



appliedblockchain

K0 BACKGROUND

**A vision for interoperable
enterprise blockchains
using zero knowledge proofs**



Created by
Applied Blockchain Ltd

July 26, 2019

V0.1

Table of contents

Executive Summary	3
What was the motivation behind building K0?	3
Isn't someone else already doing this?	3
Principles & assumptions	4
A brief history of blockchain data privacy	5
Encrypted data on chain	5
Isolated "sub-chains"	5
or "channels"	5
Hash of data on chain	5
Hardware secure modules	5
Blockchains consortia problems	6
Solving consortia interoperability	7
Incompatible enterprise platforms	7
Introduction to zero knowledge proofs	8
Zero knowledge proof schemes	8
Solving enterprise data privacy and interoperability using zero knowledge proofs	9
Solving data privacy first.....	9
...and interoperability second	9
Introducing K0	11
Vision: ZK Roadster	12
Terminology	14

Executive Summary

K0 sets out to enable general purpose private digital assets usable across multiple enterprise blockchain platforms and networks.

What was the motivation behind building K0?

We set out to solve two problems:

1. We felt that most enterprise blockchain platforms solve privacy at the expense of security (number of parties validating transactions).
2. We felt that diverging enterprise blockchain technologies are creating a technology integration gap that will become harder to bridge, and we believe that networks will need bridging in order to maximise business value and potential.

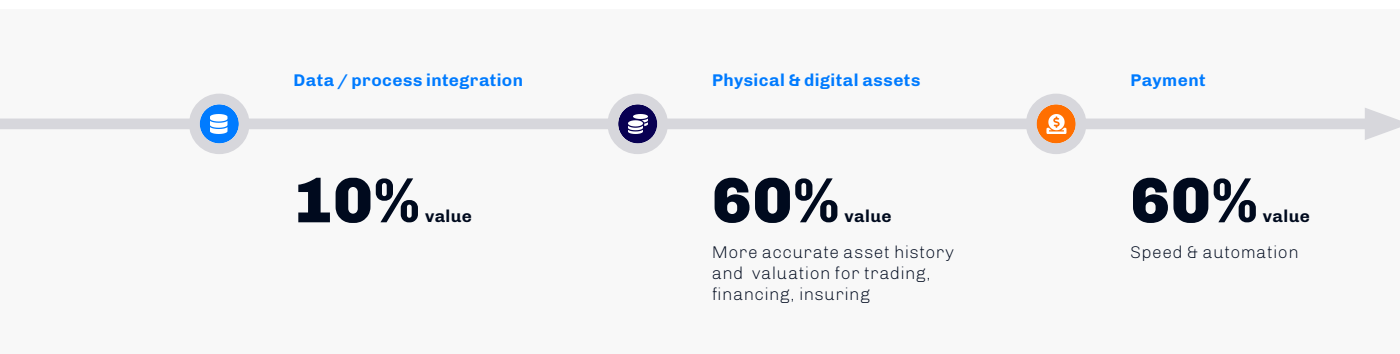
Isn't someone else already doing this?

We looked around, and noticed that:

1. Blockchain platforms started using zero knowledge proof technology to improve 1, but not 2.
2. There was already some great work on privacy and zero knowledge proofs for public blockchains, but not private, business networks that our clients use.

Principles & assumptions

The following principles, beliefs and assumptions guide the design proposed in this paper.



1. The greatest value of blockchain technology in business is in providing a more reliable, and secure record of asset ownership and history, for fungible assets (e.g. payment tokens) and non-fungible assets (e.g. unique records of physical or digital assets, commodities, certificates and securities). The technology, if implemented correctly, reduces the risk of fraud in asset histories, drives more accurate valuations, reducing risk in asset trade, financing and insurance.
2. Smart contracts automate the movement of assets, enforce the rules guiding asset movement, synchronise business processes, and will likely automate some aspects of legal contract execution in the future.
3. Solutions built on enterprise blockchains should be platform agnostic.
4. Enterprise blockchains should be designed to interoperate.
5. Enterprise blockchain solutions should also be designed to integrate with public blockchain networks in the future (should these begin to satisfy compliance requirements, and prove more secure and desirable).
6. Privacy is always a requirement, including the privacy of commercial activity, as well as individual and organisation identity, while allowing for some level of audit and compliance as required.
7. The blockchain technology layer will become simplified and commoditised, in the form of an efficient, scalable consensus ledger of transactions. More complex constructs, such as assets, data, business logic, and privacy will move up into an interoperable layer based on zero knowledge proofs.

A brief history of blockchain data privacy

Blockchains use cryptographic techniques to secure a history of transactions, and the current state. However, data in a blockchain is typically transparent (unencrypted, clear text) and held by all of the nodes in the network.

There is a requirement, especially in business blockchain environments, to restrict access to commercially sensitive information by nodes in a network that act as validators, but are not active participants in a particular

transaction. Sensitive information includes identity of the transacting parties, assets, prices, amounts exchanged, rules enforced (e.g. smart contract code), and potentially any other data sent to the blockchain.

Numerous solutions have been proposed and developed to enable data privacy in blockchains and smart contracts. A short summary is provided below.

Encrypted data on chain

This technique involves storing data in encrypted form on chain, potentially sharing keys with one or more selected counterparts to enable only them to decrypt the data. A disadvantage of this technique is that a blockchain smart contract cannot apply rules to act on the data, as the validators that would normally execute the smart contract code only have access to the data in encrypted form.

Hash of data on chain

This technique involves sending and storing only hashes of data to the blockchain. The advantage is that data is not stored on chain, and therefore not shared with validators. A disadvantage of this technique is that a blockchain smart contract cannot implement rules that act on the data, as it only has access to the hash. The blockchain is therefore limited to providing a "timestamping" function, rather than providing an asset registry and enabling business logic through smart contracts.

Isolated "sub-chains" or "channels"

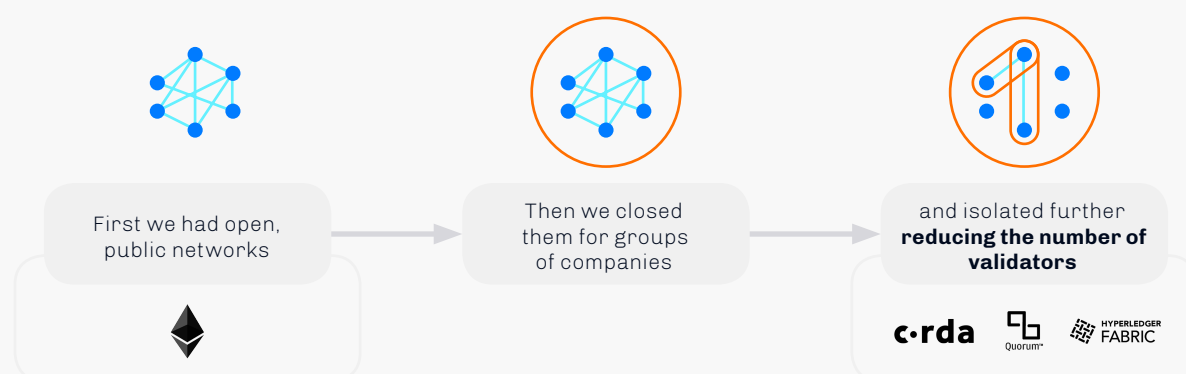
Enterprise blockchain platforms have adopted an isolation approach whereby data, including encrypted versions of the data, is only shared with counterparts to a particular transaction.

Implementations typically involve "mini-blockchains" (also known as "channels", or "private transaction") for each private transacting group. A disadvantage of this approach is that only a very small subset of the network validates the smart contract execution and secures the transactions history. This technique removes the "group security" that blockchain was designed to provide.

An additional disadvantage of this approach is that assets and their histories are no longer easily shared and secured through a single ledger, and must therefore be communicated separately in their entirety, less efficiently and ultimately less securely than in classic blockchain implementations.

Hardware secure modules

An additional avenue of research in blockchain privacy involves hardware secured modules (HSM) that are hosted by independent parties in a network, where their hardware security prevents the hosting party from accessing the blockchain data and smart contracts that are being executed and verified. This enables a large group of validators to validate transactions without having the ability to access the underlying data. A disadvantage of this approach is reliance and trust on the hardware manufacturer, and security vulnerabilities have already appeared in common implementations.



Blockchains consortia problems

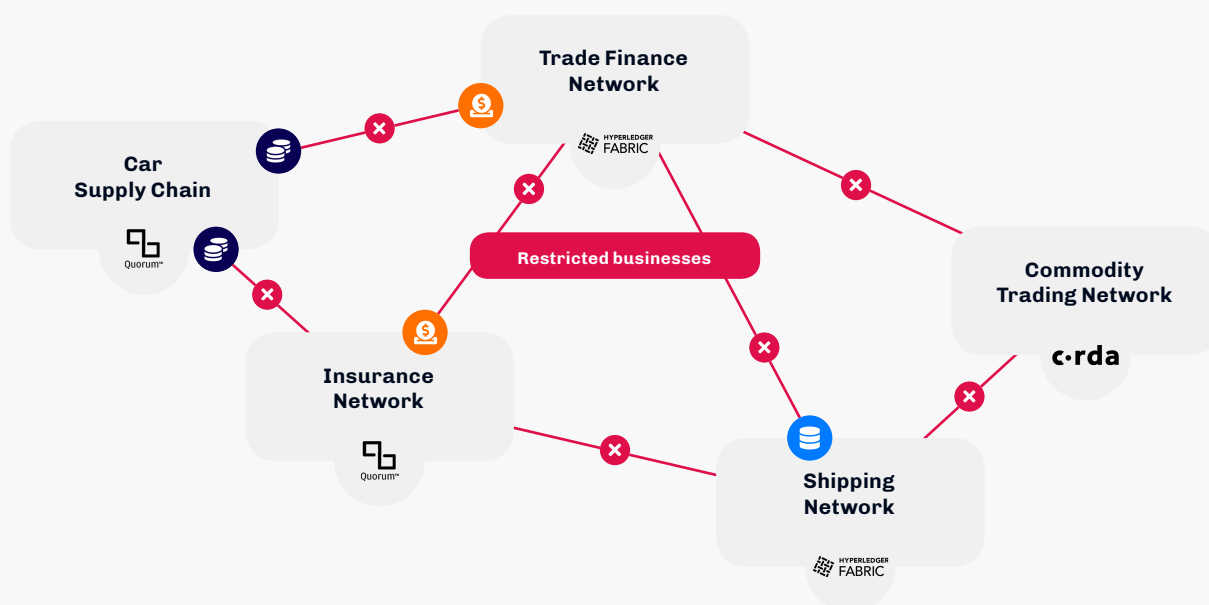
Enterprise blockchain networks are evolving inside large organisations and in industry groups of large organisations also known as consortia.

A problem arises when business activity needs to flow across these pockets of consortia. This is typically the case where the same assets need to be referred to in different blockchain networks. In the same way as a blockchain network ensures uniqueness of an asset,

so assets that cross consortia boundaries must remain globally "unique" otherwise there exists a risk recording multiple digital versions of the same asset, and potentially conflicting histories.

For example:

1. A supply chain where a digital twin of a physical component (the "asset") is recorded in one network, is purchased by an organisation that is active on another network, and the unique record must be "transferred" to the other network in order to continue its "journey" and continue to provide full provenance.
2. An insurance network comprised of insurance companies (and their business logic smart contracts) looking to insure an asset originated and residing on a different blockchain consortium network (e.g. a commodity, a physical product, a ship, a container, a car).
3. A finance network looking to finance, and potentially securitise a physical or digital "asset" that originated and resides on a different blockchain consortium network (e.g. a commodity, a physical product, a ship, a container).



In these examples we have focused on digital representations of physical assets ("digital twins"), although the same applies to movement of payment tokens, certificates, documentation, and any other unique data.

Solving consortia interoperability

The following options spring to mind for dealing with assets on different blockchains:

1. Simply read asset data from the other blockchain. This will not suffice, as the provenance of the asset must be updated, the asset record must be globally unique in order to avoid conflicting records, and any changes to the asset (e.g. applying an insurance policy or financing agreement, and potential change of ownership where circumstances arise in an insurance and/or financing policy) must only be registered in the ledger holding the unique digital representation of the asset at that point in time. So this is not a good option if we need to update the unique record of the asset.

2. Second consortium blockchain members write changes to an asset in the first consortium blockchain where the asset resides. The blockchain business logic (smart contracts) in one ledger cannot simply be applied to assets residing on another ledger. If the organisations themselves simply write directly to the asset on the first blockchain, they will be ignoring and bypassing their own consortia solution, and likely duplicating logic and creating a world of spaghetti integrations, where each organisation must be connected and deploy its own business logic to numerous blockchain networks. So this is not a satisfactory solution either.

In a perfect world, all organisations would be using the same network and could easily interoperate.

In reality, numerous organisations have invested in consortium networks and they are here to stay. We face a world of independent consortium networks and solutions that will need to interoperate ("chains of chains") in order to fully realise collaborative business opportunities and benefits.

Incompatible enterprise platforms

An additional problem is that enterprise consortia in general have chosen to use technically very different and fundamentally incompatible enterprise blockchain technology platforms. This has led to assets and associated business logic (smart contracts) expressed using incompatible technologies.

There are a number of interoperability solutions in development, primarily focused on the public blockchain networks. However, these generally begin to solve technical

interoperability, asset transfer, atomic swaps, without privacy, a fundamental requirement in business blockchain solutions.

This paper proposes using a new technology known as "zero knowledge proofs" to express assets and their business logic in a way that enables interoperability across existing (and future) enterprise blockchain platforms used by enterprise consortia, while at the same time guaranteeing absolute privacy.

Introduction to zero knowledge proofs

Conceived two decades ago in computer science academia, and brought to life in the zCash public blockchain implementation launched in 2016, this is an emerging set of cryptographic tools that enable mathematical proofs of certain properties of data (e.g. asset ownership, value range) to be shared and verified by other parties without the need to reveal the underlying data.

Zero knowledge proof schemes

A number of early zero knowledge schemes have emerged, namely zkSNARKs, sonics, bulletproofs and zkSTARKs (and even zkSHARKS). Zero knowledge proofs generally require three steps: circuit creation typically only performed once to create a "template" for a proof scheme, proof creation, proof verification. Each of the methods named above requires different levels of computation and size at each of the three stages, making them more efficient or suitable for various use cases. For example, zk-SNARKs require extensive computation for proof creation, but very little computation for proof verification,

which is a balance that works well for off-chain creation of proofs, and on-chain verification, where verification on chain is performed by a distributed network of nodes executing all of the transactions on the network. The circuit creation phase in zk-SNARKS produces what is known as "toxic waste", which includes artefacts that could be used to "forge" proofs (there is a workaround in the form of a multi-party computation ceremony). Bulletproofs and zk-STARKS do not produce toxic waste, but produce validation proofs that are larger, require more memory and/or take longer to compute.

Solving enterprise data privacy and interoperability using zero knowledge proofs

Solving data privacy first...

This paper proposes expressing enterprise digital assets, their properties, and the rules that govern their movement using zero knowledge proofs, rather than traditional blockchain smart contracts.

In this case, blockchain validators store and validate proofs of certain properties of the assets and data rather than the data itself, and the risk of sharing and revealing commercially sensitive information to parties not directly involved in a transaction, including the identities of the parties, identities of the assets, quantities of the assets and data properties of the assets, is removed. Such data is not even available in encrypted form.

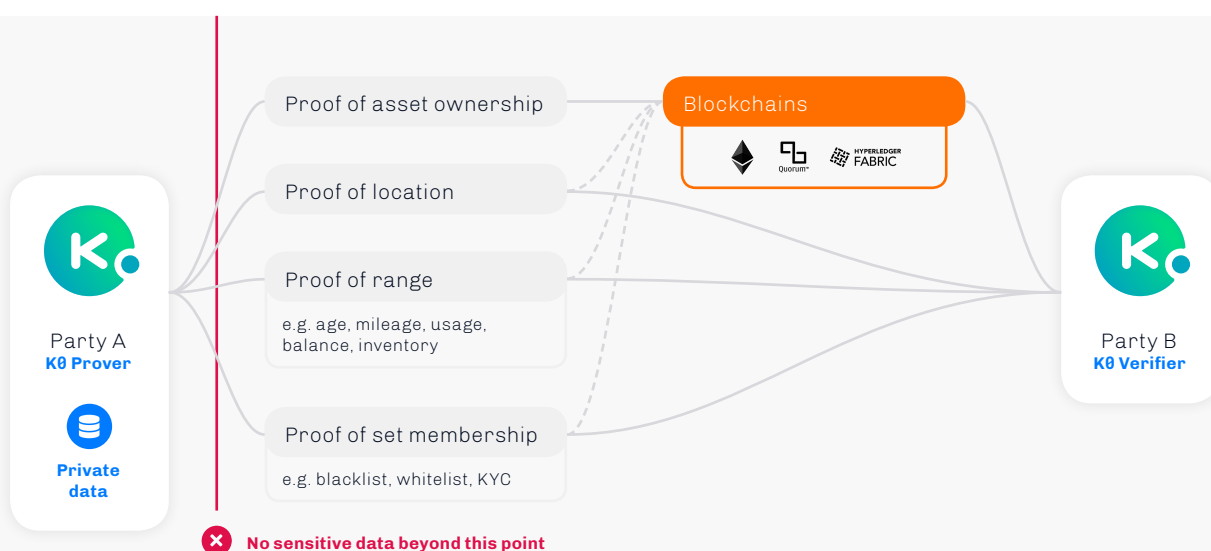
This enables validation over a much broader network of validating nodes in a blockchain network without revealing commercially sensitive information, thereby increasing security compared to "channels" and "private transaction managers" implemented in the major enterprise blockchain platforms today.

We propose continuing to use the existing enterprise platforms (and more efficient platforms in the future) as a base blockchain consensus layer, while expressing assets and rules as zero knowledge proofs, rather than using the proprietary, incompatible smart contract languages, and restrictive, proprietary privacy schemes.

...and interoperability second

By expressing assets, their histories, and their associated business rules using zero knowledge proofs, we effectively decouple them from the proprietary underlying blockchain technology platform.

In practice this means that the proof of asset ownership, history, and associated business logic can be generated independently, and off chain, and verified on any of the existing enterprise blockchain platforms (subject to their support for various math and cryptographic functions, which is becoming standard).



Portable assets

The decoupling of the assets and their business logic from the underlying technology platform, opens up the potential for them to become platform agnostic and portable. This means that as long as the unique asset only resides on one blockchain network ledger at a time (to prevent double spend), we can potentially move the asset across different ledgers that were built using different technologies while maintain absolute privacy.

Private networks use a group-security consensus mechanism to protect from an attack by a minority of rouge participants. However, when assets are transferred from one network to another it is assumed that the overall blockchain networks trust each other, that is, one overall "group" (or "majority") of participants in one network trust the other overall "group" (or "majority") of participants on the other network.

Introducing K0

K0 is a zero knowledge proof framework for digital assets and data that are private and portable.

If zero knowledge proofs are to replace Turing complete smart contract languages, as we propose, they must enable equivalent functions to be performed.

At this conjecture, zero knowledge proofs (and zkSNARKS in particular) require definition of predefined circuits of logic. We must therefore preselect and compile the generic business logic that is required.

In the principles & assumption section above we described the primary function of smart contracts to "automate the movement of assets, and enforce the rules guiding asset movement". In K0 we begin by defining a basic set of zero knowledge functions for handling fungible and non-fungible assets and the rules governing their movement and exchange:



It should be noted that K0.1 only implements a subset of these functions.

Vision: ZK Roadster

As a simple example, imagine a car product lifecycle.

Initial vehicle registration

Vehicle manufacturer ZK Cars produces its latest vehicle the ZK Roadster, and uses K0 to register each vehicle privately as a unique non-fungible asset on an automotive industry consortium chain.

The vehicles are registered privately by the manufacturer in order to prevent leakage of commercially sensitive information such as the volume of vehicles manufactured on a given day, and subsequent activities such as data of first sale, geographic destination of each vehicle etc.

The entire automotive industry consortium is able to validate and secure the transaction to register the vehicle on the blockchain, but none of the validators are able to understand the context of the transaction nor the data behind it (that this is a vehicle, the identity of the manufacturer, etc.). The transaction is therefore secured by a large group of validators, without compromising the privacy of the transaction nor the identity of the manufacturer.

Purchase

The vehicle is subsequently purchased from the manufacturer by an individual buyer. A trade to exchange ownership, including tokenized payment, is executed privately on the automotive industry consortium chain. The buyer is able to verify that the vehicle is original, and was produced and is owned by the manufacturer, and has had

no previous owners, is not blacklisted by the authorities, and the mileage is zero. The purchase transaction is executed on the automotive industry consortium network in absolute secrecy, while being validated, and therefore secured, by the entire network of industry validators.

Mileage

The new individual buyer enjoys driving the ZK Roadster and uses it extensively, clocking up almost 30,000 miles in the first year of ownership. Sensors on the vehicle regularly generate and sign the mileage data, and K0 is used to create zero knowledge proofs of the vehicle

mileage and record these in the automotive industry consortium chain. The updates are validated by the entire network, although none of the validators know the identities of the driver, the vehicle, the mileage, nor the type of data recorded.

Conditional Sale

The individual buyer notices that ZK Cars have just released a brand new model, and wishes to sell their current ZK Roadster vehicle. They use K0 to create a conditional sale proof for their ZK Roadster in exchange for £20,000. This gets recorded in the automotive industry consortium

chain. As before, the updates are validated (and therefore secured) by the entire network, yet none of the validators know the identity of the vehicle, nor the context of the update.

Conditional Purchase

A buyer agrees to purchase the ZK Roadster on condition that the claims about the asset are correct. For example, that it is an original vehicle purchased directly from the manufacturer, and has clocked less than 40,000 miles. A conditional purchase proof is generated by the buyer

using K0 and posted on to the automotive industry consortium chain. As before, the updates are validated (and therefore secured) by the entire network, yet none of the validators know the identity of the vehicle, nor the context of the update.

Conditional Execution / Atomic Swap

The conditional sale and conditional purchase are privately verified, matched and the assets (car and payment tokens) exchanged in a smart contract execute by all of the validators on automotive industry consortium chain. The validators on the network will only enable this transaction

if all of the conditions are met, yet none of the validators know anything about the underlying data, including the identities of the parties, the assets, the price, the conditions, or any of the asset properties.

Insurance

The new owner that just purchased the vehicle wishes to use insurance services from an insurer that is part of an insurance consortium blockchain. The insurer wishes to reduce risk and enjoy the operational efficiencies of providing the insurance through a blockchain smart contract that has been deployed to the insurance consortium blockchain network of which they are a member and validator.

The insurance smart contract includes business logic for verifying vehicle authenticity, ownership and vehicle history. The insurance contract will also need the ability to take ownership of the vehicle if it is involved in an accident and repair is not practical, or the vehicle is stolen, replaced, and then recovered.

In order to apply this type of logic, the insurance smart contract running on the insurance consortium network needs to access to the car asset that currently resides on the automotive industry consortium chain. The reason it needs to access the asset on the other blockchain is to enable the insurance smart contract to take ownership under certain business circumstances.

A bridge is required to connect the two networks where assets can be transferred across or "locked" by the other network. In our case, we'd like the insurance contract in the insurance consortium network to lock, with the option to transfer and take ownership of the vehicle asset in the automotive consortium network.

Under normal circumstances each consortium will use its own technology platform, so in our example the automotive consortium uses Quorum (Ethereum), and the insurance consortium uses Hyperledger Fabric.

The automotive consortium uses K0 to record the vehicle assets and history on their Quorum blockchain. The insurance smart contract on Hyperledger Fabric includes a component to verify the vehicle assets created using K0. Through a platform bridge, the insurance company can also generate and deploy the proofs required to "lock" the asset, and even conditionally transfer and take ownership of the asset on the insurance chain if required.

Terminology

This section provides brief descriptions of the terminology used throughout this paper.

Blockchain

A blockchain is a group-secured distributed and decentralised database nodes. Blockchains use a cryptographically linked chain of blocks of historical transactions, agreed across a network of nodes through a consensus mechanism to secure historical transaction records associated primarily associated with asset ownership and movement, and to prevent double-spend.

Smart contracts

A smart contract is custom code that is executed on a blockchain network. Smart contracts are used primarily to record asset ownership, and implement rules related to asset movement.

Zero Knowledge proofs

In cryptography, a zero-knowledge proof or zero-knowledge protocol is a method by which one party (the prover) can prove to another party (the verifier) that they know a value x , without conveying any information apart from the fact that they know the value x . The essence of zero-knowledge proofs is that it is trivial to prove that one possesses knowledge of certain information by simply revealing it; the challenge is to prove such possession without revealing the information itself or any additional information.



appliedblockchain

Address





Runway East
20 St Thomas St
London SE1 9RG,
United Kingdom

info@appliedblockchain.com

© Copyright 2018 Applied Blockchain Ltd.
All rights reserved.

Applied Blockchain Ltd is a company registered
in England and Wales. Company No. 09686276.

Follow us on

-  github.com/appliedblockchain/
-  linkedin.com/company/applied-blockchain/
-  medium.com/@AppBlockchain_
-  twitter.com/appblockchain