2nd Symposium on Distributed Ledger Technology

05 July 2018

Gold Coast Campus, Griffith University

Institute for Integrated and Intelligent Systems
Griffith University
170 Kessels Road
NATHAN QLD 4111
Australia

July 2018

Welcome!

On behalf of the Organising Committee, we would like to welcome you at the 2nd Symposium on Distributed Ledger Technology (SDLT 2018) that will be held at Gold Coast Campus, Griffith University, Australia on 5th July 2018. With great success of the last year's the one-of-the very first symposiums of its organised in Australia, this event covers technical, legal, regulatory, and societal aspects of the innovative distributed ledger technology and its applications.

The SDLT 2018 program features a keynote address, a number of invited talks, technical sessions, and panel discussion. We have a great line-up of speakers and registered participants, who are some of the world leading researchers and practitioners in this area from academia and industry.

This forum provides an excellent opportunity for sharing the latest development on the promise of enabling 'new deals on data' from different angles, and for collaborating on future projects.

Symposium Co-Chair

Assoc. Prof. Vallipuram Muthukkumarasamy

Prof. Jin-Song Dong

Organising Committee

Dr. Xin-Wen Wu

Dr. Wee Lum Tan

Dr. Kamanashis Biswas

Dr. Zhe Hou

Symposium Program (G11_3.61/3.62)

Time	Session	Speaker
8:15am	Registration and coffee on arrival	
8:40am	**Opening Session**	Chair: Prof Jin Song Dong, Director, Institute for Integrated and Intelligent Systems
	Welcome address	Prof Andrew Smith, Pro-Vice Chancellor, Griffith Sciences
8:45am	Keynote address Blockchain: Riding the Rollercoaster towards a Revolution	Dr Adrian McCullagh, ODMOB Lawyers
9:25am	**Technical session 1**	Chair: Prof Bill Caelli, Griffith University
	On Legal (Smart) Contracts and Blockchain Systems	Prof Guido Governatori, Regis Riveret, Xiwei Xu, Zoran Milosevic, Florian Idelberger, Giovanni Sartor (Data61)
	Decentralised Random Number Generation	Peter Robinson, Technical Director & Applied Cryptographer, ConsenSys
10:05am	Morning tea	
10:25am	**Technical session 2**:	Dr Raja Jurdak (Data61, CSIRO)
	The Power and Possibilities of Blockchain for the Enterprise	Niki Ariyasinghe, Head of Partnerships Asia-Pacific, R3
	On the Interoperability of Distributed Ledgers	Dileban Karunamoorthy, Technical Lead – Blockchain (IBM Research Lab, Melbourne)
	Blockchain for Transparent Food Supply Chains	Sidra Malik (UNSW), A/Prof Salil S. Kanhere (UNSW), Prof Raja Jurdak (CSIRO)
	Validating Smart Contract Execution Across a Heterogrneous Collection: A proposal	Dr Padmanabhan Krishnan (Oracle Labs), Babu Pillai (Griffith), Dr Kamanashis Biswas (Griffith)
	Towards Formal Verification of Solidity Smart Contracts Using PAT	Christopher Skorka, Lee Goymer, Dr Hadrien Bride, Dr Zhe Hou and Prof Jin Song Dong, (Griffith University)
	Blockchain-based Booking System - Design and Model Checking	Lung-Chen Huang, Naipeng Dong, Guangdong Bai, Siau Cheng Khoo, Prof Jin Song Dong (Griffith University)
12:25pm	Lunch	
1:25pm	**Technical session 4**:	Chair: Peter Robinson, ConsenSys
	Blockchain Platforms for IoT Use-cases	Mohammad Chowdhury (Swinburne University), Dr Md. Sadek Ferdous (Imperial College), Dr Kamanashis Biswas (Griffith University)
	Distributed Business Process Flexibility on the Blockchain	Silvano Colombotosatto, Dr Nick van Beest, Prof Guido Governatori, Regis Riveret (Data61)

	The case for DLT in Healthcare – real or hype?	Kris Vette (Vette Solutions Ltd)
	Vacci-Chain: The Smart Contract	Dr Kamanashis Biswas, Mr Thomas Csere, Dr
	Powered Vaccine Storage and Monitoring Solution	Wee Lum Tan, A/ Prof Vallipuram Muthukkumarasamy (Griffith University)
3:00pm	**Invited Talk 1**:	Chair: Prof Ron van der Meyden (UNSW)
	Blockchain Deconstructed	Prof Fritz Henglein (University of Copenhagen and Deon Digital AG)
3:25pm	Afternoon tea	
3:45pm	**Invited Talk 2**:	Chair: Dr Kal Singh, Itron Inc
	Implementation Experiences of Distributed Ledgers: The Tension between Data Sharing and Privacy	Prof Peter McBurney (King's College London)
4:25pm	**Panel discussion** Challenges and Opportunities for DLT Application and Future Directions	Chair: A/ Prof V. Muthukkumarasamy (Griffith University) Prof Ron van der Meyden (UNSW), Dr Paul Ashley (Anonyome), Paul Gampe (PCCW Global), Kanwar Singh (Public Trustee), Dileban Karunamoorthy (IBM)
4:55pm	Closing remarks	Prof Jin Song Dong, Director, Institute for Integrated and Intelligent Systems

Keynote Address

Blockchain: Riding the Rollercoaster towards a Revolution

Dr. Adrian McCullagh, ODMOB Lawyers
amccullagh@odmoblawyers.com

Abstract

Nakamoto proposed a new solution to transact value via the internet. The internet prior to the advent of bitcoin was primarily a communications environment for non-face-to-face interaction. In order to carry out a commercial transaction it was necessary to involve some third party who would validate the financial aspects of the transaction. The heart of the bitcoin solution was the blockchain construct.

The blockchain as originally proposed with its proof of work consensus protocol has been shown to have some uncommercial aspects to which researchers globally are attempting to solve. Further, since the blockchain is a distributive environment, commercial compliance requirements can impact the architecture of a blockchain. The architecture of a blockchain must meet the regulatory compliance which could be industry specific such GDPR, financial and health regulatory obligations.

An important issue with the development of new technology that has international reach is that such technology should not become siloed. This is where standards especially international standards can assist. Of course, standards by themselves will not necessarily obviate the impediments to interoperability, but if standard interfaces and standard communication structures can be developed then the uptake of blockchain environment will be more likely to be achieved, which could financially benefit the global economy.

This paper will look at some of the issues confronting the further development of blockchain technology. The following are of some importance:

- **Blockchain Governance.** This issue is impacted by both internal and external factors.
  - Internal Governance
    Concerning the issue of internal governance, the SEGWIT issue last year for bitcoin was an example of the tragedy of the anti-common. The tragedy of the anti-common arises when multiple stakeholders can impede a resolution. Much like what occurs in the UN security Council with any of the permanent members being able to veto any issue put forward by any other member. The role of the core coders and miners caused many issues in 2017.
  - External Governance:
    External Governance primarily concerns data governance where multiple instances of information can be spread across multiple parties who may be party of a consortium. Many compliance issues arise from a regulatory perspective such as security, and GDPR.
- **Smart Contracts**
  A smart contract is some code that is permitted to write to the blockchain. In the Ethereum environment the code will be stored in the blockchain itself. This results in some interesting issues. Contracts can be classified as either an instantaneous contract (also known as an executed contract) or a longitudinal contract (also known as a executory contract). Instantaneous contracts are contracts that in effect conclude at the time of their creation; like the purchasing of groceries, whereas a longitudinal contract involve some post entering performance like a contract for the delivery of some services such as a software development contracts. Another example of a longitudinal contract would be a mortgage agreement which requires the regular repayment of principal and interest over a period of time. But it should be noted that the term smart contract is a misnomer. For the most part the code is neither smart nor does it represent a contract.
- **Interoperability**
  There will not be a single blockchain unlike the internet; though initially some multinational IT organisations did try to highjack the internet in its informative development. Being multiple blockchain environments being created such as Cardano, NEM, Ethereum, Hyperledger and others it is important that siloed structured are not the norm. Solving interoperability standards should be a high priority.

The above can be assisted through the development of appropriate standards, but it will take time and that is the greatest impediment for standards development as they take time and blockchain development waits for no committee.

Invited Talk 1

Blockchain deconstructed (abstract)

Fritz Henglein
Department of Computer Science, University of Copenhagen (DIKU) and *Deon Digital AG*
henglein@diku.dk, henglein@deondigital.com

***Abstract*—We propose that blockchain/distributed ledger (DL) systems be characterized by three simultaneous general requirements: organizational and technical decentralization; tamper-proof recording of events and their evidence; and guaranteed resource (= asset) preservation. Including evidence extends blockchain/DL systems to serving as digital twins for physical processes and resources. Resource preservation generalizes the "no-double-spending" property to allowing dynamically adjustable and user-specific credit limits and having multiple, user-definable resources.**

We formulate a simple theorem that highlights that enforcing credit limits is essentially the *only* problem requiring more than point-to-point communication. In particular, without credit limit enforcement essentially all communication between authenticated parties in a (smart) contract can be kept completely private. Conversely, *some* privacy leakage to a third party is necessary for credit limit enforcement. This naturally gives rise to a lightweight architecture for permissioned blockchain/DL systems where all communication between parties is "off-chain" (= point-to-point or in separate private channels for multi-party contracts) and only resource transfers need to be validated by a decentralized system employing a suitable distributed consensus protocol. We point out that such consensus protocol need not reach agreement on a globally total order of transactions, which is the main cause of inefficiency in presently popular blockchain/DL systems, since resource transfers commute with each other and thus can be processed in any order with limited synchronization: only credit limit enforcement requires some communication amongst the on-chain nodes.

I. Elaboration

In terms of the REA accounting modeling [1]–[3], a blockchain/DL system records events such as transfers of *resources* and *information* between agents. The difference between resources and information is that the former must not be duplicated, whereas the latter can be freely copied. The system thus guarantees the invariant that the *sum* of all resources owned by anybody is invariant under *transfers*: transferring 50 ETH from account A to account B does not change the total amount of ETH. The system furthermore guarantees the no-double-spend property: the transfer is only *valid* and effected if account A contains at least 50 ETH; that is, A's balance must be nonnegative at all times. In other words, the no-double-spend property amounts to enforcing a credit limit of 0 on all accounts.

It is worthwhile keeping resource preservation separate from credit limit enforcement for two reasons. First, without credit limit enforcement no validation and thus no consensus amongst more than the involved parties is required.

Theorem: Assume all accounts have no credit limit. Let T be a set of resource transfers. Then all $t \in T$ are valid and commute with each other, that is they can be performed in arbitrary order.

In particular, if two authenticated agents agree on a contract involving resource transfers such as a loan agreement, they only need to have local communication: they need to agree on the sequence of events, including transfers, that have happened at any given point in time by sending signed messages and acknowledging their receipt. In case of disagreement a party to the contract can provide the cryptographically hashed sequence of signed message exchanges to a third party as tamper-proof evidence of the history of events. Note that tamper-proof recording does not require validation by a third party.

Second, nonzero credit limits can be agent-specific and context-dependent. For example, an airline may sell (transfer) more flight tickets or a car manufacturer more cars than it presently has in storage if it manages to produce them (just in) time. Or one designated agent—the central bank—may have a dynamic credit limit of *digital cash*, a fiat currency managed as a cryptocurrency on a blockchain/DL system. If all other agents have a zero credit limit this represents a full reserve system. If designated other agents—banks—have policy-controlled non-zero credit limits, this corresponds to a fractional reserve system. In both cases, cryptocurrency cannot only be issued, but also retired, for example as part of loan repayments.

The analysis suggests a blueprint for generalized permissioned blockchain/DL systems that are highly scalable: A distributed consensus network validating *only resource transfers*; all other messages are point-to-point and private, employing standard encryption and authentication technology such as TLS. The consensus network furthermore only needs to solve a simplified consensus problem: it need not agree on a total order of transactions nor even on a partial order; it only needs to ensure that the transfers its nodes validate are guaranteed or sufficiently unlikely to eventually violate the individual agents' credit limit requirements.

References

[1] W. E. McCarthy, "The REA accounting model: A generalized framework for accounting systems in a shared data environment," *The Accounting Review*, vol. LVII, no. 3, pp. 554–578, July 1982.

[2] J. Andersen, E. Elsborg, F. Henglein, J. G. Simonsen, and C. Stefansen, "Compositional specification of commercial contracts," *International Journal on Software Tools for Technology Transfer (STTT)*, vol. 8, no. 6, pp. 485–516, November 2006.

[3] F. Henglein, K. F. Larsen, J. G. Simonsen, and C. Stefansen, "POETS: Process-oriented event-driven transaction systems," *The Journal of Logic and Algebraic Programming*, vol. 78, no. 5, pp. 381–401, 2009.

Invited Talk 2

Implementation Experiences of Distributed Ledgers: the Tension between Data Sharing and Privacy

Peter McBurney
Department of Informatics, King's College London, Strand London UK
peter.mcburney@kcl.ac.uk

The technology of blockchains and distributed ledgers has now moved beyond hype into realworld commercial implementations and deployments, particularly in banking, insurance, transport and energy. For instance, in June 2018 a consortium involving shipping company Maersk, consulting firm EY, insurance company MS Amlin and insurance broker XL Catlin announced the joint creation of a global distributed ledger platform for shipping insurance with the aim of permitting near-real-time updating of insurance cover and claims payments as ships change their travel trajectory en route.

A common benefit of many planned applications of distributed ledger technologies (DLT) is to support the single entry and validation-at-entry of shared data. In many industries, considerable time and resources are spent in reconciling data that has been entered into multiple systems and databases, and this reconciliation activity has proven difficult to automate. Having a shared database could eliminate the need for much of this reconciliation effort. An obvious question then is why not a single centralized database to hold any shared data. One reason is that such a centralized database would need to have many of the key features of blockchains –- eg, strong proof-of-identity systems such as public/private-key infrastructures; joint protocols agreed between the participants for updates to the database; and means to ensure immutable storage of past records.

Most (although not all) proposed commercial platforms are being created as closed, or permissioned, systems, where access rights will only be granted to approved organizations with a legitimate business interest in the application. In these systems, the data on the platform will not be public, but will be accessible only to the approved participants. However, many application domains currently have complex requirements and protocols for data sharing between companies, and between companies and regulators, with the result that access to particular data items may be dynamic and differentiated. In other words, participants may only have access to some data at certain times for certain purposes, and not at other times or for other functions. There may be regulatory reasons for these constraints, for example, in the widespread prohibition under anti-collusion laws against the sharing of pricing data between competitors. Transaction data is often also commercially sensitive and companies wish to control the access of others to it.

These requirements create a tension between the need for appropriate access rights control on the one hand and the openness of shared databases and DLT on the other. I will discuss these issues and the challenges this conflict creates for the design and creation of DLT systems.

On Legal (Smart) Contracts and Blockchain Systems

Guido Governatori, Régis Riveret, Xiwei Xu
Data61 - CSIRO
Australia

Zoran Milosevic
Deontik
Australia

Florian Idelberger, Giovanni Sartor
European University Institute
Italy

***Abstract*—The aim of the project is to provide an analysis of how concepts pertinent to legal contracts can influence certain aspects of their digital implementation through smart contracts, as inspired by recent developments in distributed ledger technology. It discusses how properties of imperative and declarative languages including the underlying architectures to support contract management and lifecycle apply to various aspects of legal contracts. The investigation is pursued in the context of several blockchain architectures. While imperative languages are commonly used to implement smart contracts, we find that declarative languages provide more natural ways to deal with certain aspects of legal contracts and their automated management.**

I. Introduction

The concept of a contract is used in business, commerce and everyday life to capture any agreement between parties, and to govern their interactions. We shall speak of legal contacts to specifically denote agreements having legally binding effects.

Many commercial computer systems often labeled as 'contract management applications' have been developed to support automation of legal contracts. There, the term 'e-contract' has been used to refer to an electronic representation of a contract, suitable for contract automation activities, and there have been standardization initiatives concerning e-contracts.

The term 'smart contract' was initially proposed in the early 90s for e-commerce applications but has recently been widely used in the context of distributed ledger technologies and in particular blockchain technologies. In this context, a smart contract is any self-executing program running in the distributed ledger environment, and it is often meant to implement automated transactions agreed by the parties.

While not every smart contract has legal significance, many smart contracts are linked to legal contracts, namely with agreements meant to have legal effect. We may distinguish two cases. In the first case, a separate agreement, expressed in natural language, may exist between the parties, and the smart contract may be meant to implement automatically the content of that agreement. In this case the smart contract may provide evidence for the existence and the content of the agreement between the parties, while automating its execution. In the second case, when no other document exists recording the agreement of the party, the smart contract itself embodies the binding expression of that agreement. In this case, on which we shall focus on this paper, the smart contract itself is meant both to have certain legal effects and to implement them automatically.

Distributed ledger systems can support the implementation of a smart contract with regard to both storage and automated execution. Further, the availability of digital currencies enables the automated execution of money transfers, as needed to implement the contract. Hence, distributed ledger systems constitute computational platforms offering integrated services to run large numbers of smart contracts. As distributed ledger systems are operated by collectives, they may disrupt conventional organisations accommodating trusted third-parties. They open up new opportunities for automated agreements, leading to a wide range of useful applications, but also raising important legal issues.

Some programming languages for smart contracts may be more suitable than others to facilitate legal interactions. While imperative languages are often used to code smart contracts, declarative languages may be interesting alternatives. Declarative smart contracts, and in particular logic-based smart contracts, could provide advantages in representing smart contracts and reasoning upon them. For example, they can be more compact than their imperative counterparts, they can be easier to draft, their properties can be formally verified, parties may easily understand the content of the contract and its implications. These are some common arguments supporting the use of declarative languages to encode smart contracts, but there are also questions as to whether these arguments really hold in the context of distributed ledger.

The project aims to address the conceptual connection between legal contracts and smart contracts. It investigates the legal and technical issues relative to the use of smart contracts expected to have legal effects, and thus it contributes to the implementation and use of smart contracts as legally binding agreements. More specifically, we compare imperative and declarative smart contracts in order to understand their respective (dis)advantages, from a legal and technical perspective. The comparison is developed in relation to aspects such as legal validity, interpretation, and lifecycle of contracts. While computer-executable contracts are not a new matter, the project aims to reappraise the discourse in the context of distributed ledger technology. Our focus is on a particular distributed ledger technology, namely blockchain-based systems operated by a trusted collective.

Acknowledgement: This extended abstract is extracted from [1].

References

[1] G. Governatori, F. Idelberger, Z. Milosevic, R. Riveret, G. Sartor, and X. Xu. On legal contracts, imperative and declarative smart contracts, and blockchain systems. *Artificial Intelligence and Law*, 2018.

Decentralised Random Number Generation

Peter Robinson PegaSys, ConsenSys and University of Queensland
peter.robinson@consensys.net peter.robinson@uqconnect.edu.au

***Abstract*—Decentralised random number generation algorithms suffer from the Last Actor Problem, in which the last participant to reveal their share can manipulate the generated random value by withholding their share. This paper proposes an encrypted share threshold scheme which prevents this attack.**

I. Introduction

Historically, decentralised random number generation algorithms have used a commit-reveal process in which participants submit a commitment to their randomly generated share. Once all participants have submitted commitments, each participant reveals their share. The shares are combined using a deterministic algorithm to produce the generated random value. The last participant to reveal their share, known as the Last Actor, can view all of the other shares. They can determine the impact of revealing their share, and thus decide to reveal their share or withhold their share, thus influencing the generated random value.

DFINITY defined a Decentralised Random Beacon [1] using a threshold scheme and BLS Signatures to prevent the Last Actor Problem. Inspired by their work, the scheme described in this paper uses the Shamir Threshold Scheme [2] combined with modulo addition to provide a decentralised random number generation scheme. This scheme is similar in some aspects to the scheme recently proposed by Drake [3].

II. Background

In Shamir's Threshold Scheme random coefficients are generated for an equation of the form shown below, where the value a_0 is the secret value.

$$Y(x) = a_0 + a_1.X + a_2.X^2 + ... + a_{m-1}.X^{m-1} mod P$$

n shares are generated. Any m shares can be used to calculate the y value for any x value. As such, the a_0 secret value can be constructed from any m shares.

III. Algorithm

Set-up: Deploy a smart contract to the blockchain to manage the random number generation process.

Registration: The x value for each participant is the participant's Ethereum address $modP$. Each participant generates an ephemeral RSA or ECC encryption key pair. To register, they publish their public key to the contract. The act of publishing their public key publicises their Ethereum address and hence their x value.

Calculate Random Coefficients: All participants generate $m-1$ random coefficients for an equation in the range 1 to $P-1$. They calculate the y values for each of the participant x values.

Post Commitment: All participants post to the contract the message digest of the y values for each of the participant x values. Any participant which does not post commitment values drops out of the random number generation process and is fined.

Post Encrypted Y values All participants post to the contract the encrypted y values for each of the participant x values, encrypted against the public keys of each other participant. Any participant which does not post all of the encrypted y values drops out of the random number generation process and is fined.

Post Private Keys and Calculate Random All participants post their private decryption keys. The contract then has enough information to calculate the random value and check for correctness. Correctness can be checked for by decrypting the encrypted y values, checking commitments, and checking that the order of the curve that each entity posted is $m-1$. The random value is calculated as the sum of the a_0 values $modP$.

To save gas, all participants post the plain text values for all of the encrypted y values. All participant can off-chain check the decryptions, commitments, and order of equations. If an incorrect value is detected, this could be indicated by a call to the contract, with the contract verifying the bad value and fining the participant.

IV. Properties

Using commitments and asymmetrically encrypting the y values means that individual attackers have to commit to a single value and can not control the release of the information. Each participant holds their own private key and publishes it once all of the encrypted y values are posted, thus releasing the information for all parties to see.

V. Attacks

If m attackers collude they can decrypt the encrypted y values as they are posted. The m attackers could wait for the other $n-m$ sets of encrypted y values to be posted, and then choose one or more attackers to withhold their private key, thus affecting the generated random value. The random generation process could be stopped by $n-m+1$ attackers not publishing their private keys. Doing this would mean that at least m sets of y values can not decrypted, and hence the a_0 values can not be interpolated. Both of these attacks can be countered by fining participants who do not obey the algorithm.

References

[1] T. Hanke, M. Movahedi, and D. Williams. (2018) Dfinity technology overview seriers: Consensus system rev.1. [Online]. Available: https://dfinity.org/pdf-viewer/pdfs/viewer?file=../library/dfinity-consensus.pdf

[2] A. Shamir, "How to share a secret," *Commun. ACM*, vol. 22, no. 11, pp. 612–613, 1979. [Online]. Available: https://cs.jhu.edu/~sdoshi/crypto/papers/shamirturing.pdf

[3] J. Drake. (2018) Leaderless k-of-n random beacon. [Online]. Available: https://ethresear.ch/t/leaderless-k-of-n-random-beacon/2046

The Power and Possibilities of Blockchain for the Enterprise

Niki Ariyasinghe,

Head of Partnerships Asia-Pacific, R3

The purpose of this presentation is to provide insight in to:

- Key aspects of Corda, the only open-source blockchain platform designed from the ground up for enterprise customers
- Deployment of blockchain in real enterprise use cases

Corda is an open-source blockchain platform for recording and processing business agreements. It has evolved and developed from R3, a growing alliance of over 200 of the world's leading firms committed to applying blockchain technology to unlock new value across multiple industries. This includes large technology companies, small startups, individual developers, central banks, regulators, financial services firms and trade associations all engaged in the experimentation and refinement of blockchain technology.

The Corda platform supports smart contracts. Our smart contract is an agreement whose execution is both automatable by computer code working with human input and control, and whose rights and obligations, as expressed in legal prose, are legally enforceable. The smart contract links business logic and business data to associated legal prose in order to ensure that the financial agreements on the platform are rooted firmly in law and can be enforced and that we have a clear path to follow in the event of ambiguity, uncertainty or dispute.

Corda is specialized for use with enterprises. It is heavily inspired by blockchain systems, but without the design choices that make traditional blockchains inappropriate for many business scenarios.

Corda provides a framework to run smart contracts with these key activities and features:

- Recording and managing the evolution of financial agreements and other shared data between two or more identifiable parties in a way that is grounded in existing legal constructs and compatible with existing and emerging regulation
- Choreographing workflow between firms without a central controller
- Supporting consensus between firms at the level of individual deals, not a global system.
- Supporting the inclusion of regulatory and supervisory observer nodes
- Validating transactions solely between parties to the transaction.
- Supporting a variety of consensus mechanisms
- Recording explicit links between human-language legal prose documents and smart contract code.
- Using industry-standard tools
- Restricting access to the data within an agreement to only those explicitly entitled or logically privileged to it.

These features contribute to the design of a platform appropriate for use in complex, enterprise-scale organizations. Note that this design does not use a native cryptocurrency or impose a global transaction speed limit.

The members of the R3 alliance have undertaken over 100 projects over the past 3 years to experiment with blockchain technology, develop Corda and determine its utility for enterprise use cases in areas such as: Digital Cash, Digital Identity, Trade Finance, Digital Assets and Insurance. Some of these areas have rapidly matured into structured roadmaps for implementation; the first live deployments of Corda commenced in April 2018. In addition, partners in the R3 ecosystem are developing Corda applications ("CorDapps") for a range of industries including: Supply Chain, Oil & Gas and Healthcare.

The future of blockchain for the enterprise is bright and is accelerating rapidly as we progress towards 2019.

On the Interoperability of Distributed Ledgers

Dileban Karunamoorthy, Ziyuan Wang, Hoang Tam Vo, John Wagner, and Ermyas Abebe

IBM Research

***Abstract*—The emerging data and value silos in decentralized networks poses a number of challenges for interoperability. While naive approachs are straight forward, designing protocols for exchanging data and value while preserving key properties of decentralized networks introduces additional complexities. The design of a set of primitives for interoperability and clear understanding of its trade offs enables us to construct complex application workflows across different networks. This paper presents a summary of the key drivers and challenges in interoperability.**

***Index Terms*—blockchain, distributed ledgers, distributed systems, interoperability**

I. Introduction

The data and value silos emerging from the growth of decentralized networks presents challenges in interoperability. While the integrity of individual networks are both easier to analyze and reason about within the confines of a running network and its underlying protocol design, exchanging data or value across networks introduces additional complexities. Silos however, are a natural outcome of groups of individuals or organizations aligning along common goals. In the context of decentralized networks, some of these goals include: using distributed ledgers to create decentralized alternatives to solutions that address problems such as money, supply chain or trade finance, much of which is early experimentation and based on network protocols with different properties; forks and parallel implementations as a consequence of market competition or failings in network governance resulting from disagreements within communities on critical decisions; an approach to scalability that enables partitioning ledgers based on use, and isolation of network traffic; permissioned networks limiting access to a known set of identities for confidentiality; and networks designed to operate and comply with regulatory constraints.

II. Challenges

The original specification of Bitcoin [1] laid the foundation for a decentralized protocol. Since then a number of new protocol implementations have emerged with a similar vision offering different characteristics. While a precise definition of interoperability is deferred to an extended version of this paper, the design of an interoperability protocol must strive to preserve properties acceptable to, depending on context, one or both of the interoperating networks. Designing a set of primitives for the exchange of data and value between networks while preserving decentralization enables us to solve more complex real-world interoperability scenarios.

The primitives for interoperability can be based on a number of mechanisms, some of which include: messages and accompanying proofs that can be shared and verified within contracts in different networks; cryptographic protocols that allow for the conditional transfer of value between separate networks [2]; invocation between contracts within a single network where both contracts have similar assurances from the underlying protocol; frameworks specifically designed to enforce global invariants across different sub-ledgers [3] [4]; and methods to reason about or expose notions of trust and integrity to applications that drive cross-network communication.

Complex workflows can be built upon networks that support one or more of these primitives, ranging from: digital asset exchanges that enable tokens of value in private networks (such as a private equity secondary market) to be traded with asset-backed tokens in public networks; to supply chain networks that interoperate with trade or supply chain financing, insurance and dispute resolution networks. The construction of these workflows raises a number of problems and challenges, including: the discovery of networks and contracts and methods for reasoning about trust and logic; the discovery of classes or specific instances of assets and data; the addressability of assets and data along with their histories and dependencies; the preservation of privacy and confidentiality when exchanging messages between networks, preventing leakage; different definitions and requirements for identity across networks; complying with regulations and laws across different jurisdictions; and the stability of interoperability protocols and standards that rely on the independent governance of the interoperating networks.

III. Conclusion

Interoperability is important in order to address data and value silos emerging in decentralized networks. This paper presented a summary of some of the key challenges in interoperability.

References

[1] S. Nakamoto, “Bitcoin: A Peer-to-Peer Electronic Cash” System. 2009. [Online]. Available: http://www.bitcoin.org/ bitcoin.pdf

[2] “Hashed-Timelock Agreements (HTLAs)”, interledger. [Online]. Available: https://interledger.org/rfcs/0022-hashed-timelock-agreements/.

[3] “Ethereum Project”. [Online]. Available: https://ethereum.org/

[4] G. Wood, “Polkadot: Vision for a Heterogeneous Multi-Chain Framework”. 2016. [Online]. Available: https://github.com/polkadot-io/polkadotpaper/raw/master/PolkaDotPaper.pdf

[5] J. Kwon and E. Buchman, “Cosmos: A Network of Distributed Ledgers”. 2018. [Online]. Available: https://cosmos.network/cosmos-whitepaper.pdf

Blockchain for Transparent Food Supply Chains

1st Sidra Malik
The School of CSE, UNSW
Sydney, Australia
sidra.malik@unsw.edu.au

2nd Salil S. Kanhere
The School of CSE, UNSW
Sydney, Australia
salil.kanhere@unsw.edu.au

3rd Raja Jurdak
Data61, CSIRO
Brisbane, Australia
raja.jurdak@csiro.au

***Abstract*—Ensuring food safety and quality are essential to protecting the health and safety of consumers. This is particularly important as several food items are being shipped across the globe which involves the physical movement of goods through complex Food Supply Chains (FSCs). Due to incidents of mislabeled [6], fraudulent and infected [1] food items on the shelf; consumers are taking unprecedented interest in the way food is produced, processed and handled, and are increasingly calling for their governments to accept greater responsibility for food safety. There is a growing interest amongst consumers to know the origin of their food. Unfortunately, finding provenance information is not straight forward due to disparate repositories and complexity of aggregation of data. Moreover, a food item typically moves through a complex supply chain involving many entities with distinct operational practices and procedures.**

The process of collating data from various disparate repositories is a big challenge. To form a *product story* it is necessary to collect data from these repositories and also to ensure integrity of this data. Existing traceability systems based on centralized repositories, but organizational siloing makes the process of tracing provenance information from these distinct sources tedious and fraught with delays.Promoting provenance and quality of food items can increase consumer trust and allow the farmers and producers reach niche profits if the information can be trusted. *Blockchain (BC)* technology can be a solution for providing data integrity and a platform for sharing data across all FSC entities due to its salient features which include decentralization, security and privacy.

BC was first introduced in cryptocurrency known as Bitcoin [5] and later with increasing impact in major industry sectors as finance, health, IoT, supply chains etc. In the context of FSC, information such as the origin of raw materials, ownership details and sensor information, etc could be recorded in the BC as transactions. Additionally, a record of the physical handover of items along the chain could also be stored in the BC, thus simplifying the process of establishing provenance. The use of BC can thus revolutionize food provenance and traceability for not only FSC participants but also for ordinary consumers. For example, in the event of an sickness outbreak, it would allow us to track the following with minimal complexity and delays: 1) the FSC participants involved and 2) the stage at which the contamination occurred i.e. farm, storage, production, logistics or retailer.

In our proposed blockchain solution, we have opted to use a permissioned BC, in which only authorized supply chain participants are allowed to participate for the following reasons: to protect information from FSC stakeholders and potential competitors, rather than using proof of work (PoW), less resource demanding consensus protocols such as either voting or lottery based methods [2], [8] are appropriate to use for permissioned BCs, permissioned BC can predefine access rights of reading and writing to the BC. Another important design consideration is of scalability. BC architecture must scale to handle the total number of transactions a, i.e., achieve high throughput [7] and there should be accessibility for consumers to check provenance without compromising FSC participants' privacy. Our work proposes a conceptual framework which is comprised of FSC entities collaboratively managing a permissioned BC. We emphasize on the need to formulate FSC specific transaction vocabulary based on role of FSC entity, access rights and IoT sensor readings and information relevant to food safety standards. Our solution achieves scalability by dividing the validation task among a set of validators based on the geographic region that they belong to. The key features of our proposed work are:

1) **A consortium framework which provides a platform for FSC entities and administrative bodies to have digital collaboration.**
2) **A transaction vocabulary for storing different types of information and interactions that encompass all FSC processes.**
3) **We propose to improve scalability by using separate chains, i.e. a set of parallel BCs (known as shards [3], [4]) instead on one large BC.**
4) **Read and write access to the BC architecture is managed through *Access Control List*, *(ACL)* which is based on particular roles in the FSC and collectively managed by consortium members.**

To investigate the performance of our proposed system, we develop an implementation of the framework. We measure the performance of query time for retrieving provenance information and verification time for the transactions. We also perform a qualitative security and privacy analysis of our architecture and measure the resistance of our system to known attacks in BC domain.

References

[1] Multistate outbreak of salmonella urbana infections linked to imported maradol papayas, 2017.
[2] JP Buntinx. What is proof of elapsed time?, 2017.
[3] Adem Efe Gencer, Robbert van Renesse, and Emin Gün Sirer. Short paper: Service-oriented sharding for blockchains. In Aggelos Kiayias, editor, *Financial Cryptography and Data Security*, pages 393–401, Cham, 2017. Springer International Publishing.
[4] Wenting Li, Alessandro Sforzin, Sergey Fedorov, and Ghassan O. Karame. Towards scalable and private industrial blockchains. In *Proceedings of the ACM Workshop on Blockchain, Cryptocurrencies and Contracts*, BCC '17, pages 9–14, New York, NY, USA, 2017. ACM.
[5] Satoshi Nakamoto. Bitcoin: A peer-to-peer electronic cash system. 2008.
[6] Jagadeesan Premanandh. Horse meat scandal–a wake-up call for regulatory authorities. *Food control*, 34(2):568–569, 2013.
[7] M Staples, S Chen, S Falamaki, A Ponomarev, P Rimba, AB Tran, I Weber, X Xu, and J Zhu. Risks and opportunities for systems using blockchain and smart contracts. data61, 2017.
[8] Marko Vukolić. The quest for scalable blockchain fabric: Proof-of-work vs. bft replication. In Jan Camenisch and Doğan Kesdoğan, editors, *Open Problems in Network Security*, pages 112–125, Cham, 2016. Springer International Publishing.

Validating Smart Contract Execution Across a Heterogeneous Collection: A Proposal

Padmanabhan Krishnan
Oracle Labs
Brisbane
Email: paddy.krishnan@oracle.com

Babu Pillai
School of ICT
Griffith University, Gold Coast
Email: babu.pillai@griffithuni.edu.au

Kamanashis Biswas
School of ICT
Griffith University, Gold Coast
Email: k.biswas@griffith.edu.au

***Index Terms*—smart contracts, proof-carrying code and data**

I. Motivation

The use of smart contract has generated lots of excitement across many industries. The benefits of this approach come in the form of security offered by distributed consensus. However, most systems are restricted to a single blockchain. So interoperability of different blockchains is an open challenge [1], [2]. In future, there will be many networks of blockchains across different enterprises each of which are application specific. These networks need to communicate and transfer data each other in order to come to an agreement on a global state. However, if the data on the cross-communication channel can be tampered with the security properties may be compromised. Therefore, we need a mechanism where one blockchain system should be able to transfer the data safely and the other blockchain should be able to verify its integrity. Proposed solutions using bridges/connectors that keep track of transactions in each connected chain and manage transfers between them through a mother blockchain [3] have many overheads and is not scalable.

At an application level, communication using transactions and smart contracts must preserve the characteristics of the blockchain based system. This requires that the data received by the different sub-systems is verified against the smart contract specification. Leaving the entire task of verification to the receiver is unlikely to work in practice. As checking a proof is cheaper than generating a proof, we propose that the sender of the data generates the proof and that the receiver only needs to check the proof.

II. Proof Carrying Code

To address the issue of integrity of contract execution, ideas from proof-carrying code (PCC) [4] and proof-carrying data (PCD) [5] may be used. Proof carry data can also be used to validate and regulate data access [6]. The key concepts underlying PCC and PCD is that two parties can share information, in our case smart contract code, such that the receiver can verify the integrity of the received data with little effort. The sender of the data has to generate the proof which is typically a more expensive process. For instance, PCD requires the data to be accompanied by a proof that the message and the history leading to it is valid.

III. Proposed Solution

Scilla [7] is an intermediate language to express and verify smart contracts. But they do not address the question of a malicious user executing a modified program. Our premise is that languages like Scilla can be extended with ideas from proof-carrying code/data so that the receiver can verify that the smart contract has been executed properly. The choice of logic and proof systems [8] will influence the type of applications that can be supported. We need to generalise the work of policy-carrying data [6] which handles only access control policies to handle a wide variety of smart-contract languages. The formal definitions developed for Isabelle/HOL [9] can be used to support the verification of general contracts. However, storing the entire proof in the block may not be practical. So practical challenges like where to keep the proofs need addressing. Techniques used in other systems such as BitHalo and the Raiden network which use off-blockchain and state channel technology, may help address scalability issues.

IV. Conclusion

In this abstract we have outlined a potential solution to the problem of integrity in the execution of smart contracts. Many challenges remain including developing an implementation that can demonstrate the applicability of this approach.

References

[1] S. Underwood, "Blockchain beyond bitcoin," *Commun. ACM*, pp. 15–17, 2016.

[2] T. Hardjono, A. Lipton, and A. Pentland, "Towards a design philosophy for interoperable blockchain systems," arXiv, 2019. [Online]. Available: https://arxiv.org/abs/1805.05934

[3] H. Wang, Y. Cen, and X. Li, "Blockchain router: A cross-chain communication protocol," in *Proceedings of the 6th International Conference on Informatics, Environment, Energy and Applications*, 2017.

[4] G. C. Necula and P. Lee, "Safe, untrusted agents using proof-carrying code," in *Mobile Agents and Security*, ser. LNCS 1419, 1998, pp. 61–91.

[5] S. Chong, E. Tromer, and J. A. Vaughan, "Enforcing language semantics using proof-carrying data," Cryptology ePrint Archive, Report 2013/513, 2013.

[6] J. A. Padget and W. W. Vasconcelos, "Fine-grained access control via policy-carrying data," *ACM Trans. Internet Technol.*, vol. 18, no. 3, pp. 31:1–31:24, 2018.

[7] I. Sergey, A. Kumar, and A. Hobor, "Scilla: A smart contract intermediate-level language," 2018. [Online]. Available: http://ilyasergey.net

[8] M. Herlihy and M. Moir, "Blockchains and the logic of accountability: Keynote address," in *Proceedings of the 31st Annual ACM/IEEE Symposium on Logic in Computer Science*. ACM, 2016, pp. 27–30.

[9] Y. Hirai, "Defining the ethereum virtual machine for interactive theorem provers," in *Financial Cryptography and Data Security (FC)*, ser. LNCS 10323, 2017.

Towards Formal Verification of Solidity Smart Contracts Using PAT

Christopher Skorka, Lee Goymer, Hadrien Bride, Zhé Hóu and Jin Song Dong
Institute for Integrated and Intelligent Systems, Griffith University, Australia
Email: {christopher.skorka,lee.goymer}@griffithuni.edu.au, {h.bride,z.hou,j.dong}@griffith.edu.au

***Abstract*—This paper investigates the feasibility of translating Solidity Smart contracts and their underlying features into the model checker PATs (Process Analysis Toolkit) CSP# (Communicating Sequential Processes #) language and formally verifying that the contracts obey specified properties. We describe how to verify some basic underlying features and a selected set of properties defining the Solidity language and the execution of smart contracts.**

I. Introduction

Decentralized cryptocurrencies have gained significant interest and adoption since the introduction of Bitcoin in 2009. A prominent use of the blockchain are smart contracts; a platform for decentralized code execution. *Ethereum* is a smart contract platform which uses its own digital currency called *Ether* to pay for the execution of these contracts. One issue related to using smart contracts on a public blockchain is that bugs, including security holes, are visible to all users. Further, due to the immutable nature of most blockchains, bugs cannot be patched easily. This has caused the digital currency such as Ether to be volatile under circumstances (e.g. the price of ether dropped from 21.50 to 8 when the Decentralised Autonomous Organisation (DAO) was hacked on 17 June 2016). Therefore, it is crucial to formally analyse the reliability and trustworthiness of smart-contracts before they go live. To do that, we use Process Analysis Toolkit (PAT) [4] – a formal verification tool – to formally model the features of the *Solidity* language – Ethereum smart-contracts' language – and find vulnerabilities in smart contracts.

Some previous research has attempted to formally verify certain properties of smart contracts and the Ethereum Virtual Machine (EVM). Typically these methods convert the syntax of Solidity into a mathematical model such as higher order logic for verification. Examples include Amani's formalisation in the proof assistant Isabelle/HOL [1]. Additionally, in similar research, parts of the syntax of both Solidity and EVM bytecode was translated into the F* programming language [3]. We propose to use PAT to efficiently test a wide range of assertions, while providing human-readable counterexamples.

II. Verification of Solidity Contracts

We focus on modelling and verifying Solidity code against the following vulnerabilities: (1) Timestamp dependence: timestamps can be modified at runtime by EVM to manipulate the outcome of timestamp-reliant function calls and events. (2) Transaction values can be changed last minute (before a new block is added) and added in the same block as another such that the second transaction has unintended effects. (3) Re-entrance: Sending ether or calling other contracts functions can call the original function again which prevents the original instance of the original function from proceeding. (4) Stack overflow. (5) Integer overflow. (6) Exceptions: Attacks can be engineered to cause exceptions at specific lines of code. (7) Delegate calls can be used to execute new unknown code. (8) Sending money from contracts automatically is prone to vulnerabilities, rather, money should be added to a balance map from which users can withdraw their balance manually.

The process of translating solidity contracts into PAT's CSP# language is given below at a very high level: First, all the global variables are translated into their corresponding PAT's variables. Then, the smart-contract constructor (named $Init$) and every exposed Solidity functions are translated into their corresponding CSP processes such that their semantic is preserved. Further, a new process called $Action$ that selects any of the functions available excluding the initial smart-contract constructor is introduced. The full behaviour of the modelled smart contract can then be modelled by the process $Init; Action \rightarrow Action$ (i.e., $Init$ followed by any number of $Action$). Finally, assertions (i.e., properties to be verified) are written as a part of the PAT code. Then the processes can be simulated and assertions can be verified.

As future work, we plan to translate all solidity properties into PAT's CSP# language. We will develop a fully automated verification module to convert Solidity contracts into CSP# models and verify them.

References

[1] Amani, S., Bgel, M., Bortin, M. and Staples, M., 2018. Towards Verifying Ethereum Smart Contract Bytecode in Isabelle/HOL. CPP. ACM. To appear.

[2] Hildenbrandt, E., Saxena, M., Zhu, X., Rodrigues, N., Daian, P., Guth, D. and Rosu, G., 2017. KEVM: A Complete Semantics of the Ethereum Virtual Machine.

[3] Bhargavan, K., Delignat-Lavaud, A., Fournet, C., Gollamudi, A., Gonthier, G., Kobeissi, N., Kulatova, N., Rastogi, A., Sibut-Pinote, T., Swamy, N. and Zanella-Bguelin, S., 2016, October. Formal verification of smart contracts: Short paper. In Proceedings of the 2016 ACM Workshop on Programming Languages and Analysis for Security (pp. 91-96). ACM.

[4] Sun, J., Liu, Y., Dong, J.S. and Pang, J., 2009, June. PAT: Towards flexible verification under fairness. In International Conference on Computer Aided Verification (pp. 709-714). Springer, Berlin, Heidelberg.

[5] Sharma, T. 2018, April. Learn Solidity: Exceptions in Solidity . Retrieved from Blockchain Council.

Blockchain-based Booking System - Design and Model Checking

Lung-Chen Huang*, Naipeng Dong*, Guangdong Bai†, Siau Cheng Khoo*, Jin Song Dong*‡
* National University of Singapore, Singapore
† Singapore Institute of Technology, Singapore
‡ Griffith University, Australia

I. Introduction

The blockchain technology has rapidly emerged in recent years, especially when the concept of smart contract was first introduced to and later relied on the technology. Once the smart contract code is deployed to the blockchain, it can be executed by any computer node in the blockchain network that keeps the same historical record of transactions as other nodes. This makes it difficult to be compromised by a single node on the network, unlike centralized platforms that could be easily breached and prone to the failure of a single point. Thus, blockchain and smart contract together provide a decentralized platform that ensures trust and security. Due to these advantages, various communities and companies have proposed use cases, for example, the logistics and insurance industry.

In this work, we propose to design and analyze a decentralized hotel booking solution based on blockchain. Our work is motivated by a fact that most travelers have hard times finding a hotel room for their next journey. They possibly browse through pages of entries on online travel agencies (OTAs) such as Booking.com or Agoda.com. As what can be observed, it is time-consuming that each time a search result appears on the screen, they have to delve into the deals one by one, which could be overlapping in the last search results. In order to alleviate this monotonous experience, we leverage the blockchain technology. On one hand, it allows search requests to be deployed as smart contracts such that the process of discovery is left to the decentralized system, where hoteliers can easily match their rooms with the criteria required by a traveler; on the other hand, the blockchain technology ensures that the contracts between the travelers and hoteliers will be executed as agreed.

II. System design, modelling and verification

The system aims to provide users with an interface where they can draft their booking request with the requirement of a hotel room in a domain-specific language, which is similar to a real contract in a human-readable form. Thereafter, the interface compiles the request into a machine-readable form and injects some predefined functions. Later, the user can deploy the request onto the blockchain. Once the request is visible to other nodes, hoteliers can propose offers by invoking a function in the request. When the user is notified of new proposals, the interface automatically starts a selection process of the proposals depending on the request criteria from the user, showing the matched results. At the end, the user and one of the hoteliers seal a deal.

In order to ensure the system design satisfies the desired properties and ensure that there is no ambiguity when implementing the system, we use CSP# to formally model the system design, including *user*, *miner* and *request*.

- A *user* is a traveler or hotelier who sends a transaction by invoking a function that includes the address of the user, function name, gas, and gas price. A traveler can invoke **Fetch** to acquire the latest set of proposals submitted by hoteliers, or to **Settle** for a specific proposal. A hotelier can invoke **Propose** to send their proposals to the request.
- Each *miner* has its own transaction pool **TxPool** that continuously receives a new transaction with **gas** greater than zero. Moreover, **TxBlock** to be executed by a miner also finds the new transactions with more than or equal to a specified transaction **gasPrice** from the pool.
- A *request* deployed by a traveler to the blockchain is a smart contract that accepts proposals and can be invoked by the users.

Then we use PAT (Process Analysis Toolkit) to verify whether the following set of properties are satisfied.

- No deadlock situation occurs in the system.
- Each transaction in a block has enough gas to be executed to completion by miners.
- Each miner reaches the same block eventually.
- The request owner has settled with some proposer, and thus each miner receives the same transaction.
- Proposals have been accepted by the request.

As the preliminary result, we check these properties with settings of five users and two miners in the model.

III. Discussion and future work

When increasing the number of miners to be larger than ten, the model checker can only terminate for verifying one of the properties. This shows that the main challenge for verifying a blockchain-based decentralized system is the blockchain network itself. Therefore, a formal modeled and verified blockchain platform is necessary so that the blockchain part can be abstracted when verifying blockchain-based systems. In addition, our model has not take network attackers and malicious nodes into account, but they are important for ensuring security of the entire system.

Blockchain Platforms for IoT Use-cases

Mohammad Chowdhury, Md. Sadek Ferdous, Kamanashis Biswas
mjchowdhury@swin.edu.au, s.ferdous@imperial.ac.uk, k.biswas@griffith.edu.au

IoT & Blockchain: The Internet of Things (IoT) is experiencing an exponential growth in a wide variety of use cases, such as wearable devices, agriculture, smart cities, smart homes, supply chain and so on. IoT technology is fundamentally different, mainly due to its decentralized topology and the resource-constraints devices. Thus, IoT systems often rely on centralized computing and storage system (e.g., cloud infrastructure) for processing distributed data. This computation model usually suffers from privacy and security vulnerabilities. In addition, such devices depend on a heterogeneous underlying network infrastructure which is easy to attack as evident in several recent cyber attacks. Recently, blockchain technology has gained popularity in different domains because of its multiple properties such as resiliency, support for integrity, anonymity, decentralisation and autonomous control. Thus, blockchain technology can be an effective mechanism to address the issues involving IoT. Hence, there has been enthusiasms to combine blockchain technology with IoT.

Existing IoT-focused blockchain platforms: Towards this aim, several blockchain solutions for IoT environments have been proposed: IOTA [1], Waltonchain [2] and OriginTrail [3]. IOTA uses a special consensus algorithm, called Tangle, which uses Direct Acyclic Graph (DAG) and is much lighter than conventional consensus algorithms such as Proof-of-Work and Proof-of-Stack. Waltonchain combines blockchain with IoT (specifically RFID) to create a management system for supply chains. It uses their own Proof of Stake & Trust (PoST) consensus along with a node reputation mechanism. Finally, OriginTrail is used in the supply chain domain where different IoT devices are expected to create a network to track the state of a product within the supply chain network. To enable this, OriginTrail utilizes a layer-based approach where a blockchain platform functions in the bottom layer with the option to attach any blockchain platform as required by the application.

Analysing & comparing IoT use case requirements: Different IoT use-cases need different requirements. For example, requirements in smart cities are different to that of the wearable fitness tracking or goods tracking system in supply chain management. Even so, we can identify several core requirements prevailing in all use-cases:

- Transaction speed & cost
- Scalability
- Data security & privacy
- Trust establishment
- Virtual network among partners

Among these, transaction speed and cost, and scalability will mainly determine if a particular blockchain platform can handle the amount of data generated by multitude of IoT devices as well as if it feasible in terms of the associated cost. Data security and privacy will need to consider the confidentiality, integrity, access control and ownership of data and IoT devices in the network. Trust establishment will consider how a platform can establish trust. Finally, the virtual network among partners will consider the scenarios when different partners need to share their data generated from their corresponding IoT devices just within themselves. Table 1 shows a brief analysis of the derived general requirements and the coverage of the existing blockchain solutions. It is evident from the table that the existing platforms do not address all our identified requirements.

Future work: In future, a detailed analysis of each of the requirement in the table will be done. The proposed concept level requirements and comparison of blockchain platforms will lay the foundation for understanding and developing blockchain platforms.

Table 1: Comparing IoT requirements and Blockchain platforms

Requirements	IOTA	Waltonchain	OriginTrail
Transaction speed and cost	500-800 transactions per second. 0 Fees.	4 transactions per second. Uses side chain to speed up.	Depends on IOTA, Ethereum, or NEO for consensus.
Data Security & Privacy	Support data security but not privacy of data	Support data security but not privacy of data	ZKP [4] to provide privacy of the transacted data.
Trust Establishment	IDoT [1] is used to build reputation systems	Use node reputation mechanism	Each stakeholder has to be approved by the previous node
Virtual network	Does not support. However, plan is in the pipeline.	Does not support private communication	Does not support private communication

Reference:

[1]. IOTA White Paper, https://iota.org/IOTA_Whitepaper.pdf
[2]. Waltonchain, White Paper, https://www.waltonchain.org/doc/Waltonchain-whitepaper_en_20180208.pdf
[3]. OriginalTrail White Paper, https://origintrail.io/storage/documents/OriginTrail-White-Paper.pdf
[4]. Feige, U., Fiat, A. and Shamir, A., 1988.Zero-knowledge proofs of identity.Journal of cryptology, 1(2), pp.77-94.

Distributed Business Process Flexibility on Blockchain

Guido Governatori
DATA61
CSIRO
Brisbane, Australia
guido.governatori@data61.csiro.au

Nick van Beest
DATA61
CSIRO
Brisbane, Australia
nick.vanbeest@data61.csiro.au

Regis Riveret
DATA61
CSIRO
Brisbane, Australia
regis.riveret@data61.csiro.au

Silvano Colombo Tosatto
DATA61
CSIRO
Brisbane, Australia
silvano.colombo@data61.csiro.au

***Abstract*—Organisations commonly outsource parts of their business processes to third parties. Through the use of blockchain technology and smart contracts, we aim at providing an untrusted framework that allows seamless interaction between business processes from different organisations.**

***Index Terms*—BPM, DLT, Modular Deployment**

I. Executing Business Processes on a Blockchain

Business process models are abstract models capable of compactly describing the different ways that a business organisation can achieve a given business objective. The capability of deploying a business process model on a distributed environment, such as a blockchain-based system, allows to easily share the services provided with other organisations. Additionally, due to the tamper-evident nature of blockchain technology, an organisation deploying a process can be sure that the deployed business process cannot be maliciously altered without being visible to all participants.

A substantial amount of work has already been done in this area to move business processes to blockchains for exactly those reasons (consider e.g. [1]–[3]). A notable example of such an implementation is Caterpillar, proposed by López-Pintado et al. [3], which allows to transform business processes modelled in BPMN to an executable smart contract.

II. Process Deployment Monitor

When organisations execute a business process, certain parts of their process may be delegated to external parties. This so-called process outsourcing is commonly bound by a contract between the involved parties specifying the service requested, the agreed service levels, etc.

However, current approaches for deploying processes on blockchains do not consider outsourcing as an essential part of modern business practice and do, as such, not support delegation of multiple contracts provided by different parties. To achieve outsourcing as illustrated in Fig. 1, we introduce a *process deployment monitor*. This component handles the deployment of the smart contracts representing the processes on a blockchain. It keeps track of the deployed processes' properties, allowing other organisations to identify which third party processes can be integrated in their own processes.

The deployed processes are identified by a logical identifier, while the monitor also keeps track of the physical references of the deployed processes. This allows organisations to refer to third party processes using a logical identifier, while the monitor translates it in order to remotely call the execution of another one.

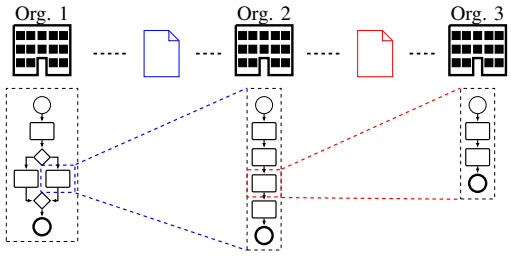

Fig. 1. Example of business process outsourcing.

III. Advantages

- The deployment monitor takes care of linking the physical addresses of the deployed processes with their logical identifiers. This allows organisations to seamlessly include third party procedures via logical reference.
- The deployment monitor exposes the process contracts and their properties. This allows an organisation to choose the best solution for their process.
- The deployment monitor allows organisations to update their deployed processes. When this does not alter the process contracts, the monitor simply switches the physical address, making the update seamless for each of the updated process users.

References

[1] I. Weber, X. Xu, R. Riveret, G. Governatori, A. Ponomarev, and J. Mendling, "Untrusted business process monitoring and execution using blockchain," in *Proc. of the 14th Int. Conf. on Business Process Management*. Springer, 2016, pp. 329–347.

[2] L. García-Bañuelos, A. Ponomarev, M. Dumas, and I. Weber, "Optimized execution of business processes on blockchain," in *Proc. of the 15th Int. Conf. on Business Process Management*. Springer, 2017, pp. 130–146.

[3] O. López-Pintado, L. García-Bañuelos, M. Dumas, and I. Weber, "Caterpillar: A blockchain-based business process management system," in *Proc. of the BPM Demo Track and BPM Dissertation Award co-located with 15th Int. Conf. on Business Process Management*, 2017.

The case for DLT in Healthcare – real or hype?

Kris Vette
Vette Solutions Ltd
kris@vettesolutions.com

Abstract

Andreas Antonopoulos has stated that he fails to see a use case for blockchains in healthcare. This paper looks at that contention from the perspective of the challenges currently existing but not yet met in healthcare systems worldwide.

1. Introduction

Healthcare worldwide has advanced at significant pace over the last fifty years yet much of that progress fails to materialize at the clinical coalface. Can Distributed Ledger Technology (or blockchain) solve this problem and if so what would that protocol look like? I propose a counterview to Andreas Antonopoulos' view that there is no use case for blockchain in healthcare.

2. Current State of Healthcare

There are two very different sides to the Healthcare coin today. On the one hand we are seeing rapid technological advancement in medical research such as genome sequencing, precision medicine, stem cell therapies, biologics and clot retrieval. On the other hand healthcare often fails to deliver basic care to the patient.

Part of that problem are the 'rails' that the data runs on. For example a cancer patient's GP will not be able to access their hospital record of current care. Neither will a patient's records be viewable if they move from one hospital to another or from one GP to another or even to the patient themselves. While various commercial forces are at play, there is no reason why, in 2018, patient records and traceable best practice should be inaccessible to vital players. This is a significant real world problem.

3. The Promise of Information Technology

It is true that centralized databases are undoubtably the best technology for performance speed, data storage and scalability Yet for all their promise the latest digital technology has failed to deliver interoperability and real-time access to clinical practice. Apart from other failings once a patient transgresses into a separate health system virtually all data interoperability is lost. And with that comes the threat of the system-induced error that technology is supposed to mitigate.

4. Distributed Ledger Technology enters the Saloon

Bitcoin combined a number of nascent technologies to produce the first use-case for modern DLT. However many commentators have stated that there are limited commercial applications beyond crypto-currency. While there are many blockchain Start-Up's emerging in the healthcare space, Antonopoulos' negative view of the utility for blockchain in this field cannot be dismissed. He has stated that many potential blockchain use-cases fail to understand that blockchain is a slow, decentralised database.

5. Can DLT match up?

Ultimately DLT or a blockchain solution must hold a characteristic different to high speed centralized databases if it is to solve the real problem described for healthcare.

That problem is the lack of a ubiquitous, real-time shared patient data picture. Healthcare is an eco-system of multiple players and everyone needs an accurate current view, along with patient controlled record portability.

This paper explores the characteristics, including interoperability, of both public and permissioned blockchains and makes the case that both protocols hold properties that will disintermediate control of patient data. This will provide ubiquitous but controlled access to those records and best practice.

Additionally, different potential (public and permissioned) protocol configurations can be modified to incentivise different parts of the system to produce more transparent outcomes.

Vacci-Chain: The Smart Contract Powered Vaccine Storage and Monitoring Solution

Kamanashis Biswas, Thomas Csere, Wee Lum Tan, Vallipuram Muthukkumarasamy
{k.biswas, w.tan, v.muthu}@griffith.edu.au, thomas.csere@griffithuni.edu.au
Institute for Integrated and Intelligent Systems, Griffith University, Australia

Abstract– Vaccine is a delicate biological substance which must be kept within 2°C – 8°C temperature during storage and transportation to preserve its effectiveness. In practice, refrigerators are used to store vaccines, while thermometers and data loggers are used to record and monitor temperatures and trigger an alert to any deviation from the recommended range. This research proposes a blockchain-based solution which uses a smart contract to provide an enhanced level of safety, transparency and traceability.

I. INTRODUCTION

Vaccines are mainly stored in purpose-built, domestic and portable refrigerators that should maintain the recommended temperature to keep them effective and safe. However, the safety aspects of vaccine storage have not advanced at the speed at which current computing has allowed. Current systems usually do not include battery backup systems and require manual readings/logging of temperatures, which can be subject to human tampering/error [1-3]. Further, temperature fluctuations of domestic refrigerators can be caused by the defrosting cycle in frost-free refrigerators, as well as the door opening. Every deviation from the recommended temperature range must be recorded and reported immediately to ensure the safety of vaccines.

II. VACCI-CHAIN IMPLEMENTATION

Vacci-Chain overcomes most hurdles faced by currently implemented vaccine monitoring systems. Vacci-Chain, by using smart contracts, is able to carry out all monitoring of temperatures autonomously in an immutable and reliable fashion. The proposed system stores all information about the received vaccine on an immutable blockchain. It uses a number of sensors to collect real-time temperatures and send the readings to an aggregator. The aggregator further sends the information to all participants in the system in the form of a transaction as shown in Fig. 1. If the recorded temperature is outside of the recommended range then the smart contract is executed. As a result, all vaccines in the given refrigerator are marked as invalid and are no longer able to be used within the system.

This functionality is easily implemented into a smart contract written in *Solidity* on the *Ethereum* platform. The use of multiple, redundant monitoring devices inside a refrigerator is within the feasible limit of the Vacci-Chain functionality. This means that faulty monitoring equipment resulting in erroneous reporting is far less likely and can be handled very effectively, for e.g., requesting component servicing before critical failure. The following figure shows the "tempReceive" function used to check whether the received temperature is within the recommended range.

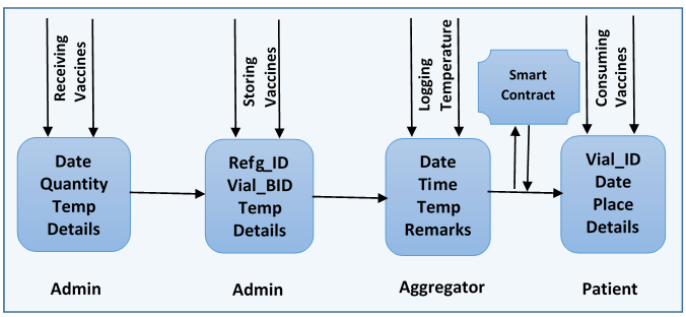

Fig. 1: Vacci-Chain entities and their relationship

```
121     //----temp----\\
122     function tempRecieve(int _temp, uint _fridgeID) {
123         temp = _temp;
124         tempUpdateDate = now;
125         if (temp >= tempRange.max || temp <= tempRange.min) {
126             for (uint i = 0; i < Vials.length; i++) {
127                 if (Vials[i].fridgeID == _fridgeID) {
128                     Vials[i].fault = true;
129                 }
130             }
131         }
132     }
```

Fig. 2: The 'tempReceive' function

The proposed smart contract based solution ensures that every violation in the recorded temperature will be notified immediately and the corresponding vaccines will be marked as unusable. Thus the proposed Vacci-Chain system enables a number of unique benefits such as transparency, traceability, safety, and trust. Anyone can trace back the recorded information such as the temperature history, manufacturer and delivery information simply by entering the vial identification number on a mobile application or website.

III. CONCLUSION

Vacci-Chain, a blockchain powered system, overcomes many of the pitfalls of current solutions to vaccine storage. With the use of blockchain's immutable nature and smart contracts this system is able to monitor and report temperature fluctuations with security in mind at every step of the supply chain. The future work aims to develop an appropriate user interface to provide an easy access to the system for all participants.

ACKNOWLEDGMENT

This research is partially funded by the Institute for Integrated and Intelligent Systems, Griffith University.

REFERENCES

[1] Biswas, Kamanashis & Muthukkumarasamy, Vallipuram & Tan, Wee Lum. (2017). Vacci-Chain: A Safe and Smarter Vaccine Storage and Monitoring System.

[2] Australian Government Department of Health and Ageing, NationalVaccine Storage Standards, Strive for 5, 2nd Edition, 2013

[3] IBM Institute of Business Value, Healthcare rallies for blockchains: Keeping patients at the centre, https://www-01.ibm.com/common /ssi/cgi-bin/ssialias?htmlfid=GBE03790USEN, 2016.

Notes