Accountable Decentralized Event Reasoning Using Blockchains

Jean-Pierre Münch[1] , **Florian Weinacker**[1] , **Guido Salvaneschi**[1] , **Alessandro Margara**[2]
[1]Technische Universität Darmstadt
[2]Politecnico di Milano
jean-pierre.muench@stud.tu-darmstadt.de, florian.weinacker@stud.tu-darmstadt.de,
salvaneschi@cs.tu-darmstadt.de, alessandro.margara@polimi.it

Abstract

Events are a powerful abstraction that allows developers to model several computing activities, including user interactions, system monitoring, business processes, and coordination among services. In this context, event reasoning can be implemented to combine, filter and correlate events and detect higher-level events of interest. In many scenarios, however, events are generated and exchanged across the boundaries of business organizations that may not necessarily trust each other.

We discuss our ongoing work to ensure that the actors that take part to the event exchange are accountable for the events they generate. We integrate event reasoning with blockchain technology to ensure that event generation is recorded by the blockchain consensus protocol and we investigate optimization strategies to improve efficiency.

1 Background and Motivations

Events have increasingly become a core abstraction to model activities of interest in many diverse domains. Software systems that support complex processes are often implemented as event-based systems, where loosely-coupled software components interact by producing and consuming events (Muhl, Fiege, and Pietzuch 2010). In fact, event-based interaction has been advocated as a suitable communication pattern in modern microservices architectures (Stopford 2018). In this context, individual components implement their functionality by *reasoning* (Giatrakos et al. 2019) on the events they observe from the environment in which they are deployed, and output new events in response. For instance, a shipping service could observe events related to orders and locations of couriers to derive and output events related to the status of shippings.

It is common for event-based systems to cross the boundaries of individual business units or even organizations. In this setting, the actors involved in the communication might not fully trust each other and the events they produce. For instance, a shipping service could erroneously or maliciously ignore an order. Due to the loose coupling promoted by event-based systems, it is hard to trace the sequence of events leading to an unexpected behavior and identify the responsible component or components.

Blockchain technologies like Bitcoin (Nakamoto 2008) and Ethereum (Buterin 2014) allow parties that do not trust each other to reach an agreement on the operations they perform without the need to involve a central trusted entity. However, one of the major limitations to their adoption is their large overhead and poor scalability.

In this talk, we introduce our ongoing research on using blockchains to make untrusted components of event-driven systems traceable and accountable for their derivations and for the events they produce. To mitigate performance limitations, we propose a solution where the event space is partitioned across multiple blockchains. Moreover, developers can decide to consume events that have not been validated through the blockchain to reduce latency at the expense of security. As a long-term vision, we plan to make accountability and trust first-class citizens of event-based architectures, and incorporate them in the formalisms and technologies for reasoning on events.

2 Accountable Event-Based Communication

In this section, we overview the solution we are designing for an accountable event-based middleware. We introduce the system model and the interface it exposes and then we discuss the optimizations we are developing to improve performance and scalability.

2.1 Accountable Event Reasoning

As in traditional event-based architectures, the actors in the system can have two roles: (1) they can *publish* events and (2) they can *subscribe* to events and be notified when an event having a certain type and content occurs. Roles can be mixed to enable *reasoning* about events: actors can analyze various types of events and correlate them to infer higher-level knowledge, encode it in the form of other (derived or composite) events, and propagate them to interested parties (Giatrakos et al. 2019).

In our model, publications of events are recorded into a blockchain, thus making data sources accountable for the events they publish. Recording events in the blockchain occurs via a *smart contract*, which has a `write()` method to declare that an event occurred. The publications of composite events are also recorded in the blockchain, thus making the publisher accountable for the derivation process. In line with recent work that builds on blockchains (El-Hindi et al. 2019), we enable thin clients to access the services of the

event-based infrastructure by connecting to a node of the infrastructure that they trust.

2.2 Performance Optimizations

Distributed ledgers employ consensus algorithms that impose a significant performance overhead on the system. For this reason, we are experimenting a number of optimizations.

First, we allow consumers to specify trust policies when subscribing to events. For instance, a consumer can trust all sources and receive events even before they are stored on the blockchain, which provides low delay but no accountability. Conversely, it can consume events only after they have been recorded on the blockchain, which ensures the source is accountable for the publication at the cost of a higher delay.

Finer-grained configurations are possible: for instance, a consumer might trust the reasoning performed by a certain actor, but only if the reasoning derives from traceable events that have been recorded on a blockchain. Also, a consumer might select different policies for different classes of event types, based on their role in the specific domain.

Second, users can specify a partitioning scheme for events, and events belonging to different partitions can be stored on different blockchains. Different partitions can have different replication factors, thus enabling to trade security for performance. In fact, partitioning is a widely studied technique to improve the scalability of blochain tecnology (Zamani, Movahedi, and Raykova 2018).

3 Related Work

This section discusses related work that investigated the use of blockchain technologies in combination with data management and processing systems.

BlockchainDB (El-Hindi et al. 2019) builds a database layer on top of blockchains, where actors are accountable for the operations they perform. Despite targeting a different type of systems (shared database providing operations to read and update its state), BlockchainDB inspired some of the optimizations we are considering in our work, including partitioning and the possibility to consume data with multiple levels of trust. Related to these optimizations, several sharding protocols have been proposed to improve the scalability of blockchain (Zamani, Movahedi, and Raykova 2018), which could be beneficial in the context of event reasoning we target.

Richard Hull emphasized the similarity between the content of blockchains —lists of changes recorded into an append-only log— and events (Hull 2017). Indeed, the use of blockchains has been investigated in application domains that traditionally exploit event-based communication patterns, such as business process management (Carminati, Rondanini, and Ferrari 2018), supply chain monitoring (Sund et al. 2020), IoT data storage and access (Shafagh et al. 2017).

4 Conclusions

Event-based systems often cross the boundaries of single organizations, thus introducing a problem of trust. We present our ongoing work on adopting blockchain technology to ensure that the actors that are part of the system are accountable for the events they derive and publish. We discuss optimization techniques to mitigate performance and scalability problems, and we propose a flexible interface where consumers can trade security for reduced latency in the delivery of event notifications.

References

Buterin, V. 2014. A next-generation smart contract and decentralized application platform.

Carminati, B.; Rondanini, C.; and Ferrari, E. 2018. Confidential business process execution on blockchain. In *2018 IEEE International Conference on Web Services (ICWS)*, ICWS '18, 58–65. IEEE.

El-Hindi, M.; Heyden, M.; Binnig, C.; Ramamurthy, R.; Arasu, A.; and Kossmann, D. 2019. Blockchaindb - towards a shared database on blockchains. In *Procs. of the Int. Conf. on Management of Data*, SIGMOD '19, 1905–1908. New York, NY, USA: ACM.

Giatrakos, N.; Alevizos, E.; Artikis, A.; Deligiannakis, A.; and Garofalakis, M. 2019. Complex event recognition in the big data era: a survey. *The VLDB Journal*.

Hull, R. 2017. Blockchain: Distributed event-based processing in a data-centric world. In *Procs. of the Int. Conf. on Distributed and Event-Based Systems*, DEBS '17, 2–4. New York, NY, USA: ACM.

Muhl, G.; Fiege, L.; and Pietzuch, P. 2010. *Distributed Event-Based Systems*. Springer, 1st edition.

Nakamoto, S. 2008. Bitcoin: a peer-to-peer electronic cash system.

Shafagh, H.; Burkhalter, L.; Hithnawi, A.; and Duquennoy, S. 2017. Towards blockchain-based auditable storage and sharing of iot data. In *Procs. of the Cloud Computing Security Workshop*, CCSW '17, 45–50. New York, NY, USA: ACM.

Stopford, B. 2018. *Designing Event-Drive Systems: Concepts and Patterns for Streaming Services with Apache Kafka*. O'Reilly Media, 1st edition.

Sund, T.; Loof, C.; Nadjm-Tehrani, S.; and Asplund, M. 2020. Blockchain-based event processing in supply chains - a case study at ikea. *Robotics and Computer-Integrated Manufacturing* 65:1–16.

Zamani, M.; Movahedi, M.; and Raykova, M. 2018. Rapidchain: Scaling blockchain via full sharding. In *Procs. of the Conf. on Computer and Communications Security*, CCS '18, 931–948. New York, NY, USA: ACM.