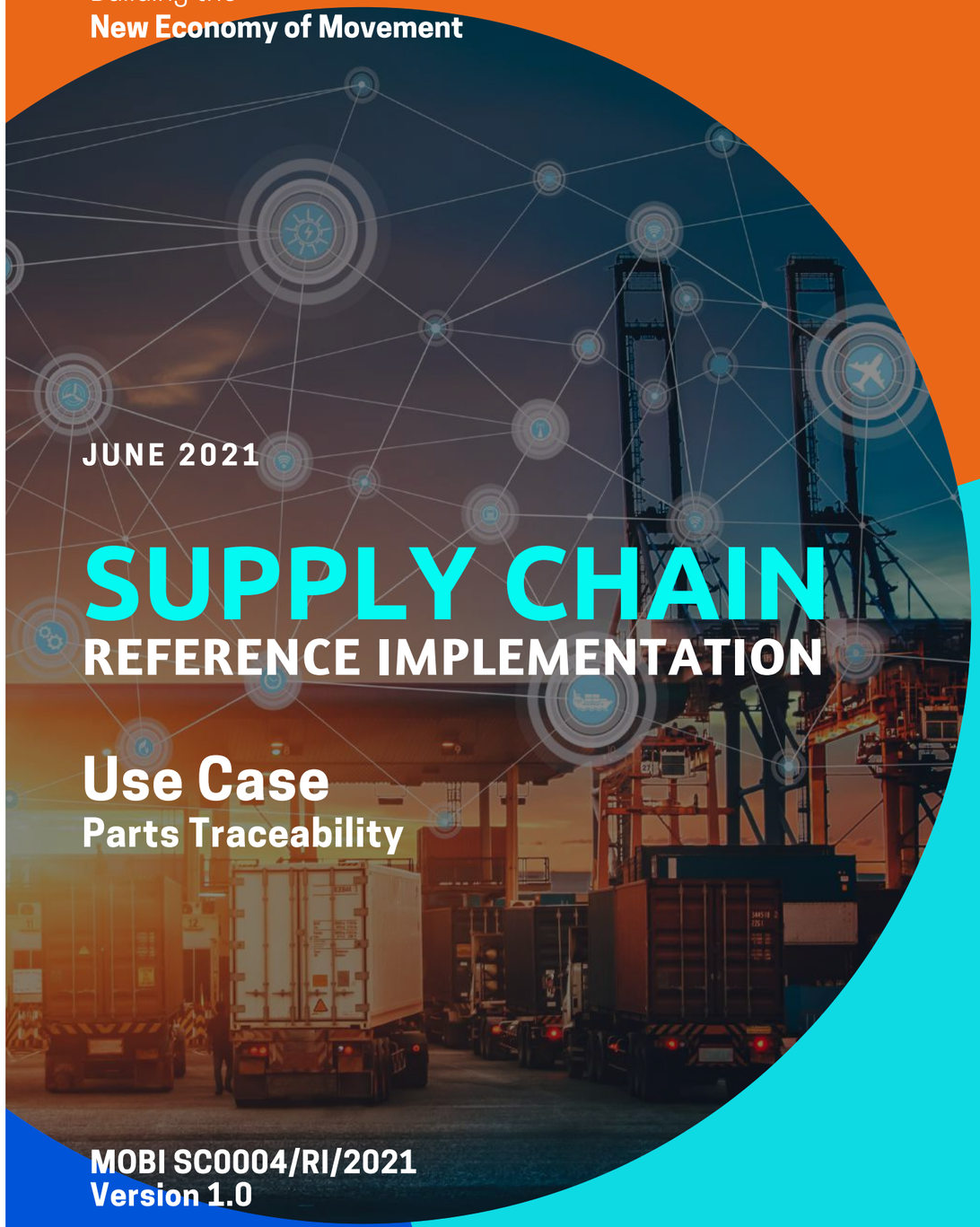
MOBI

Building the
New Economy of Movement



JUNE 2021

SUPPLY CHAIN
REFERENCE IMPLEMENTATION

Use Case
Parts Traceability

MOBI SC0004/RI/2021
Version 1.0

INTRODUCTION

The Mobility Open Blockchain Initiative (MOBI) Supply Chain Working Group is a global, multi-stakeholder project working to co-design blockchain and distributed ledger technologies standards to improve efficiency in the automotive supply chain. The project engages stakeholders across the chain: OEMs, Tier-N suppliers, dealers, consumers, and regulatory agencies. This report is based on numerous discussions, workshops, and research. Opinions expressed herein do not necessarily reflect the views of individual members.

Sincere thanks are extended to those who contributed their unique insights to this report.

Author

Johannes Sedlmeir, FIM Research Center

SC Working Group Co-Chairs

Pramita Mitra, Ford
Daniel Miehle/Dominik Batz, BMW Group

SC Working Group Team Members

Loren Adams, Accenture
Chris Ballinger, MOBI
Sebastian Banescu, Quantstamp
Olof Belfrage, CEVT
Roger Berg, DENSO
Betul Betul Kahya, MOBI
José Manuel Cantera, IOTA Foundation
Josh Cartellone, Accenture
Alisa DiCaprio, R3
Thad Dungan, Amazon
Chris Floersch, Honda
Josh Fodale, Ford
Kami Gaweda, Arxum
Jeremy Goodwin, SyncFab
Shannon Hamilton, DLT Labs
Philips Harrison, Fifth-9
Carsten Hiemsch, IBM
Thi Hong Tran, NAIST
Chris Hwee, Honda
Karthik Krishnamurthy, Amazon
Marco Lang, Marelli
Jens Lund-Nielsen, IOTA Foundation
Piyush Manocha, Accenture
Vinay Munjewar, SyncFab
Heiko Musa, BMW
Diwakar Muthu, Ford
Richard Nolk, CEVT
Drew Paroz, Honda
Dean Philips, Amazon
Angela Ruthenberg, AutoData Solutions
Kristin Marie Slanina, Thirdware
Anne Smith, IOTA
Kellie Treppa, Thirdware
Carl Youngblood, Amazon

MOBI Team

Chris Ballinger, Co-director + Founder
Tram Vo, Co-director + Founder
Rajat Rajbhandari, Working Groups Lead
Robin Pilling, Technical Lead
Matt Shi, Fellow
Kelly Clark, Communications Manager
Grace Pulliam, Communications Associate

TABLE OF CONTENTS

Executive Summary

Blockchain/Distributed Ledger Technology in supply chain improves efficiency and automation by coordinating multi-stakeholder processes and dissemination of mutually beneficial events.

Blockchain/Distributed Ledger Technology (DLT) provides many capabilities to make supply chains more transparent and efficient. It can address acknowledged challenges of today's complex supply chains, for example, by making it easier to check for data integrity and authenticity, allowing the prevention of double-spending of production decisions, parts custody, recalls etc. Blockchain improves efficiency and automation by coordinating multi-stakeholder processes and the dissemination of events.

Blockchain/DLT provides a trust anchor to support multiple stakeholders to authenticate each other's identities and data.

Two foundational capabilities are required to address these challenges, i.e., the ability for stakeholders to securely authenticate each other's identities and exchange data, and the ability to track and trace part lifecycle events from birth to end of life. Both of these capabilities require a single source of truth for supporting multi-stakeholder workflows around trusted data, and that's where blockchain / distributed ledger technology act as a pivotal enabler.

Significant challenges exists in implementing blockchain/DLT primarily in solving tradeoffs of privacy, scalability, and replicated storage.

However, there are also significant technical challenges involved with the adoption of blockchain in supply chains, as the specific scalability and privacy requirements of each use case need to be met. Tackling these challenges, which are associated with the replicated storage and execution of transactions on a blockchain, is the goal of many ongoing research and industry projects. Solutions that have been proposed exhibit different tradeoffs: for example, integrating Trusted Execution Environments or developing hand-crafted solutions based on advanced cryptography, such as Secure Multiparty Computation or Zero-Knowledge Proofs, can avoid information exposure but add additional complexity and computational overhead.

Executive Summary

Other established methods, such as writing only transaction hashes on a blockchain, are simple to implement; yet they cannot bring the anticipated benefits in every use case as this approach reduces the number of participants with access to the blockchain or the information is stored on-chain and hence accessible to other participants and smart contracts. Besides the application-layer design choices, there are also many degrees of freedom regarding the setup of a blockchain-based supply chain solution on the infrastructure layer: Tradeoffs exist for the usage of permissionless or permissioned architectures; and, in the latter case, there may be one blockchain that comprises the whole supply chain ecosystem, one blockchain per node, or even many blockchains that may involve only two or three parties each.

Use cases in supply chain are diverse in terms of requirements pertaining to data protection, antitrust regulation. Hence, one-size fits all reference architecture is difficult to design.

On the other hand, use cases differ considerably regarding the sensitivity of the related data in terms of data protection regulation, antitrust regulation, and their classification of business secrets. Stakeholders also do not have the same technical capabilities and willingness to adopt innovative yet complex solutions that help to leverage the advantages of blockchains while addressing the challenges. In summary, the heterogeneity of use cases and the variety of potential design choices prevent the feasibility of a one-size-fits-all blockchain architecture as a solution.

Consequently, this reference architecture focuses on discussing the benefits and potential interplay of different blockchain setups, embedded into a system that provides a basic communication layer, and the role of privacy enhancing technologies. The system would offer a rich toolkit to account for the spectrum of requirements of use cases that need to be implemented. The reference implementation suggests

This reference architecture focuses on standardized bilateral communication using extensive API calls, decentralized identity with blockchain/DLT as a trust anchor.

standardized bilateral communication and process management between business partners in the supply chain in combination with replicated execution on a DLT where useful.

Application Programming Interfaces (API) can shield the specific choices of stakeholders regarding their integration of a blockchain behind a layer of abstraction and enable multi-stakeholder workflows such as recalls through the supply chain without disclosing related information to third parties. This design is facilitated by cross-organizational digital identities for all parties, managed in a decentralized approach that may be anchored to a blockchain and oriented at the W3C DID and Verifiable Credentials (VC) standards that have already built the basis for the MOBI VID II standard.

Decentralized identity management enables a cross-organizational access management for the standardized API endpoints that we propose as the base infrastructure for the exchange of information between business partners, as well as the source for demonstrating permissions on a joint DLT platform. As this approach supports many degrees of exposing the exchanged information to further stakeholders, benefits of increased transparency can be leveraged while challenges associated with sensitive data, that blockchains frequently encounter, can be avoided.

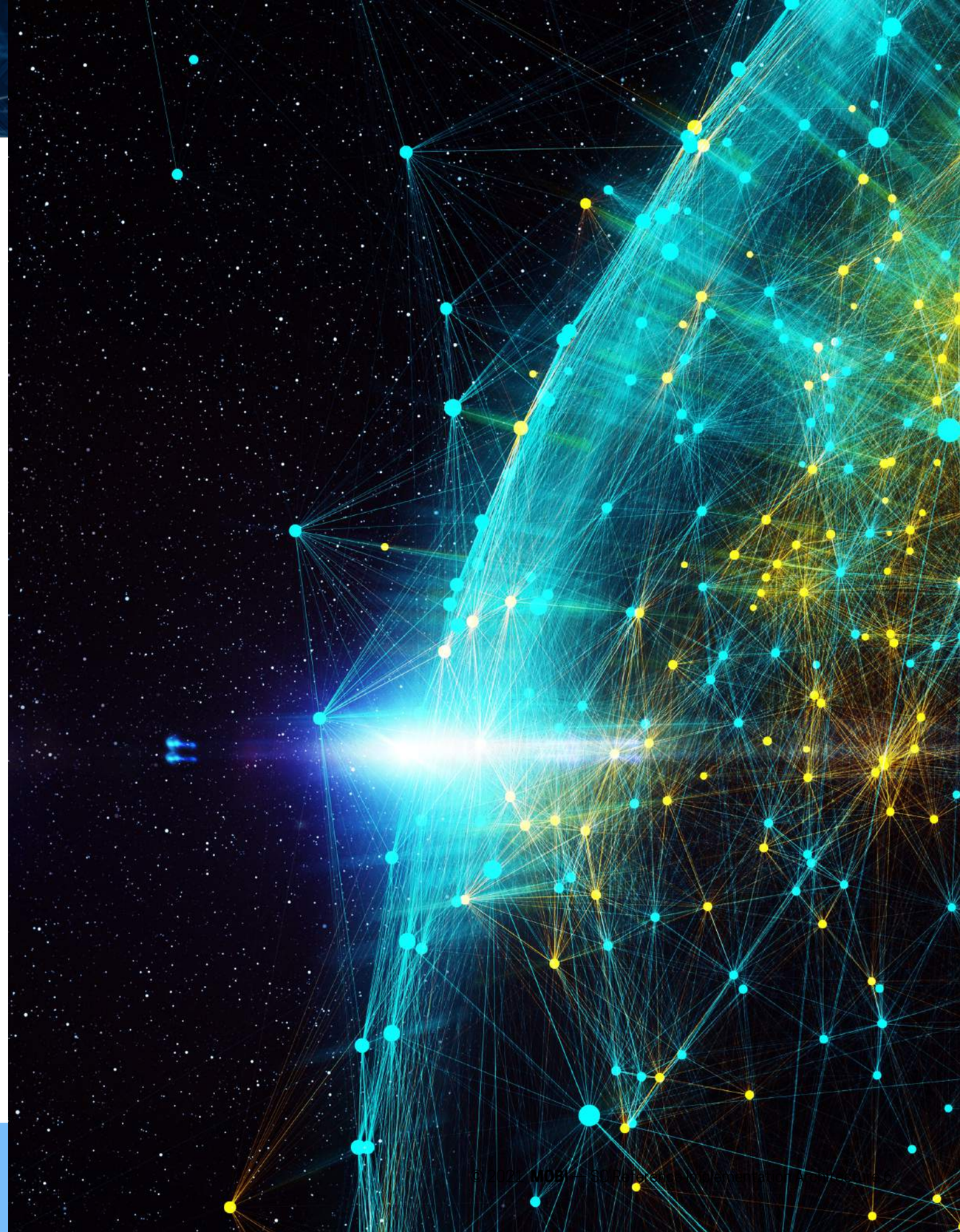
This architecture uses "cascading API calls" and smart contracts based event handling.

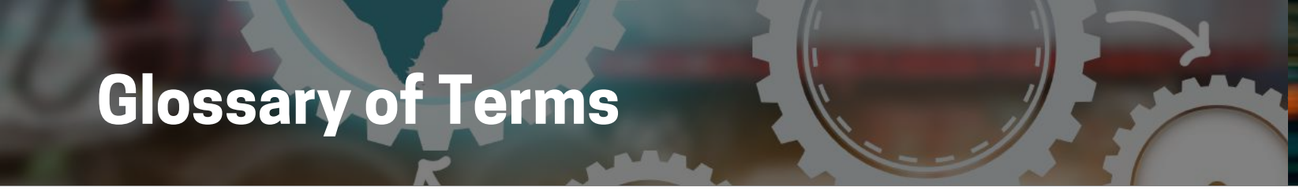
Depending on the use case under consideration, multi-stakeholder processes, such as recalls, can be automated through “cascading API calls” or smart-contract based event handling. This spectrum hence supports both coordinating use cases that allow for a high degree of transparency across multiple stakeholders with a highly robust design that does not

Executive Summary

rely on the availability of information from each individual party in the supply chain of a specific component in a recall.

The standard hence motivates the combination of digital identities, bilateral data exchange, distributed ledgers, and privacy-enhancing technologies to offer transparency to the extent that is required and providing the benefits of smart contracts in terms of efficiency where the related information exposure is justified or privacy-enhancing technologies can reduce information exposure while maintaining the core benefits.





Glossary of Terms

Application Programming Interface (API) — An API is defined as a specification of possible interactions with a software component, allowing two applications to interact with each other. It defines the kinds of calls or requests that can be made, how to make them, the data formats that should be used, the conventions to follow, etc (Hubspire, 2021). Examples for APIs are digital services that can be triggered via HTTP requests to change the state of, or receive information from, this application over the internet.

Differential Privacy (DP) — A method for obfuscating sensitive information by adding noise with expectation value 0 to data (deterministically if repeated queries are possible as otherwise the true value could be localized easily). Differential Privacy is best suitable in scenarios where large datasets are aggregated as the expected value of the average does not change under adding noise whose distribution is centered at 0, and where mathematical guarantees for the indistinguishability of entities with respect to query results are required.

Digital Identifier (DID) — from VID II (W3C-oriented).

Homomorphic Encryption (HE) — Allows performing computations on encrypted data (i.e., making meaningful operations without decrypting and knowledge of the decryption key). HE is either very limited in scope (either additions or multiplications, such as in the Pallier cryptosystem) or brings significant performance challenges when arbitrary operations need to be supported ("fully homomorphic encryption").

Logistics Service Provider (LSP) — from the business draft.

Master Data — Specifications and information related to products and business partners, such as supplier's address and banking information. Master data management aims to provide data and processes for collecting, matching, persisting, and distributing such data within an organization but potentially also with business partners to ensure a single source of truth that provides consistency and control.

Open ID Connect (OIDC) — A standard for federated identity management (that is often used in enterprise single sign-on and social login) in which clients (such as a mobile application or a server) can request and receive information about valid sessions. In essence, a user proves attributes attested by a dedicated identity provider (such as the company HR or IT department, or Google/Facebook) by means of a signed JSON document that contains information about their identity.

Oracle — An oracle can be used to store data from the outside world, such as the daily temperature, a flight's delay, or the number of votes a political candidate received, on a blockchain such that a data consumer or smart contracts that trusts the Oracle (or at least trusted the Oracle when the smart contract was deployed) can use the data to make decisions (adapted from https://www.coindesk.com/what-is-an-oracle and https://www.itu.int/en/ITU-T/focusgroups/dlt/Documents/d11.pdf). Oracles can build on a single trusted third party that signs the data directly or indirectly (the data was signed by a sensor that in turn was certified by the trusted third party), a consortium of third parties such that the majority is trusted (often involving multi-signatures and threshold-schemes).

(Secure) Multiparty Computation (MPC) — allows the computation of a function with multiple inputs provided by different entities where every participant does not learn the others' private inputs but only the final result. Often, MPC also involves methods to make sure that the participants perform their task correctly, e.g., through using ZKP. This is based on cryptographic primitives such as secret sharing (splitting an input into multiple parts and distributing it to other parties), and/or oblivious transfer (sending multiple messages to another party and making sure that the recipient can only decrypt exactly one of them, but the sender does not learn which one). In this sense, MPC is strictly more general than ZKP, as the latter requires a party that has access to all private inputs for a computation to create a proof of the correct execution of a program. More details on this can, e.g., be taken from (Buterin, 2014).

Original Equipment Manufacturer (OEM) — from VID II.

Glossary of Terms

Trusted Execution Environment (TEE) — Hardware for which the manufacturer guarantees that code is executed correctly on it ("remote attestation") and no information is leaked outside the hardware. A prominent example is Intel's Software Guard Extensions (SGX) solution. By creating a private key inside the TEE and only making the corresponding public key available, information can be encrypted in a way that makes sure it can only be decrypted inside a TEE. Besides known vulnerabilities such as side-channel attacks, challenges of TEEs include high integration efforts (particularly in the cloud) and the required trust in the TEE's manufacturer as it provides a single point of failure. On the other hand, TEEs can run basically arbitrary code in a rather efficient way, which offers advantages over cryptographic alternatives such as MPC and ZKP that typically exhibit lower performance and require the implementation of hand-crafted solutions by experts with deep cryptographic expertise.

Verifiable Credential (VC) — from VID II (W3C-oriented). There are related standards that have many commonalities with Verifiable Credentials, such as X.509 certificates that are used to authenticate servers in the world wide web (HTTPS), JSON Web Tokens (JWT) that are frequently used in federated identity management schemes such as the OIDC standard, or JSON-LD documents that may include a Linked Data Proof, such as the GS1 EPCIS 2.0 standard. In the following, we will refer to the W3C VC standard, however, more narrow or not compatible competing standards would also enable us to reach the goals of this reference implementation, with a few exceptions (not all standards support ZKPs, for example).

Verifiable Presentation (VP) — from VID II.

Zero-Knowledge Proof (ZKP) — In cryptography, a zero-knowledge proof (ZKP) is a method by which one entity (the prover, assumed to be computationally powerful) can prove to another party (the verifier, assumed to be computationally restricted) that a particular statement is true without revealing any further information. A digital signature of a challenge for authentication can be thought of as "almost" a ZKP: The verifier knows the prover's public key and wants to be convinced that the entity that wants to authenticate possesses the associated private key. By signing the verifier's challenge with the private key, the prover can convince the verifier that they possess the private key without revealing it (it is, however, not zero-knowledge as the verifier learns at least something new that they could not have computed on their own, namely the value of the signature on the nonce. However, this information will be pretty useless in any other context, so for illustration, the analogy fits). A specialized form of ZKPs is used in so-called Anonymous Credentials that are also being used in some SSI implementations, such as the Anoncreds in Hyperledger Aries and the BBS+ Linked Data Proofs in MATTR's Verifiable Credentials. They allow proving selected attributes contained in a VC or predicates derived thereof, for example, an inequality proof for an integer-valued attribute such as age or expiration timestamp. These specialized ZKPs are highly efficient, with signing and verification lasting only a few milliseconds.

In general, ZKPs can also be used to prove that a specific computation has been conducted correctly, without the need to disclose inputs, outputs, or intermediate steps. For example, using a zk-SNARK or zk-STARK, one could prove that given two Merkle roots (hashes) of different composite parts represented by Merkle trees (where the root hashes are known to the verifier and both Merkle trees are known to the prover), a specific parts ID is present in both of them, and checking this statement can be conducted by the verifier quite efficiently (in milliseconds) and without learning any additional information. However, the prover needs to know all the data for the original computation for the proof, in particular, in the given example, the prover needs to know the preimages of the hashes.

Full access to this Standard and other MOBI Standards are available to MOBI members.

If you are not part of the MOBI community and would like to become a member, please fill out our "Membership Inquiry Form" at dlt.mobi/join.

Members gain access to standards, working groups, and many other benefits. Join us in building the New Economy of Movement!

If you have any questions regarding the MOBI Supply Chain working group, please email SC@dlt.mobi.

dlt.MOBI @dltMOBI MOBI @dltMOBI SC@dlt.MOBI