permissionless networks (Garzik, 2015). Solutions suggested and pursued for permissionless networks include proof of work (Dwork and Naor, 1992; Back, 1997; Nakamoto, 2008; Juels and Brainard, 2017), native tokens for transaction fees (Nakamoto, 2008; Buterin, 2013) and accountability delegates (Naylor et al., 2015; Lee et al., 2016; Ma et al., 2020). How those or other measures could be employed to secure a permissionless DibiChain implementation remains an open question. In this section, we propose instead an extension to the DibiChain protocol for a permissioned setting in which users register with a trusted third party called the *administrator*. Users can be held accountable by the administrator but remain anonymous to each other and to the operators of the data store and the Tor network.

To achieve this, each user maintains a hierarchical deterministic wallet (HDW). The concept of an HDW was introduced as an extension to the Bitcoin protocol (Wuille, 2012). Gutoski and Stebila (2015) and Das et al. (2019) extend the original implementation to reduce the negative impact of private key leakage. An HDW has a master public/private key pair from which its owner can derive an arbitrary number of child key pairs. Without knowledge of the master key, the child keys are unlinkable; in particular, an observer cannot determine whether two child public keys were derived from the same master key (Narayanan et al., 2016, pp. 80–81). Using an HDW for address creation therefore does not impair unlinkability on the DibiChain data store. At the same time, anyone who knows the master public key can determine whether a given public key was derived from it.

Accountability against the administrator is therefore achieved in the following way:

- At initialization, each user sets up an HDW by creating a master public/private key pair.
- The user shares the master public key and some identifying information (depending on the use case: a self-chosen user name, an e-mail address, a means of legal identification, etc.) with the administrator. The master private key is never shared.
- The user creates new addresses based on child key pairs derived via the HDW.
- When a data store operator receives a new address for publication, she sends the corresponding public key to the administrator. The administrator checks whether the public key has been derived from a known master public key and responds with an "OK" or "not OK." In the latter case, the operator refuses to publish the address on the data store.

In this way, addresses on the DibiChain remain anonymous and unlinkable to all participants except the administrator. In case of undesired behavior, the administrator can take action on his own (e.g., by blacklisting the misbehaving user) or reveal the identifying information (e.g., to law enforcement

in case of illegal activity).

It may be interesting to note that this extension also enables anonymous pay-per-use schemes for the DibiChain data store. The administrator can keep track of how many addresses and links are published by each user and either bill the users directly or forward the aggregated information to the data store operators. In this way, operators can offer usage-based payment models without learning which addresses belong to whom.

3.2 Adding capabilities for timestamping and notarization

The desirable features of DLT make it suitable for the implementation of timestamping and notarization services (Casino et al., 2019a). Publishing (a hash of) data on an immutable ledger creates a timestamped and tamper-proof copy which can lend external validity to claims about the data and assist in conflict resolution. We can extend the DibiChain protocol to offer trustworthy timestamping and notarization if the DibiChain data store is operated on a distributed ledger or the data store operators are trusted to provide and persist correct timestamps.

Timestamping capabilities are implemented by adding a `timestamp` field to the link object. In the interest of external validity and incentive alignment, the timestamp is not created by the user who publishes the link. Instead, it is added by the data store operators during publication. The timestamp field is *not* incorporated by the user when computing the link ID and the signatures as it is unknown before publication.

Notarization capabilities are implemented by adding a `dataHash` field to the link object. The field can contain arbitrary strings but should be a (salted) hash to ensure confidentiality. The content of the `dataHash` field is determined by the users who create the link and, unlike the timestamp, is incorporated when computing the link ID and the signatures. The data hash therefore inherits the following properties from the link object: It is (1) public among those who can access the data store, (2) timestamped by the data store operators, (3) confirmed by all involved users through cryptographic signatures and (4) tamper-proof (because changing it would invalidate the link ID and the signatures). These properties give data hashes on the DibiChain a high degree of trustworthiness.

4 Exchanging supply chain information with the DibiChain protocol

In this section we describe an implementation in which the DibiChain protocol powers an ecosystem for the secure and trustworthy exchange of supply chain information. To ensure privacy and data control, data storage is federated: Users store their own business-relevant data (or rely on a trusted third party to store the data for them). On top of the user-operated data stores, an implementation of the DibiChain protocol on a distributed ledger acts as a "discovery service" (GS1, 2017b, Section 4.3.3). The DibiChain data store is accessible to all users, allowing them to register business information via anonymous DibiChain addresses and connect information within or across local data stores via DibiChain links. It is run by a small number of operators who coordinate through the DLT's consensus mechanism. In addition, an administrator as described in Section 3.1 acts as a gatekeeper to the ecosystem. Figure 2 shows a sketch of the system architecture.

The DibiChain data store is implemented as a distributed ledger. Every operator maintains a full copy of the ledger and uses it to answer read requests from users. A fault-tolerant consensus algorithm ensures that operator copies of the ledger stay in sync and that the ledger tolerates errors or misbehavior from a minority of operators. Our prototype simulates four data store operators and implements the distributed ledger with crash fault tolerance (CFT) on Hyperledger Fabric (Androulaki et al., 2018). Stronger fault tolerance could be achieved by implementing a Byzantine fault tolerant (BFT) consensus algorithm (Lamport et al., 1982; Castro and Liskov, 1999; Cachin and Vukolić, 2017).

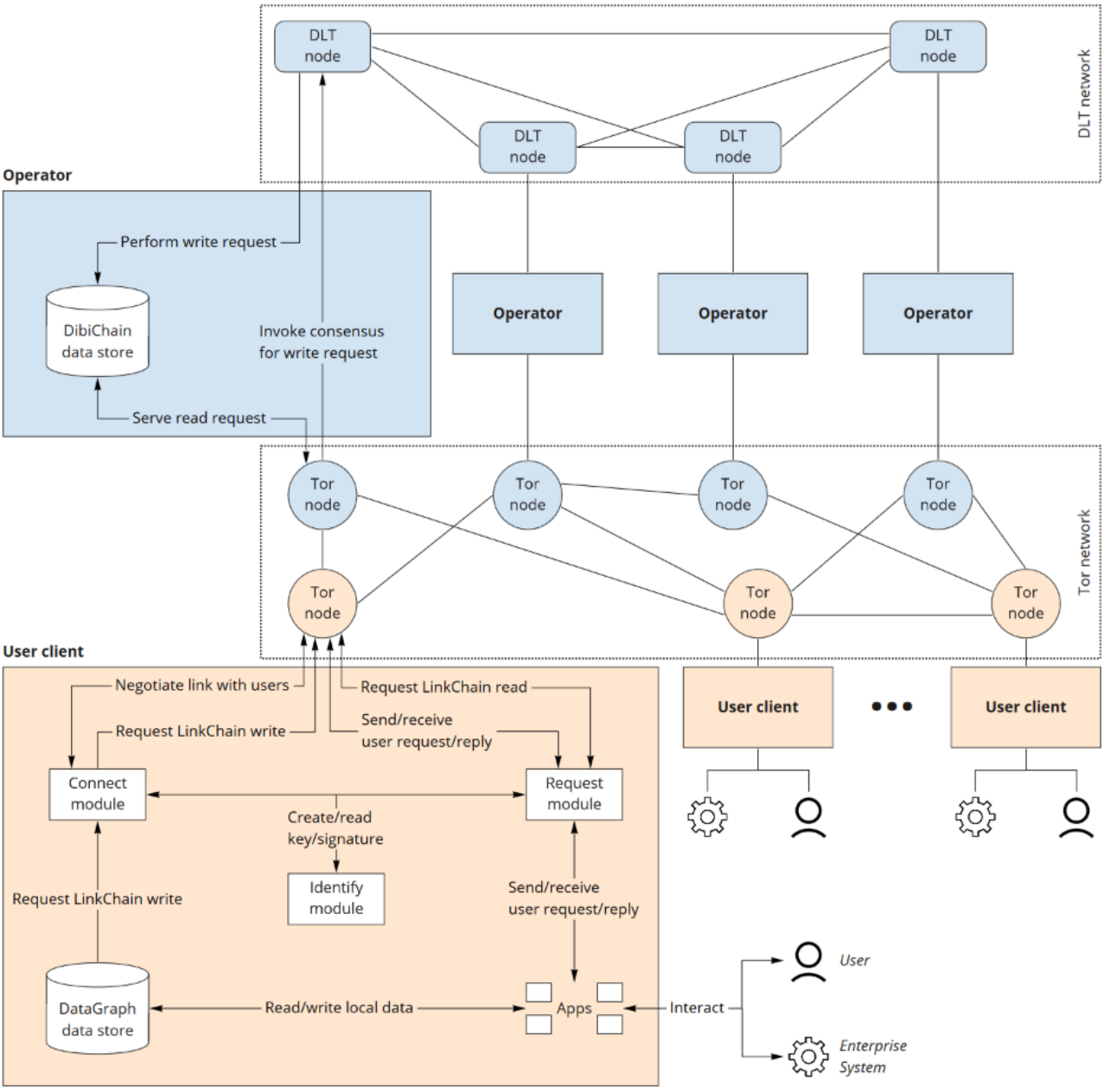

Figure 2: Ecosystem architecture. The DibiChain administrator is not shown.

To foster user acceptance and trust, ensure scalability and allow data store operators to monetize their services, the ecosystem is permissioned. To be able to publish information on the DibiChain data store and use the system's Tor network, users must register with an administrator as described in Section 3.1. The system's Tor nodes are operated by the users. This ensures that the network is large enough to provide sufficient anonymity.

Users access the ecosystem through a local user client. In addition to the Tor node, the user client consists of three core modules (Identify, Connect, Request), a graph data store (DataGraph) and a collection of so-called apps.

The Identify core module maintains the user's HDW (see Section 3.1), derives and stores key pairs and creates and checks cryptographic signatures. The Connect core module manages the publication of addresses and links in the DibiChain data store and provides tools to negotiate multi-party links with other user clients. The Request module allows users to send and receive requests and responses. Requests can be sent to DibiChain addresses through the Tor network and include a reply address to which the receiver can send a response, again through the Tor network. This allows users to request and exchange business-relevant information in a privacy-preserving way.[3]

The DataGraph is the user's local data store. In the interest of flexibility and extensibility, the data store is a graph database. Users can store business information in the DataGraph and make it discoverable by publishing corresponding addresses and links on the DibiChain data store. In addition, the DataGraph provides support for connecting user enterprise systems to minimize the need for data duplication or migration. Figure 3 illustrates the three data layers—enterprise systems, DataGraph, DibiChain data store—of the supply chain ecosystem.

Specific use cases are implemented within the ecosystem via apps. An app provides a user interface, connects to enterprise systems, defines and enforces data models and business logic, stores information in the DataGraph and interacts with other user clients and the DibiChain data store through the user client's core components. In this way, apps enable the implementation of a wide range of supply chain use cases (tracing, exchange of compliance documents, collection of sustainability data, etc.) within a single ecosystem.

4.1 A discovery service for EPCIS event data

To illustrate the practical implications of the ecosystem, we sketch how it can serve as a discovery service for EPCIS event data. In line with the federated approach to data storage, no event data are stored in shared databases—neither in full nor in sanitized form. Instead, every company stores its data locally in its own DataGraph. For each data object, the company publishes a corresponding address on the DibiChain data store. Connections between event data are published as DibiChain links.

As discussed in Section 3, addresses can be connected indirectly through multiple links with the same tag or directly through a single link. For product tracing, the (hash of the) product's unique identifier is a natural candidate for an indirect connection tag. A direct connection would require companies to collect addresses and signatures at the time of link publication. While this requires more coordination, it can also provide increased trust by documenting that all involved companies agree that a direct connection between the two events exists. In particular, a direct connection attests that there is no

[3] An important implementation detail is to ensure that users do not overutilize their own Tor node as an introduction point for their addresses, as this would allow observers to correlate them. The prototype implementation of the Connect core module randomly chooses introduction points from a regularly updated list of available Tor nodes.

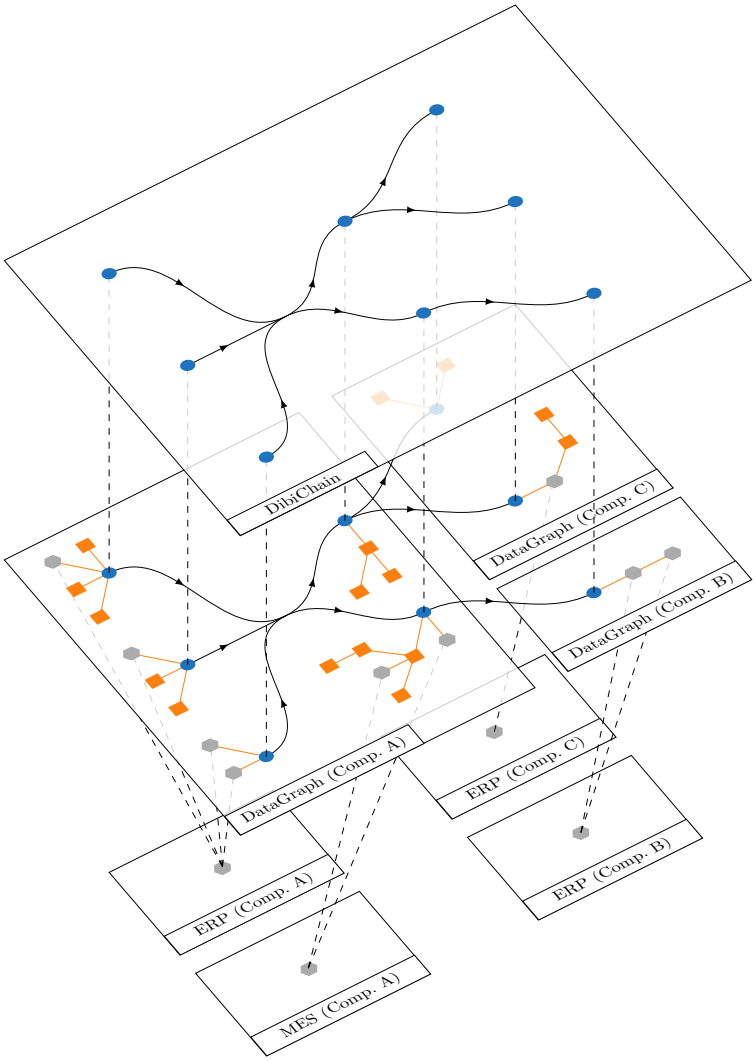

Figure 3: Data layers of the ecosystem for supply chain information.

"hole" in the chain of custody and can therefore help prevent the introduction of counterfeits into the supply chain.

Once the links are established, a company can determine which addresses are (directly or indirectly) connected to its own events on the DibiChain data store. The company can then send queries from its user client to the address IDs via Tor and request the corresponding EPCIS data. The address owner and the data itself remain unknown to the sender until the owner chooses to answer. Furthermore, the request itself and any subsequent communication are only visible to the sender and the receiver and cannot be observed by any third party, including the data store operators and the administrator.

Figure 4 illustrates how EPCIS event data are registered and connected on the DibiChain. In addition to the chain of events, we show how the DibiChain can also represent the chain of custody. Maintaining a separate chain of custody makes it possible to define access rights based on the distance in the supply chain. For example, Farmer A could prove to Distributor C that he is a direct supplier of their common business contact (Processor B) by signing a request with the private key corresponding to his address in the chain of custody.

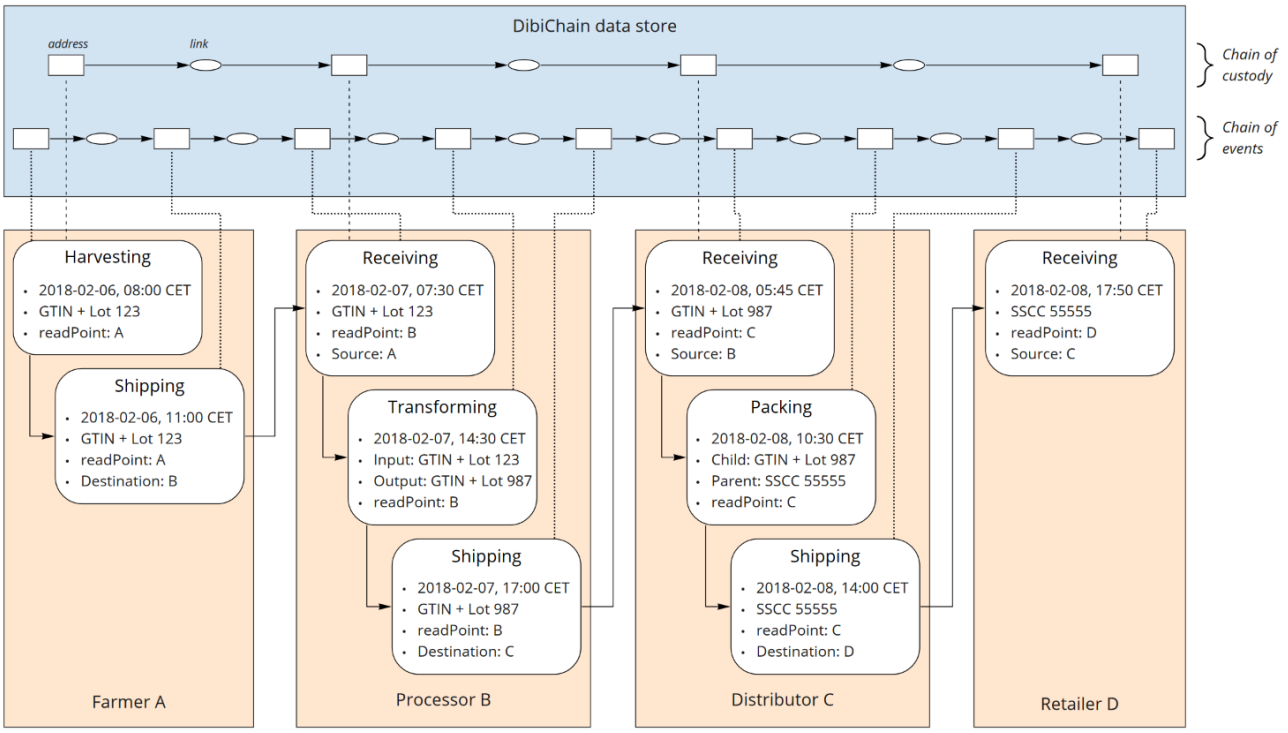

Figure 4: Illustration of how a chain of EPCIS event data are registered and connected on the DibiChain data store. The example EPCIS events are taken from Figure 2 in Tröger et al. (2018). For simplicity, we show all connections as direct connections. Supply chain members can request EPCIS data by sending queries to the corresponding address IDs via the Tor network.

5 Discussion

The DibiChain protocol combines ideas and approaches from various internet technologies. It builds on the Tor protocol for anonymous communication. Indeed, the DibiChain data store can be viewed as a "Tor lookup service with links." Tamper-resistance by means of cryptographic hashes and signatures is achieved in a similar way as in the Bitcoin protocol. In comparison to protocols for blockchain-based cryptocurrency, the DibiChain protocol is more generic. Whereas cryptocurrency protocols implement validity checks to prevent double-spending and the uncontrolled creation of new currency, the DibiChain protocol employs a lower degree of "algorithmic self-policing" (Swan, 2016, p. vii) as all well-formed addresses and links are considered valid. This makes the DibiChain more flexible but less strict.

DibiChain addresses and links also share similarities with concepts of self-sovereign identity (SSI) (Allen, 2016), such as the W3C's specifications for decentralized identifiers (DIDs) (Reed et al., 2020) and verifiable credentials (VCs) (Sporny et al., 2019). These specifications, too, deal with the publication of anonymous identifiers on a public data store and achieve non-repudiability and tamper-resistance by cryptographic means. In the same way that DibiChain links make claims about DibiChain addresses, VCs make claims about DIDs. DibiChain links, however, differ from VCs in three important respects: (1) They are visible to anyone who can access the DibiChain data store, (2) they are self-signed by the affected address owners and (3) they make generic claims ("address A is connected to address B") as opposed to specific claims ("this DID belongs to a person who is at least 18 years old").

Many aspects of the DibiChain specification were chosen to ensure a high level of privacy. By using Tor, users remain anonymous on the communication level. (A user can nonetheless provide identifying information via message payloads, particularly as part of user-to-user communication.)

Furthermore, address and link identifiers are single-use. There are no persistent user identifiers which would make it easy to correlate addresses and analyze usage patterns or user relationships.

Adding timestamps and data hashes as described in Section 3.2, however, could potentially jeopardize privacy. Analysis of timestamps may allow observers to correlate addresses, for example when links are published at regular time intervals (Dorri et al., 2019). Data hashes of structured data (e.g., JSON objects with known field names and data ranges) are prone to brute-force and dictionary attacks (Federal Office for Information Security, 2019). For each use case, the benefits and potential privacy risks of timestamping and notarization capabilities should be weighed carefully.

Some supply chain blockchains which store business information in shared data stores address privacy concerns by restricting the visibility of data to a predefined group of users in a "channel" or "stream" (see, e.g., Biswas et al., 2017). This, however, requires a priori knowledge of who should be part of which channel, which can be problematic in supply chains where upstream producers do not know where their products end up. In addition, fine-grained access control in large applications can require a staggering number of channels. While not a supply chain use case, this is illustrated by the banking cooperative SWIFT[4], which estimated that a DLT implementation of its international money transfer system would require more than 100,000 channels to address all privacy requirements (Finextra, 2018). While channels could be added to the DibiChain data store if required, the protocol's focus is on data minimization to ensure that the information value of shared data is as small as possible.

The high degree of privacy in the DibiChain-based discovery service comes at the cost of efficiency and availability. A company may only be interested in a single event in a highly complex supply chain (e.g., the provenance of the tin contained in one component of an airplane). Without out-of-band information, the company must send a query to every connected DibiChain address. Furthermore, if a company's DibiChain client is unavailable, the query cannot be delivered and thus not answered. Efficiency can be increased by publishing additional information, e.g., by extending the DibiChain address object to include additional use case specific information. But this also increases the information content within the DibiChain data store, thus increasing the risk of correlation. Availability can be increased by making user clients highly available or by delegating the task of storing data and answering queries to a service provider who guarantees high availability.

In addition to ensuring privacy, the DibiChain protocol is designed to provide favorable security and trust properties. For addresses, deriving the ID from a public key and attaching a cryptographic signature ensures that users can only control addresses for which they know the corresponding private key. For links, the signatures document that the affected address owners have seen and approved the link. These measures provide non-repudiation against address owners. They also provide tamper-resistance against the DibiChain operators: All users can check the link ID as well as address and link signatures to convince themselves that the data object has not been modified by an operator, e.g., by switching out the introduction points or adding tags.

Link signatures document approval from address owners at the cost of increased communication effort during link creation. For some use cases, requiring a full signature set may in fact be unnecessary. As an example, consider a cryptocurrency like Bitcoin (Nakamoto, 2008). Bitcoin transactions only require a proof of ownership from the senders of Bitcoin, not the receivers. This one-sided approval is sufficient because presumably nobody would object to receiving Bitcoins. On the DibiChain, the Bitcoin example corresponds to a link which only contains signatures for the addresses in `fromAddressIds`, not the addresses in `toAddressIds`. Other use cases, however, are likely to require a full signature set to be considered valid, especially when a DibiChain link represents a transaction which incurs liabilities for all involved users.

[4] https://www.swift.com.

With regard to security and trust, it is also important to note that the capabilities of the administrator as described in Section 3.1 are clearly staked out. A malicious administrator could leak identifying information or censor a user's new addresses (assuming the data store operators follow his verdict), but he could *not* publish addresses or links in the name of another user, tamper with DibiChain data or access locally stored data unnoticed or unauthorized. The administrator acts as a gatekeeper for the DibiChain data store, in line with the idea of integrating trusted third parties into "trustless" systems in a controlled manner (Strehle, 2020).

6 Conclusion

Connecting information across organizations remains a challenge, particularly when private information must be protected to maintain a competitive advantage. For DLT projects, concerns about confidentiality and privacy are among the most common reasons for project abandonment (Rauchs et al., 2019, Figure 27). The challenge of addressing such concerns shows itself clearly in supply chain relationships. A blockchain study by the World Economic Forum finds that in a given supply chain, one or more actors may "try to enforce a lack of visibility about the identity of upstream suppliers, the prices paid by downstream suppliers, the true length of a cash-conversion cycle, the status of regulatory compliance, true levels of demand and available inventory, and details about the production process" (Flanagan et al., 2019, p. 7).

We propose the DibiChain protocol as a way of connecting, discovering and exchanging information in a privacy-preserving way. The protocol puts heavy emphasis on data minimization, which in our view should always precede and supplement other protective measures such as access control, data encryption or obfuscation techniques. Consequently, the protocol refrains from assigning persistent user identifiers and does not require users to store private information—in full or in sanitized form—in shared data stores. The DibiChain protocol puts a thin layer of addresses and links above user-operated data silos, exposing just enough information to allow users to discover and exchange relevant information.

Enabling cooperation within and across supply chains becomes increasingly important as we reap the benefits and face the challenges of tightly integrated, global supply chains. The DibiChain protocol aims to offer a novel choice in the tradeoff between privacy and transparency. For use cases which require a high level of privacy and can do without the validity checks and smart contract logic offered by more transparent blockchain solutions, it may provide a viable alternative.

References

Abeyratne, S. A. and Monfared, R. P. (2016). Blockchain ready manufacturing supply chain using distributed ledger. *International Journal of Research in Engineering and Technology*, 5(9):1–10. https://doi.org/10.15623/ijret.2016.0509001.

Allen, C. (2016). The path to self-sovereign identity. Personal Blog. Retrieved Feb 4, 2021, from http://www.lifewithalacrity.com/2016/04/the-path-to-self-soverereign-identity.html.

Androulaki, E., Barger, A., Bortnikov, V., Cachin, C., Christidis, K., De Caro, A., Enyeart, D., Ferris, C., Laventman, G., Manevich, Y., Muralidharan, S., Murthy, C., Nguyen, B., Sethi, M., Singh, G., Smith, K., Sorniotti, A., Stathakopoulou, C., Vukolić, M., Cocco, S. W., and Yellick, J. (2018). Hyperledger Fabric: A distributed operating system for permissioned blockchains. In *Proceedings of the Thirteenth EuroSys Conference (EuroSys '18)*, pp. 1–15, Porto, PT. ACM Press. https://doi.org/10.1145/3190508.3190538.

Back, A. (1997). A partial hash collision based postage scheme. Technical report, Cypherpunks Mailing List. Retrieved Feb 4, 2021, from http://hashcash.org/papers/announce.txt.

Banker, S. (2018). Blockchain gains traction in the food supply chain. Forbes. Retrieved Feb 4, 2021, from https://www.forbes.com/sites/stevebanker/2018/07/25/blockchain-gains-traction-in-the-food-supply-chain/#21d846f91cf9.

Bettín-Díaz, R., Rojas, A. E., and Mejía-Moncayo, C. (2018). Methodological approach to the definition of a blockchain system for the food industry supply chain traceability. In *Computational Science and Its Applications – ICCSA 2018*, pp. 19–33. Springer International Publishing. https://doi.org/10.1007/978-3-319-95165-2_2.

Biswas, K., Muthukkumarasamy, V., and Tan, W. L. (2017). Blockchain based wine supply chain traceability system. In *Future Technologies Conference (FTC) 2017*, Vancouver, Canada.

Buterin, V. (2013). Ethereum whitepaper. Whitepaper, Ethereum Foundation. Retrieved Feb 4, 2021, from https://ethereum.org/whitepaper.

Cachin, C. and Vukolić, M. (2017). Blockchain consensus protocols in the wild. Working paper, IBM Research Zurich. Retrieved Feb 4, 2021, from https://arxiv.org/pdf/1707.01873.pdf.

Caro, M. P., Ali, M. S., Vecchio, M., and Giaffreda, R. (2018). Blockchain-based traceability in agri-food supply chain management: A practical implementation. In *2018 IoT Vertical and Topical Summit on Agriculture - Tuscany (IOT Tuscany)*. IEEE. https://doi.org/10.1109/iot-tuscany.2018.8373021.

Casado-Vara, R., Prieto, J., la Prieta, F. D., and Corchado, J. M. (2018). How blockchain improves the supply chain: Case study alimentary supply chain. *Procedia Computer Science*, 134:393–398. https://doi.org/10.1016/j.procs.2018.07.193.

Casino, F., Dasaklis, T. K., and Patsakis, C. (2019a). A systematic literature review of blockchain-based applications: Current status, classification and open issues. *Telematics and Informatics*, 36:55–81. https://doi.org/10.1016/j.tele.2018.11.006.

Casino, F., Kanakaris, V., Dasaklis, T. K., Moschuris, S., and Rachaniotis, N. P. (2019b). Modeling food supply chain traceability based on blockchain technology. *IFAC-PapersOnLine*, 52(13):2728–2733. https://doi.org/10.1016/j.ifacol.2019.11.620.

Castro, M. and Liskov, B. (1999). Practical Byzantine fault tolerance. In *Proceedings of the Third Symposium on Operating Systems Design and Implementation*, pp. 173–186.

Das, P., Faust, S., and Loss, J. (2019). A formal treatment of deterministic wallets. In *Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security*. ACM. https://doi.org/10.1145/3319535.3354236.

Dasaklis, T. K., Casino, F., Patsakis, C., and Douligeris, C. (2019). A framework for supply chain traceability based on blockchain tokens. In di Francescomarino, C., Dijkman, R., and Zdun, U. (eds.), *Business process management workshops. BPM 2019.*, pp. 704–716. Springer International Publishing, Cham, first edition.

Dingledine, R., Mathewson, N., and Syverson, P. (2004). Tor: The second-generation Onion Router. Whitepaper. Retrieved Feb 4, 2021 from https://svn-archive.torproject.org/svn/projects/design-paper/tor-design.pdf.

Dorri, A., Roulin, C., Jurdak, R., and Kanhere, S. S. (2019). On the activity privacy of blockchain for IoT. In *2019 IEEE 44th Conference on Local Computer Networks (LCN)*. IEEE. https://doi.org/10.1109/lcn44214.2019.8990819.

Dwork, C. and Naor, M. (1992). Pricing via processing or combatting junk mail. In *Advances in Cryptology — CRYPTO' 92*, pp. 139–147. Springer Berlin Heidelberg. https://doi.org/10.1007/3-540-48071-4_10.

Farkas, C., Ziegler, G., Meretei, A., and Lörincz, A. (2002). Anonymity and accountability in self-organizing electronic communities. In *Proceeding of the ACM Workshop on Privacy in the Electronic Society - WPES '02*. ACM Press. https://doi.org/10.1145/644527.644536.

Federal Office for Information Security (2019). Towards secure blockchains. Report, Federal Office for Information Security. Retrieved Feb 4, 2021, from https://www.bsi.bund.de/SharedDocs/Downloads/EN/BSI/Crypto/Secure_Blockchain.html.

Ferdousi, T., Gruenbacher, D., and Scoglio, C. M. (2020). A permissioned distributed ledger for the US beef cattle supply chain. *IEEE Access*, 8:154833–154847. https://doi.org/10.1109/access.2020.3019000.

Finextra (2018). Adoption of DLT presents significant operational challenges for Swift member banks. Finextra. Retrieved Feb 4, 2021, from https://www.finextra.com/newsarticle/31787/adoption-of-dlt-presents-significant-operational-challenges-for-swift-member-banks.

Flanagan, A. J., Maclean, F., Sun, M., Hewett, N., and Liao, R. (2019). Inclusive deployment of blockchain for supply chains: Part 4 – Protecting your data. Whitepaper, World Economic Forum. Retrieved Feb 4, 2021, from http://www3.weforum.org/docs/WEF_Inclusive_Deployment_of_Blockchain_for_Supply_Chains_Part_4_Report.pdf.

Garzik, J. (2015). Public versus private blockchains: Part 1: Permissioned blockchains. Whitepaper, BitFury Group. Retrieved Feb 4, 2021, from https://bitfury.com/content/downloads/public-vs-private-pt1-1.pdf.

Geng, Y., Sarkis, J., and Bleischwitz, R. (2019). How to globalize the circular economy. *Nature*, 565(7738):153–155. https://doi.org/10.1038/d41586-019-00017-z.

GS1 (2016a). EPC Information Services (EPCIS) standard (Release 1.2). Standards document, GS1. Retrieved Feb 4, 2021, from https://www.gs1.org/sites/default/files/docs/epc/EPCIS-Standard-1.2-r-2016-09-29.pdf.

GS1 (2016b). EPCIS and CBV implementation guideline (Release 1.2). Technical report, GS1. Retrieved Feb 4, 2021, from https://www.gs1.org/docs/epc/EPCIS_Guideline.pdf.

GS1 (2017a). Core Business Vocabulary (CBV) standard (Release 1.2.2). Standards document, GS1. Retrieved Feb 4, 2021, from https://www.gs1.org/sites/default/files/docs/epc/CBV-Standard-1-2-2-r-2017-10-12.pdf.

GS1 (2017b). GS1 global traceability standard (Release 2.0). Standards document, GS1. Retrieved Feb 4, 2021, from https://www.gs1.org/standards/traceability/traceability/2-0.

GS1 (2021). GS1 general specifications (Release 21.0.1). Standards document, GS1. Retrieved Feb 4, 2021, from https://www.gs1.org/sites/default/files/docs/barcodes/GS1_General_Specifications.pdf.

Gutoski, G. and Stebila, D. (2015). Hierarchical deterministic bitcoin wallets that tolerate key leakage. In *Financial Cryptography and Data Security*, pp. 497–504. Springer Berlin Heidelberg. https://doi.org/10.1007/978-3-662-47854-7_31.

Hackius, N., Reimers, S., and Kersten, W. (2019). The privacy barrier for blockchain in logistics: First lessons from the Port of Hamburg. In *Logistics Management*, pp. 45–61. Springer International Publishing. https://doi.org/10.1007/978-3-030-29821-0_4.

Hastig, G. M. and Sodhi, M. S. (2020). Blockchain for supply chain traceability: Business requirements and critical success factors. *Production and Operations Management*, 29(4):935–954. https://doi.org/10.1111/poms.13147.

Hewett, N., Lehmacher, W., and Wang, Y. (2019). Inclusive deployment of blockchain for supply chains: Part 1 – Introduction. Whitepaper, World Economic Forum. Retrieved Feb 4, 2020, from http://www3.weforum.org/docs/WEF_Introduction_to_Blockchain_for_Supply_Chains.pdf.

Juels, A. and Brainard, J. (2017). Client puzzles: A cryptographic defense against connection depletion attacks. Conference presentation, RSA Laboraties. Retrieved Feb 4, 2021, from https://www.ndss-symposium.org/wp-content/uploads/2017/09/Client-Puzzles-A-Cryptographic-Defense-Against-Connection-Depletion-Attacks.pdf.

Lamport, L., Shostak, R., and Pease, M. (1982). The Byzantine generals problem. *ACM Transactions on Programming Languages and Systems*, 4(3):382–401.

Lee, T., Pappas, C., Barrera, D., Szalachowski, P., and Perrig, A. (2016). Source accountability with domain-brokered privacy. Working paper, ETH Zurich. Retrieved Feb 4, 2021, from https://arxiv.org/pdf/1610.00461.pdf.

Leng, K., Bi, Y., Jing, L., Fu, H.-C., and Van Nieuwenhuyse, I. (2018). Research on agricultural supply chain system with double chain architecture based on blockchain technology. *Future Generation Computer Systems*, 86:641–649. https://doi.org/10.1016/j.future.2018.04.061.

Ma, Y., Wu, Y., Li, J., and Ge, J. (2020). APCN: A scalable architecture for balancing accountability and privacy in large-scale content-based networks. *Information Sciences*, 527:511–532. https://doi.org/10.1016/j.ins.2019.01.054.

Nakamoto, S. (2008). Bitcoin: A peer-to-peer electronic cash system. Whitepaper. Retrieved Feb 4, 2021, from https://bitcoin.org/bitcoin.pdf.

Narayanan, A., Bonneau, J., Felten, E., Miller, A., and Goldfeder, S. (2016). *Bitcoin and Cryptocurrency Technologies*. Princeton University Press.

Naylor, D., Mukerjee, M. K., and Steenkiste, P. (2015). Balancing accountability and privacy in the network. *ACM SIGCOMM Computer Communication Review*, 44(4):75–86. https://doi.org/10.1145/2740070.2626306.

Pearson, S., May, D., Leontidis, G., Swainson, M., Brewer, S., Bidaut, L., Frey, J. G., Parr, G., Maull, R., and Zisman, A. (2019). Are distributed ledger technologies the panacea for food traceability? *Global Food Security*, 20:145–149. https://doi.org/10.1016/j.gfs.2019.02.002.

Rauchs, M., Blandin, A., Bear, K., and McKeon, S. (2019). 2nd global enterprise blockchain benchmarking study. *SSRN Electronic Journal*. Retrieved Feb 4, 2021, from https://doi.org/10.2139/ssrn.3461765.

Rauchs, M., Glidden, A., Gordon, B., Pieters, G., Recanatini, M., Rostand, F., Vagneur, K., and Zhang, B. (2018). Distributed ledger technology systems: A conceptual framework. *SSRN Electronic Journal*. https://doi.org/10.2139/ssrn.3230013.

Reed, D., Sporny, M., Longley, D., Allen, C., Grant, R., Sabadello, M., and Holt, J. (2020). Decentralized Identifiers (DIDs) (Version 1.0). W3C working draft, W3C. Retrieved Feb 4, 2021, from https://www.w3.org/TR/did-core/.

Sporny, M., Longley, D., and Chadwick, D. (2019). Verifiable Credentials data model (Version 1.0). W3C recommendation, W3C. Retrieved Feb 4, 2021, from https://www.w3.org/TR/vc-data-model/.

Strehle, E. (2020). Public versus private blockchains. BRL working paper, Blockchain Research Lab. Retrieved Feb 4, 2021, from https://www.blockchainresearchlab.org/wp-content/uploads/2020/05/BRL-Working-Paper-No-14-Public-vs-Private-Blockchains.pdf.

Swan, M. (2016). *Blockchain: Blueprint for a New Economy*. O'Reilly UK Ltd., first edition.

Tian, F. (2018). *An Information System for Food Safety Monitoring in Supply Chains Based on HACCP, Blockchain and Internet of Things*. PhD thesis, WU Vienna University of Economics and Business. Retrieved Feb 4, 2021, from https://epub.wu.ac.at/6090/.

Tröger, R., Clanzett, S., and Lehmann, R. J. (2018). Innovative solution approach for controlling access to visibility data in open food supply chains. *Proceedings in Food System Dynamics*, p. Proceedings in System Dynamics and Innovation in Food Networks 2018. https://doi.org/10.18461/PFSD.2018.1817.

Wilhelm, M. and Sydow, J. (2018). Managing coopetition in supplier networks: A paradox perspective. *Journal of Supply Chain Management*, 54(3):22–41. https://doi.org/10.1111/jscm.12167.

Winans, K., Kendall, A., and Deng, H. (2017). The history and current applications of the circular economy concept. *Renewable and Sustainable Energy Reviews*, 68:825–833. https://doi.org/10.1016/j.rser.2016.09.123.

Wuille, P. (2012). BIP 0032: Hierarchical deterministic wallets. Bitcoin Wiki. Retrieved Feb 4, 2021, from https://en.bitcoin.it/wiki/BIP_0032.

Zhang, H. (2009). Vertical information exchange in a supply chain with duopoly retailers. *Production and Operations Management*, 11(4):531–546. https://doi.org/10.1111/j.1937-5956.2002.tb00476.x.

Declarations

Availability of data and materials
Not applicable.

Conflicts of interest
Not applicable.

Funding
The authors gratefully acknowledge funding through the joint research project DIBICHAIN within the framework of the ReziProK program, which is funded by Germany's Federal Ministry of Education (funding reference number: 033R241). Apart from the provision of funding, the funding source had no involvement in this study.

Acknowledgments
The authors thank Lennart Ante for an extensive review of an earlier draft.

About the Blockchain Research Lab

The Blockchain Research Lab promotes independent science and research on blockchain technologies and the publication of the results in the form of scientific papers and contributions to conferences and other media. The BRL is a non-profit organization aiming, on the one hand, to further the general understanding of the blockchain technology and, on the other hand, to analyze the resulting challenges and opportunities as well as their socio-economic consequences.

www.blockchainresearchlab.org