From Paper Trails to Trust on Tracks: Adding Public Transparency to Railways via zk-SNARKs

Tarek Galal
Information Systems Engineering
TU Berlin
Berlin, Germany
t.galal@tu-berlin.de

Valeria Tisch
Hasso Plattner Institute,
University of Potsdam
Potsdam, Germany
valeria.tisch@student.hpi.uni-potsdam.de

Katja Assaf, Andreas Polze
Hasso Plattner Institute,
University of Potsdam
Potsdam, Germany
{katja.assaf, andreas.polze}@hpi.de

Abstract—**Railways provide a critical service and operate under strict regulatory frameworks for implementing changes or upgrades. Despite their impact on the public, these frameworks do not define means or mechanisms for transparency towards the public, leading to reduced trust and complex tracking processes.**

We analyse the German guideline for railway-infrastructural modifications from proposal to approval, using the guideline as a motivating example for modelling decisions in processes using digital signatures and zero-knowledge proofs. Therein, a verifier can verify that a process was executed correctly by the involved parties and according to specification without learning confidential information such as trade secrets or identities of the participants. We validate our system by applying it to the railway process, demonstrating how it realises various rules, and we evaluate its scalability with increased process complexities. Our solution is not railway-specific but also applicable to other contexts, helping leverage zero-knowledge proofs for public transparency and trust.

Index Terms—**Railway, Blockchain, zkSNARKs, Transparency**

I. INTRODUCTION

Railway systems follow lengthy safety assurance processes for assessing proposals for infrastructural changes. These processes are carried out according to established specifications that define all requirements and procedures for implementing a particular type of project. Additionally, several national and international regulatory and accreditation bodies may review and certify each step to ensure compliance. Such measures are necessary for the safe operation of individual components as well as the entire system.

Despite their direct impact on the public, these processes typically do not have public transparency as a goal and are hard to trace. The interested public often has to work through press releases to find information or make use of their freedom of information (FOI) rights and explicitly request documents from government bodies. In most cases, these projects are partially or fully publicly funded. In Germany, for example, the government is responsible for the expansion and maintenance of the railway network[1].

Public Transparency Framework: Following the idea of "Public Money, Public Information" that advocates for freedom of information and transparency, we believe that modifications to publicly critical infrastructures should be publicly transparent and traceable. Therefore, we analyse the German guidelines for railway-infrastructural modifications, known as Sektorleitlinie 22, to understand typical workflows in railway processes. Accordingly, we propose our framework *Trust on Tracks* that augments not only railway industry processes but any similar business process with public transparency without requiring modifications to the process itself. Our framework enables modelling responsibilities and actions in a business process like railway modifications and constructs publicly verifiable zero-knowledge proofs that attest to the correctness of the process execution without revealing any potentially process-sensitive details like the identities of involved parties.

Specialized Business Processes: A business process is a structured set of activities designed to achieve a specific organisational goal. Although the definition is not unanimous and unspecific, it is widely used [1]. While Business Process Modelling (BPM) helps optimise complex workflows using standardised methods and notations, we focus on a specialised solution for the category of processes similar to Sektorleitlinie 22, described in [2][2]. Therein, a process is simply a hierarchy of roles with different responsibilities and tasks. Instead of following BPM patterns, which can add unnecessary complexity, our specialised approach prioritises simplicity and efficiency.

A. Contributions

We present *Trust on Tracks* as our framework for modelling hierarchical responsibilities in a business process and adding public transparency to the process outcomes. The framework supports processes involving multiple participants with hierarchical relationships amongst them, as well as specifying arbitrary business rules for the tasks they perform.

We construct the framework using zk-SNARKs and digital signatures. This ensures the authenticity of participants and their tasks in a process but also enables verification of arbitrary business requirements for relationships between participants or tasks and one another. Furthermore, the zero-knowledge property of the SNARK enables confidentiality of any aspect

---

[1]DB Capital Expenditures, https://ir.deutschebahn.com/en/db-group/capital-expenditures/ (accessed 16.03.2025)

[2]A newer version adding a security process became available in 2024.

of the process if necessary. Finally, we use the framework to implement the first phase of the railway specification, modelling the hierarchy of responsibilities and business rules, and demonstrating the feasibility and practicality of the framework.

B. Related Work

Research works investigating blockchain applications in the railway sector are primarily concerned with general applicability [3], decentralized railway control [4], communication bandwidth sharing [5], storing and sharing information about the railway system [6]–[9], or key respectively identity management [10], [11]. In 2020, the industry seemed fond of applying blockchain, stating, "DB[3] is currently working on around 20 use cases for blockchain technology, including logistics supply chains, more convenient ticketing across modes of transport, and rail operations."[4] However, links to the advertised *blockchain team*'s website have been taken down. Besides such statements, blockchain technology is not reported as being used by railway systems.

Beyond railways, several works address the use of the blockchain for adding transparency to various business processes. Some propose frameworks for executing business processes entirely on-chain [12]–[14], while others propose specialized solutions for concrete sectors like supply chain [15]. Here, smart contracts on the blockchain coordinate the execution of processes among different parties, allowing the processes to inherit public transparency of the blockchain. Our approach for adding transparency to business processes does not require modelling or executing the process on-chain. We let the process execute off-chain, and only use the blockchain for publishing proofs of the correctness of the process execution.

Combining zero-knowledge proofs with blockchains for transparency in business process was addressed in [16], [17]. These provide powerful frameworks for modelling entire business processes using the blockchain and zk-SNARKs, building upon Business Process Model and Notation (BPMN). However, as we are primarily interested in the subset of processes similar to Sektorleitlinie 22, we forgo the complexities of formal business process modelling and instead provide a specialized but simpler framework.

II. Exemplary Process from the German Railway

The Sektorleitlinie 22 [2] describes the verification process for railway applications in Germany. It consists of three main phases: *requirement specification (Lastenheft)*, *technical specification (Pflichtenheft)*[5], and *product (Produkt)*, and is in the jurisdiction of three main bodies: the *federal railway authority (FRA, dt. Eisenbahnbundesamt)*, the *German railway operator (Deutsche Bahn)* and the vendor. Each phase ends with a milestone, where a collection of documents, notifications, and any additional information the phase requires are submitted.

Participants involved in the process are organized in a hierarchical structure, illustrated in Figure 1, with the German government at the root of the hierarchy, being the official primary regulator of such processes. The *notified body* (NoBo) assesses conformity with European regulations to ensure interoperability. The *common safety method assessment body* (AsBo) ensures that the system is safe. The *designated body* (DeBo) is responsible for conformity with technical regulations. The product manager (BAV), appointed by the railway operator, is responsible for writing the requirement specification. Further, the railway operator may assign employees with special expertise to the approver role (FGV), and is required to notify the FRA of this assignment.

Phase requirement specification (RS): For accessibility of our model, we assume that a railway operator initiated the development of a new product and started the requirement specification phase. We assume our new product is type B[6] with no international interoperability requirements and two involved FGVs. The main deliverable of the requirement specification phase is the requirement specification (DocSR), a document provided to a vendor with supplementary material, such as safety assessment (DocSA), a report on conformity with technical regulations (DocTR) and partial test declaration (DocPTD), at the end of the requirement specification phase.

For our exemplary application scenario, Figure 2 illustrates how the deliverables in the specification requirement phase are connected. The role tree (from above) represents the hierarchical structure of the involved organizations and roles, and the document tree (from below) represents the project phase structure and its deliverables.

The RS Phase requires three notifications to be sent to the FRA: the system to be developed (SUC), the appointment of FGV_1, and the appointment of FGV_2. FGV_2 creates the final partial test declaration PTD. The declaration is supported by the safety assessment provided by the AsBo, the technical conformity report provided by the DeBo, the partial test declaration created by the other FGV_1 and the requirements specification itself created by the BAV.

Data Structure and Graph Transformation: The process in Figure 2 is represented by two trees. These are connected by edges representing roles acting on documents and notifications. Based on biology terminology where a fungal network connects trees, we call this data structure a *mycorrhizal network*.

To verify that a process phase was completed, all of its child nodes representing documents must be verifiable such that authorship and other actions are traceable to the responsible roles, roles were assigned by the designated, authorised roles, and documents and notifications correctly reference other documents according to the process specification. Since we have exactly one action from a role to a document or notification, we can transform our graph into a tree where each leaf represents a document or notification to be verified (Figure 3). Edges between the leaves represent references between documents or notifications. While we distinguish between

---

[3]DB is the German railway operator.
[4]https://deutschebahn.com/en/blockchain-6935084, (accessed 16.02.2025)
[5]The most common translation for both terms is *specification*.

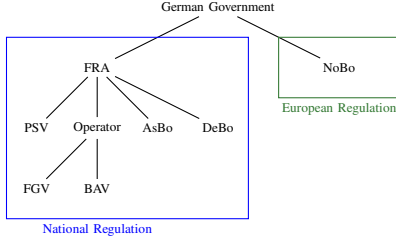[6]The list of type A products is provided in the appendix of [2].
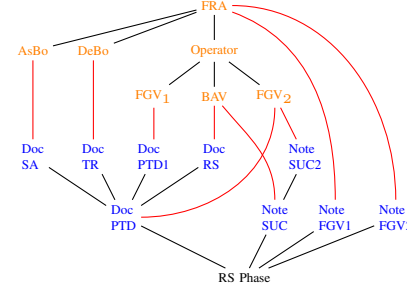
Fig. 1. Tree structure of roles

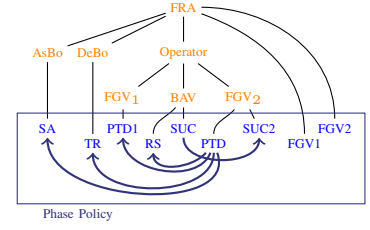Fig. 2. Mycorrhizal network of the requirement specification phase

Fig. 3. Tree structure of the requirement specification phase

documents and notifications to match the process specification, practically, they are only distinct in the kind of data they refer to. Thus, in our framework we abstract both as simply documents with different types, as seen in Figure 3.

III. TRUST ON TRACKS FRAMEWORK

In this section, we describe the setting and parties, then introduce Attestation Chains as a cryptographic building block that we use together with zk-SNARKs and a standard signature scheme to construct the Trust on Tracks framework.

A. Preliminaries

For an efficient binary relation $\mathcal{R}$, a Zero Knowledge Succinct Non-interactive Argument of Knowledge (zk-SNARK) is the three algorithms (Setup, Prove, Verify) where:

Setup$(1^\lambda, \mathcal{R}) \rightarrow crs$ takes a security parameter λ and the relation $\mathcal{R}$ and outputs a common reference string crs.

Prove$(crs, x, w) \rightarrow \pi$ takes input crs, public input x, and witness w, and generates the proof π.

Verify$(crs, x, \pi) \rightarrow \{0, 1\}$ outputs 1 if π is valid.

We use the notation $\mathcal{R} = \{(x; w) : stmt_1 \wedge ... \wedge stmt_n\}$ to describe the relations $\mathcal{R}$ for zk-SNARK with public inputs x, witness w, and a set of statements to prove.

B. Parties & Setting

Three kinds of parties participate in our system: One or more Attestor $\mathcal{A}$, a Prover $\mathcal{P}$, and a Verifier $\mathcal{V}$. An attestor $\mathcal{A}$ holds some role $rinfo$ and can assign role $rinfo'$ to another attestor $\mathcal{A}'$. Documents are objects created by attestors and represent any kind of data an attestor may issue. These can be actual documents, notifications, or arbitrary actions over other documents. We refer to a document as $dinfo$, a collection of documents as a *phase*, and a group of phases as a *process*. At the end of a phase, the prover creates a proof which can be confirmed by anyone acting as a verifier.

Framework Idea: Our process begins with a single *root attestor* $\mathcal{A}_r$ who *recruits* other attestors by assigning them some $rinfo$. Without loss of generality, we assume that every attestor holds exactly one $rinfo$. Thus, the group of attestors in a given process naturally map to a tree, with the *root attestor* $\mathcal{A}_r$ occupying the root, and the remaining attestors forming the remaining nodes. Documents issued by an attestor become children of that attestor's node and their respective branches

stop growing, setting them as leaves. Our goal is to verify the integrity of the tree. We do this in two steps: (1) we verify that the entire paths from all documents towards the root are legitimate – we refer to those paths as *attestation chains*, – and (2) verify that the attestors and documents conform to business rules by assigning meaning to all used $dinfo$ and $rinfo$ values, and matching them against the process specification.

C. Attestation Chain System

An *Attestation Chain system* is the tuple of algorithms (KGen, AttestDoc, AttestRole, VerifyChain) where:

KGen$(1^\lambda) \rightarrow (sk, pk)$**:** Generates key pair.

AttestDoc$(sk, dinfo) \rightarrow \sigma^{\mathsf{doc}}$**:** Outputs document attestation.

AttestRole$(sk, pk, rinfo) \rightarrow \sigma^{\mathsf{role}}$**:** Outputs role attestation.

VerifyChain$(dinfo, pk_0, rinfo_0, \sigma^{\mathsf{doc}}, \{(pk_i, rinfo_i, \sigma_i^{\mathsf{role}})\}^n)$
$\rightarrow \{0, 1\}$: Verifies the given attestation chain.

An Attestor $\mathcal{A}$ generates a dedicated key pair sk, pk by running KGen. $\mathcal{A}$ is assigned some role $rinfo$ if they obtain a role attestation σ^{role} over $(pk, rinfo)$ from another attestor $\mathcal{A}'$ who runs AttestRole with their secret key. Similarly, an attestor may attest to some document information $dinfo$ by running AttestDoc with her secret key.

In our tree model, an Attestation Chain is represented as the sequence of nodes starting with any document and ending at any attestor on the path to the root attestor. Thus, an attestation chain is the sequence:

$$(dinfo, (pk_0, rinfo_0, \sigma^{\mathsf{doc}}), (pk_1, rinfo_1, \sigma_0^{\mathsf{role}}), ..., (pk_n, rinfo_n, \sigma_{n-1}^{\mathsf{role}}))$$

We say that an Attestation Chain is valid iff every attestation in the sequence (over a document or an attestor) was indeed created by the subsequent attestor, i.e.,

$$\mathsf{AttestDoc}(sk_0, dinfo) = \sigma^{\mathsf{doc}} \wedge$$
$$\mathsf{AttestRole}(sk_1, rinfo_0) = \sigma_0^{\mathsf{role}} \wedge$$
$$\vdots$$
$$\mathsf{AttestRole}(sk_n, rinfo_{n-1}) = \sigma_{n-1}^{\mathsf{role}}$$

Verifying the tree's integrity thus requires verifying all chains.

Construction: We construct the system using a standard signature scheme Sig = (KGen, Sign, Verify) where KGen(pp) generates a key pair (sk, pk), Sign(sk, m) outputs signature σ for secret key sk and message m, and Verify(pk, m, σ) outputs

$\mathsf{KGen}(1^\lambda)$
___

$(sk, pk) \leftarrow \mathsf{Sig.KGen}(1^\lambda)$
return (sk, pk)

___

$\mathsf{AttestDoc}(sk, dinfo)$ $\qquad$ $\mathsf{AttestRole}(sk, pk, rinfo)$

___

$\sigma \leftarrow \mathsf{Sig.Sign}(sk, dinfo)$ $\quad$ $\sigma \leftarrow \mathsf{Sig.Sign}(sk, pk \| rinfo)$
return $\sigma^{\mathsf{doc}} := \sigma$ $\qquad\quad$ **return** $\sigma^{\mathsf{role}} := \sigma$

___

$\mathsf{VerifyChain}(dinfo, pk_0, rinfo_0, \sigma^{\mathsf{doc}}, \{(pk_i, rinfo_i, \sigma_i^{\mathsf{role}})\}^n)$

___

return 1 **if** $\mathsf{Sig.Verify}(pk_0, dinfo, \sigma^{\mathsf{doc}}) = 1 \wedge$

$\qquad \forall i \in [n]\ \mathsf{Sig.Verify}(pk_i, pk_{i-1} \| rinfo_{i-1}, \sigma_i^{\mathsf{role}}) = 1$

$\quad$ **else** 0

Fig. 4. Attestation Chain Construction using Standard Signature Scheme Sig.

1 if the signature is valid for m under the public key pk and 0 otherwise. Figure 4 shows the construction.

D. The Framework

We construct a zk-SNARK for verifying process trees using the Attestation Chain system as a building block. Given a set of Attestation Chains, a Root Attestor $\mathcal{A}_r$, and a description of business rules, the SNARK verifies all attestation chains with respect to $\mathcal{A}_r$ as well as conformance to the business rules.

1) Overview: The SNARK takes as input a root attestor public key rpk, and a set of l Attestation Chains with size n each as input. The root attestor is appointed by the process initiator who thereby requires all attestation chains to verify with respect to its public key rpk. To perform this verification, the SNARK requires that VerifyChain outputs 1 for each of the Attestation Chains, and that $pk_n = rpk$ in all of the chains. This ensures that the final attestation in every chain was created by the root attestor, effectively setting rpk at the root of the attestation tree as visualized in Figure 5.
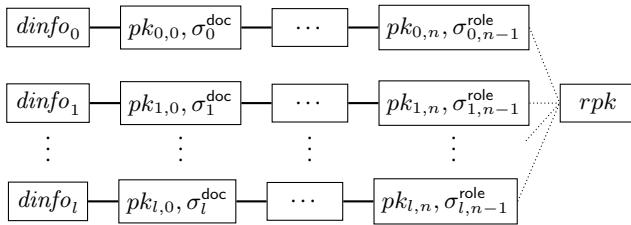

Fig. 5. Preprocessing of Attestation Chains. All chains are extended with the root attestor public key rpk, forming a tree of attestations rooted at rpk.

Business Rules: In addition to the cryptographic validity of the attestation chains, the SNARK enables two types of functional verifications to be specifiable via a Chain Policy function $f_c(dinfo, \{rinfo_i\}^n) \rightarrow \{0, 1\}$, and a Phase Policy function $f_p(\{dinfo_i\}^l) \rightarrow \{0, 1\}$. The Chain Policy takes as input the set of all $dinfo, rinfo$ building a particular attestation chain of size n. Within the function, meaning can be assigned to the raw values of $rinfo$ and get checked individually or in relation to any other given $rinfo' \neq rinfo$. For example, one

can model an attribute-based access control by encoding attribute names into $rinfo$ and checking in f_c whether attributes of a given $rinfo$ allow creating $dinfo$.

Analogously, the Phase Policy function f_p verifies the meaning assigned to all $dinfo$ across the entire set of all attestation chains as well as associations across them. For example, given $(dinfo_1, dinfo_2)$ as inputs to f_p, one can require that $dinfo_1$ is a review report for $dinfo_2$.

2) zk-SNARK Definition: We require a zk-SNARK that proves the following relation:

$$
\begin{aligned}
R = \big\{ &(rpk, f_p, f_c); \{(dinfo_i, pk_{i,0}, rinfo_{i,0}, \sigma_i^{\mathsf{doc}}, \\
&\qquad\qquad \{(pk_{i,j}, rinfo_{i,j}, \sigma_{i,j}^{\mathsf{role}})\}^{n_i})\}^l \big\} : \\
&\quad f_p(dinfo_0, ..., dinfo_l) \quad = 1 \ \wedge \\
\forall i \in [l]\quad & f_c(rinfo_{i,0}, ..., rinfo_{i,n_i}) = 1 \ \wedge \\
&\qquad\qquad\qquad rpk = pk_{i,n_i} \ \wedge \\
&\mathsf{VerifyChain}(dinfo_i, pk_{i,0}, rinfo_{i,0}, \sigma_i^{\mathsf{doc}}, \\
&\qquad\qquad \{(pk_{i,j}, rinfo_{i,j}, \sigma_{i,j}^{\mathsf{role}})\}^{n_i}) = 1
\end{aligned}
$$

IV. EVALUATION

In this section, we evaluate our Trust on Tracks framework by instantiating it for the first phase of Sektorleitlinie 22, the requirements specification phase, as described in Section II.

A. Railway Instance

To instantiate our framework for Phase 1 of Sektorleitlinie 22, we implement the Attestation Chain Scheme, the zk-SNARK, as well the chain and phase policy verification functions. For constructing zk-SNARK circuits, high-level languages, such as ZoKrates [18], provide a syntax that closely resemble familiar programming languages, abstracting away the underlying circuit details and complexities. To be compatible with ZoKrates, we realise our Attestation Chain scheme using EdDSA over the BabyJubJub curve, which corresponds to the scalar field of BN128 which is used by ZoKrates. The code is open source and available on Github [7].

1) Modelling dinfo: We encode into $dinfo$ the information (`doctype`, `identifier`, `ref`) where `doctype` denotes the type of a document, `identifier` is a SHA-256 hash of the document, and `ref` is a SHA-256 hash of concatenated `identifier` values of any referenced $dinfo$.

Phase 1, as described in Section II, requires SA, TR, PTD, PTD1, RS, SUC, SUC2, FGV1, FGV2 as deliverable document types `doctype`. The `identifier` field is populated by the SHA-256 hash of the digital copy of the respective document. Finally, according to the specification, PTD references RS, PTD1, TR, and SA, while SUC references SUC2. To model those relations, we generate `ref` for PTD as SHA-256(ID(RS)$\|$ID(PTD1)$\|$ID(TR)$\|$ID(SA)) and for SUC as SHA-256(ID(SUC2)) where ID(`doctype`) outputs the document identifier for a document of type `doctype`.

___

[7] https://github.com/tgalal/trust-on-tracks

2) Modelling rinfo: We encode into $rinfo$ the set of document types an Attestor may create. This is a subset of the power set $\mathcal{P}(\{\texttt{SA}, \texttt{TR}, \texttt{PTD}, \texttt{RS}, \texttt{SUC}, \texttt{PTD1}, \texttt{SUC2}, \texttt{FGV1}, \texttt{FGV2}\})$. We implement those sets by encoding every `doctype` as a unique power of 2 integer, and setting the value of $rinfo$ to the bit-wise OR of all permitted document types.

3) Chain & Phase Policies: In the Chain Policy function f_c we implement hierarchical attribute based roles that support delegation of attributes down the hierarchy. Attributes in $rinfo$ represent document types the holder is allowed to create or delegate a subset of to another attestor. This places every Attestor as an *Attribute Authority* for the attributes it holds.

In the Phase Policy function f_p we implement the business rules for the first phase Sektorleitlinie 22. Therein, the function verifies the references between the submitted documents.

B. Performance

We execute our implementation on a `c5.4xlarge` EC2 instance with 32 GB memory and 16 3.00 GHz vCPUs. We measure the execution time and maximum memory consumption for compilation, setup, and proof generation for different attestation tree sizes and show the results in Table I, divided in two sections. The upper section shows measurements for arbitrary input sizes and shows that the execution times and memory usage grow linear in the total number of signatures. The total number of signatures is influenced by the number of input documents together with the tree depth, which in turn influences the number of attestations per document. To simplify the implementation, we fix the path from any document to the root, i.e., the number of attestations per document, to always match the tree depth. Thus, the total number of signatures is always equal the number of documents $\times$ tree depth. The lower section of the table contains measurements obtained from the configuration of Phase 1 of Sektorleitlinie 22 specification. The number of CPUs does not impact the performance as all process steps, i.e. compilation, setup, and proof generation, are single-threaded.

C. Publishing Proofs on the Blockchain

Proofs generated by ZoKrates are highly optimized for storage and verification on the Ethereum blockchain. In fact, ZoKrates generates the respective smart contract code in solidity, ready for deployment on the blockchain. Using the blockchain for storage and verification of those proofs strengthens trust in the business process. On the one hand, the blockchain acts as an immutable storage for those proofs, on the other hand, private auditors can verify the consistency between published proofs and any confidential data used in creating them - as part of any necessary auditing procedure.

It costs 5193982 Gas to deploy the ZoKrates-generated smart contract for Trust on Tracks, and 463359 Gas to store a single proof on the blockchain. On-chain verification of a proof costs 3679995 Gas, expected to be paid once by the prover. For an exemplary gas price of 0.42 gwei and Ethereum price of 1800 USD, these gas costs approximately correspond to $4.67, $0.42, $3.31 respectively.

The prover may also choose to split a process into phases and generate a proof for every phase individually. This has the advantage of added granularity into the progress of a process that gets communicated to verifiers. In this case, the costs for storing and verifying the different proofs are summed, but deploying the contract is largely not affected.

V. DISCUSSION

A. Trust on Tracks vs Blockchain-only

Modelling an entire process on blockchain [19]–[21] has drawbacks. It either requires the migration of an existing and working process, to be coordinated and executed entirely on the blockchain in form of a smart contract, or running the smart contract in sync with the process, mirroring every action or step taken. The former is unlikely to be adopted, and the latter adds complexity. Furthermore, dealing with confidential data presents a challenge to a blockchain-only solution and typically invokes off-chain workarounds such as IPFS.

We identified public transparency as a main motivation behind these works i.e., proving that a process was correctly executed. Trust on Tracks achieves that without modelling the entire process using smart contracts, and inherently enables confidentiality for any aspect of the process.

B. Tree Optimization

There are a few optimizable aspects in Trust on Tracks:

1) Redundant Signatures: All the attestation chains of a particular tree are of the same size that corresponds to the tree height; this carries a performance penalty. More specifically, according to Figure 3, there should be exactly 17 unique digital signatures in total, however our evaluation produced 27 for the same process. This discrepancy is attributed to a trade-off between flexibility and efficiency in modelling a tree using ZoKrates. When the circuit is compiled, all input sizes must be fixed. This would require determining the exact structure of the tree beforehand, i.e., the sizes of all branches and how the nodes are connected. Instead, we let the prover decide this structure and it gets checked during verification. Only the tree height is fixed during compilation, which resembles the maximum size of an attestation chain. Thus in Figure 3, although the chain from FRA to SA should optimally be of size 2, it is of size 3. As the overall complexity of a ZoKrates program is the sum of the cost of *all branches*, it does not matter whether this extra attestation is skipped or verified, hence we counted it as a redundant signature.

2) Redundant Branches: Additional redundant signatures appear due to representing the tree as paths from leaves to root. According to Figure 3, the Operator subtree should have 8 signatures. In reality, processing this subtree results in 10 signatures. Although the same BAV node lies on two attestation chains: from Operator to RS, and from Operator to SUC, the Operator's signature over BAV is verified twice. This is because we do not cross-check if an attestation that exists on two paths has been already verified and can be skipped. However, due to the reasons outlined in Section V-B1, simply cross-checking will not have a performance benefit as all

TABLE I
EXECUTION TIME AND MEMORY USAGE FOR VARIOUS EXEMPLARY INPUTS AND THE SEKTORLEITLINIE 22 SPECIFICATION.

| | Count | | | Compile | | Setup | | Proof Generation | |
Documents	Attestations	Signatures	Constraints	Time	Memory	Time	Memory	Time	Memory
Benchmarks with Exemplary Inputs									
1	1	1	101775	7s	0.9 GB	4s	0.4 GB	6s	0.5 GB
1	2	2	203314	14s	1.9 GB	8s	0.9 GB	12s	1 GB
2	2	4	406628	30s	3.8 GB	15s	1.8 GB	24s	2 GB
Phase 1 of Sektorleitlinie 22									
9	3	27	2843728	266s	26.1 GB	118s	12.3GB	167s	13.6GB

branches in the program are executed. Instead, addressing this requires an alternative and more complex tree representation.

VI. CONCLUSION

We presented the Trust on Tracks framework for adding public transparency to processes, inspired by the intricacies of the German guidelines for railway modifications Sektorleitlinie 22. In contrast to other approaches, our framework does not require existing processes to be modified or executed on the blockchain, instead, processes still execute off-chain, and only proofs of correctness are published on-chain for public verifiability and transparency. We evaluated the framework by implementing it for the first Phase of German railway guidelines and have shown its practicality for adoption.

AUTHOR STATEMENT

Valeria Tisch developed our blockchain-only solution and manuscript, which Katja Assaf and Andreas Polze supervised. Tarek Galal and Katja Assaf developed the Trust on Tracks solution and wrote the manuscript. Tarek Galal provided the implementation.

REFERENCES

[1] N. Melão and M. Pidd, "A conceptual framework for understanding business processes and business process modelling," *Information systems journal*, vol. 10, no. 2, pp. 105–129, 2000.

[2] "Sektorleitlinie für die zulassungsbewertung von signal-, telekommunikations- und elektrotechnischen anlagen (technische vorschrift)," Technische Vorschrift, 2021, german.

[3] F. Naser, "The potential use of blockchain technology in railway applications: an introduction of a mobility and speech recognition prototype," in *2018 IEEE International Conference on Big Data (Big Data)*, 2018, pp. 4516–4524.

[4] M. Kuperberg, D. Kindler, and S. Jeschke, "Are smart contracts and blockchains suitable for decentralized railway control?" *Ledger*, vol. 5, Aug. 2020. [Online]. Available: https://ledger.pitt.edu/ojs/ledger/article/view/158

[5] S. Qian, Q. Li, C. Wu, and J. Sheng, "Blockchain-based spectrum resource sharing and matching algorithm in high-speed railway communication networks," in *2021 Computing, Communications and IoT Applications (ComComAp)*, 2021, pp. 89–94.

[6] G. Muniandi, "Blockchain-enabled secure crowdsensing for trackside infrastructure information collection and validation in railway signalling data preparation," *IET Blockchain*, vol. 1, no. 1, pp. 16–32, 2021. [Online]. Available: https://doi.org/10.1049/blc2.12002

[7] G. Hua, L. Zhu, J. Wu, C. Shen, L. Zhou, and Q. Lin, "Blockchain-based federated learning for intelligent control in heavy haul railway," *IEEE Access*, vol. 8, pp. 176830–176839, 2020.

[8] S. Rüsch, K. Bleeke, I. Messadi, S. Schmidt, A. Krampf, K. Olze, S. Stahnke, R. Schmid, L. Pirl, R. Kittel, A. Polze, M. Franz, M. Müller, L. Jehl, and R. Kapitza, "Zugchain: Blockchain-based juridical data recording in railway systems," in *52nd Annual IEEE/IFIP International Conference on Dependable Systems and Networks, DSN 2022, Baltimore, MD, USA, June 27-30, 2022*. IEEE, 2022, pp. 67–78. [Online]. Available: https://doi.org/10.1109/DSN53405.2022.00019

[9] L. Ni and R. Zhang, "A secure storage system for high-speed railway monitoring data based on blockchain technology," in *4th International Conference on Machine Learning and Computer Application, ICMLCA 2023, Hangzhou, China, October 27-29, 2023*. ACM, 2023, pp. 333–338. [Online]. Available: https://doi.org/10.1145/3650215.3650274

[10] Z. Zhang, J. Li, Y. Sun, Y. Li, H. Song, and H. Dong, "Blockchain-based secure key management model for high-speed railway," in *2022 IEEE 25th International Conference on Intelligent Transportation Systems (ITSC)*, 2022, pp. 1988–1993.

[11] Y. Feng, Z. Zhong, X. Sun, L. Wang, Y. Lu, and Y. Zhu, "Blockchain enabled zero trust based authentication scheme for railway communication networks," *J. Cloud Comput.*, vol. 12, no. 1, p. 62, 2023. [Online]. Available: https://doi.org/10.1186/s13677-023-00411-z

[12] I. Weber, X. Xu, R. Riveret, G. Governatori, A. Ponomarev, and J. Mendling, "Untrusted business process monitoring and execution using blockchain," in *Business Process Management: 14th International Conference, BPM 2016, Rio de Janeiro, Brazil, September 18-22, 2016. Proceedings 14*. Springer, 2016, pp. 329–347.

[13] L. García-Bañuelos, A. Ponomarev, M. Dumas, and I. Weber, "Optimized execution of business processes on blockchain," in *Business Process Management: 15th International Conference, BPM 2017, Barcelona, Spain, September 10–15, 2017, Proceedings 15*. Springer, 2017, pp. 130–146.

[14] N. O. Nawari and S. Ravindran, "Blockchain technology and bim process: Review and potential applications." *Journal of Information Technology in Construction*, vol. 24, 2019.

[15] S. E. Chang, Y.-C. Chen, and M.-F. Lu, "Supply chain re-engineering using blockchain technology: A case of smart contract based tracking process," *Technological forecasting and social change*, vol. 144, pp. 1–11, 2019.

[16] A. Toots, "Zero-knowledge proofs for business processes," Ph.D. dissertation, Master's thesis, University of Tartu, 2020.

[17] O. Petto, T. Preindl, and M. Kjäer, "Interpreted and confidential execution of process choreographies on a blockchain," in *International Conference on Business Process Management*. Springer, 2024, pp. 40–54.

[18] J. Eberhardt and S. Tai, "Zokrates - scalable privacy-preserving off-chain computations," in *2018 IEEE International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData)*, 2018, pp. 1084–1091.

[19] G. Fridgen, S. Radszuwill, N. Urbach, and L. Utz, "Cross-organizational workflow management using blockchain technology - towards applicability, auditability, and automation," pp. 1–10, 2018.

[20] M. Díaz, E. Soler, L. Llopis, and J. Trillo, "Integrating blockchain in safety-critical systems: An application to the nuclear industry," *IEEE Access*, vol. 8, pp. 190605–190619, 2020.

[21] J. Baumann, "Nachvollziehbare, versionierte und verteile speicherung von digitaler eisenbahninfrastrukturplanung," Master thesis, Hasso-Plattner-Institut, Universität Potsdam, Sep. 2024, german.