# Technical Perspective: Revealing Every Story of Data in Blockchain Systems

Yaron Kanza
AT&T Labs-Research
kanza@research.att.com

For many applications, data are worthy only if they are trustworthy. The concept of trust is sometimes elusive, and yet it is fundamental in data management. Even when not expressed explicitly, the correctness of computations and reliability of applications depend on trustworthy management of the data. These notions received new attention with the advent of blockchain and distributed ledger technology.

Blockchain was originally introduced as a decentralized ledger of cryptocurrency transactions, in order to solve the "double-spending" problem [2]. Cryptocurrency coins that are given to a user should not be spent more than once. This is crucial for establishing trust in the currency and guaranteeing that the total number of coins will be limited. To prevent double spending, blockchain is tamper-proof and transparent—it is very hard computationally to change stored transactions (practically impossible).

The ability to create a trusted ledger in a decentralized environment, by consensus, attracted the attention of practitioners, theoreticians, organizations and application developers. A large variety of blockchain technologies and blockchain-based applications were developed [3]. But while blockchain technologies have many advantages, they still lack many capabilities that exist in database management systems, e.g., query language, views, data provenance, etc.

The database community has extensively studied data provenance (also known as data lineage) as a concept and a set of tools that are aimed to make data history more transparent [1]. Being able to examine the "story" of a data instance, starting with the data sources and through the operations that were applied to the data, has been promoted as a way to increase credibility and the user's understanding of the data in complex databases.

The paper of Ruan et al. presents a powerful combination of blockchain and provenance. It lays foundations for building a bridge between database systems and blockchains by showing that the marriage of blockchain technologies and database concepts like provenance can yield a better solution for transparent data management, while tracing historical changes in the data.

Originally, blockchains were not designed for tracking historical data. Once an amount of cryptocoins has been spent, it can no longer be a part of a valid payment, so its does not need to be accessed. Furthermore, dependencies between operations are not recorded, e.g., when a value is read from the blockchain, modified and written back to the blockchain, the dependency between the stored values is not recorded. The challenge the authors had to solve was how to provide provenance information in a way that is both trustworthy and efficient. This is not an easy task given that blockchain systems often sacrifice efficiency for reliability and security.

To track provenance data, the authors present novel data structures—a novel index based on skip lists and Merkle DAG which is an adaptation of Merkle tree—and their integration to implement efficient and reliable storage and retrieval of provenance data. To suit the blockchain environment, these index structures are required to satisfy the following three properties. (1) The index should provide a verifiable digest of tracked states, without the need to read the entire transaction history. (2) Updates should be incremental and succinct, because the storage and the management of provenance data must be efficient. (3) The index should be tamper-proof, similar to the transaction and state information for which it was built. The paper shows how to achieve these requirements using the proposed index structures.

The paper demonstrates a clever use of smart contracts to achieve the desired goals. A smart contract is essentially code that is triggered when particular events occur, and executed by the peers that manage the blockchain, in a decentralized fashion. Smart contracts are somewhat similar to a combination of triggers and stored procedures in database management systems, but there are also differences between these mechanisms. Papers like this work of Ruan et al. shed light on some of the differences and similarities between triggers and smart contracts, but more papers like this work are needed to further investigate the limits of smart contracts in data management applications.

This paper is of high significance because it presents a new way to examine how historical data on a blockchain can be retrieved and used—rather than just looking at the latest state, the entire history that led to the state can be examined and used. It paves the way for systems that would manage tamper-proof records of data transformations in a decentralized fashion. This work is also important because it gives us a glimpse into how advanced decentralized data management should look like and how we could increase transparency and trustworthiness in data management.

## 1. REFERENCES

[1] J. Cheney, L. Chiticariu, and W.-C. Tan. Provenance in databases: Why, how, and where. *Found. Trends Databases*, 1(4):379–474, 2009.

[2] S. Nakamoto. Bitcoin: A peer-to-peer electronic cash system. Technical report, Manubot, 2008.

[3] S. Underwood. Blockchain beyond bitcoin. *Commun. ACM*, 59(11):15–17, 2016.