Synergy of Distributed Ledger Technologies and the Internet of Things*

Sebastian Kanz
Distributed Ledger Technologies
MaibornWolff GmbH
Frankfurt a.M., Germany
sebastian.kanz@maibornwolff.de

***Abstract*—Manufacturers of household appliances have different opportunities to sell or rent their devices not exclusively to private consumers but also to companies and the gastronomic environment. Transforming those devices into IOT devices by making them "smart" with the concept of a "digital twin" enables manufacturers to utilize new technology and offer new payment models to their customers. This allows the development of an ecosystem in which new business processes can emerge and serveral stakeholders can participate: A platform that connects those smart devices and handles the communication, data persistency, security and the automated processing of the business logic. An evident approach for a suitable solution to create such a platfrom might be the application of blockchain technology - or in general the distributed ledger technology. Due to their characteristics like distribution, no-trust environment, integrity of data and the ability of automated code-processing it is worth investigating on that topic.**
In this paper the synergy between IOT and DLT is shown by the analyzation of a representative, practical IOT use case and the prototypical implementation based on the blockchain platform Ethereum.

[1]

***Index Terms*—Blockchain, Distributed Ledger Technologies, Internet of Things, State Channel**

I. Pay-As-You-Use renting of gastronomy devices

The use case investigated in this paper is as follows:
Manufacturers offer their smart household appliances on a platform where customers can rent those devices and pay for the usage by a pay-as-you-use payment model. Therefore the devices have onboard sensors which are able to track and analyze the usage and the internal machine state. Furthermore other stakeholders like service-providers or suppliers take part on the platform and offer their services. The renting contracts as well as the service contracts - like the service provided by the service-provider or the supply contract for supplying customers with goods - is automatically processed by the platform. This includes the communication with the devices and their onboard sensors, the payment for consuming services and goods and the information providing to the involved stakeholders.

The present use case focuses on smart coffee machines rented to companies. Imagine the following setup: A company - here in the role of a customer - has an active renting contract with the manufacturer. The supplier is commissioned to deliver the machine as well as optinal goods to the customer. The payment and the delivery tracking are defined through the supply contract which is processed by the platform. As soon as the machine is plugged in at the customer's office and connected to the internet, the employees of the customer can order their coffee. They can authorize as company employees and pay the coffee with their company-wallet via a smartphone app - the communication between their smartphone and the coffee machine is done through near field communication (NFC). The coffee machine is able to check whether the company has enough credits for the coffee to be served. If the machine requires maintenance or if an error occures the service-provider is contacted. This process as well as the payment to the service-provider is fully automated through the platform.

The design of the present use case requires the platform to implement a set of features. To find a suitable platform solution the key requirements were elaborated and transferred into the DLT context:

- Smart-contracts allow automated information processing on-chain.
- Oracle-services provide off-chain information to the DLT.
- Payment is needed for processing invoices of services and goods.
- Asynchronicity empowers IOT use cases.
- Performance is needed by every application and adjusted to the specific use case.
- Encryption needs to be ensured to protect customer data and privacy.

performance

For the evaluation the security and encryption mechanisms in DLTs reference is made to [XYZ]. Those topics are evaluated in detail in the given references and are not in the focus of this paper. The integration of payment can be done via a third-party or by built-in DLT functionality like cryptocurrency or tokens (see [XYZ]).

An important concept that needs to be mentioned is the smart-contract which enables this use case to be fully automated and process contracts and contract-logic onchain as well as providing oracle-services. Due to the fact that most modern DLTs offer smart-contract functionality this topic is not investigated on in detail. The key concept that empowers this use case is the need of asynchroneous communication and information processing. Most IOT use cases have too few ressources for providing user-friendly synchroneous communication. Due to connection-loss, power-outage or less computing ressources IOT devices can not guarantee a constant communication channel to other devices. Pushing the concept of asynchronicity a little further and increasing the out time between connections this is referred to as offline capability. To be able to offer the customers a satisfying user-experience the most critical point is the user interaction - in this usecase namely the serving of the coffee. If this process takes to long or aborts due to errors, the customers will decline the use case and the acceptance decreases as well as the economic efficiency.

Therefore technology of the underlying platform needs a mechanism for this offline functionality: The processing, the payment and the synchronization needs to take place in different time intervals. DLTs as a possible, modern solution offer so called state channels to tackle this issue. With offline communication, hashing and signing algorithms and the utilization of smart-contracts DLTs can ensure security, data integrity and payment processing in this scenario. The concept of state-channels is examined in detail in the following section.

A. State channels for empowering and scaling IOT use cases

Blockchains (4) are a compromise due to their nature. Due to the consensus protocol, independent of block size and network capacitya, a natural limit is given: For example, the Bitcoin network is limited on a block time of 10 minutes by the complex calculation of Proof-of-Work (PoW). If the block size is very small, blocks are propagated very quickly over the network, but only a few transactions can be transferred at once. If the block size is very large, it is very difficult for nodes to synchronize but more transactions can be transferred at once. This limitation of block size and block duration means that in the case of Bitcoin currently (as of 01/2020) on average about 7 transactions per second are possible (30). Other implementations may use other consensus mechanisms and other parameters, but there are also natural barriers there. It becomes clear that with increasing demands mainly to transaction processing per time interval (mostly transactions per second (TPS)) improved performance and new solutions become necessary. (30)

As a possible answer to the scaling problem of blockchains so-called state channels were developed. The aim is to be able to transmit any kind of off-chain processing of status-changing operations, which are typically processed and stored on-chain. Thus the number of of transactions can be reduced and the interaction time between involved parties can be improved. In the context of payments, so-called micro-payments are made possible. State channels, which are limited to payment processing, are called payment channels. These can be faster and cheaper than normal transactions take place. The costs of such micro-payments can be kept very low, since not all transactions must be stored on-chain. (15)

Another big advantage of state channels is the empowerment of asynchroneous transactions. If participants are not in the blockchain network (e.g. due to connectivity problems), then no transactions are carried out. If real-world actions like the opening of a barrier or the processing of an action are dependant from an on-chain status update, then with loss of connectivity the real process will stand still until the connection is restored. State-channels could be used here in order to create an alternative processing that can also be used when the blockchain is not reachable. Some example implementations of state channels are Bitcoin's Lightning Network (34), Ethereum's Raiden Network or the implementation of Neo called Trinity (41).

B. A practical realization of one-way payment-channels with IOT devices

The payment-channels in this usecase are implemented as smart-contracts on the Ethereum blockchain. When a user submits to a rental offering and the renting contract is created, the corresponding smart-contract, that models a rental agreement, creates the data structure on-chain and initializes a payment agreement that is designed by a separate smart-contract. This creates the payment-channel between the involved parties and holds all relevant data.

Initially the payment-channel between the coffee machine and the customer is charged with a predefined amount of credits transferred on contract creation. The payment-channel is implemented as a one-way payment-channel, because the value is exchanged only to the wallet of the machine. Credits that remain in the channel after the machine closes the channel will be transferred back to the customer. Apart from the general concept of payment-channels, the present one implements a redeem function that pays out the transferred credits the receivers. Due to this functionality the payment-channels can be reused and recharged and do not need to be recreated; this is for performance and cost issues.

While the payment agreement data structure is an on-chain information container for the payment-channel, the communication between the participants happens off-chain. The data is only needed for data persistency purposes and for synchronization of the participants. To be able to transfer data from the coffee machines to the employees and back there must be a communication channel besides the blochchain. This is implemented by exchanging data via NFC: After pressing the coffee serving button, the coffee machine sends the payment data as a receipt via NFC to the employee's smartphone. The data is signed with the employee's private key and sent back from the smartphone to the coffee machine. The machine is able to verify the signature and persists the latest receipt locally. This process can be repeated as many times as there are enough credits in the payment-channel. Because there might occure a connection-loss the coffee machine might not be able to check the balance on-chain and has to rely on its local information. Because the payment-channels are one-way, there is no possibility for the customer to pay out his own credits and order a new coffee while the machine is offline. This must be ensured to avoid data inconsistency.

If the balance of the payment-channel reaches a critical threshold or a defined time interval is reached, the machine redeems the receipts and the money is transferred on-chain. The customer - in this use case the company or a authorized employee - can charge the payment-channel again to order more coffee in the future.

The described payment process with the utilization of the payment-channel as part of the renting agreement can be transferred to the other payment processes between more stakeholders like the service-provider mentioned above. The base concept and implementation remain the same.

C. Architectural overview of involved components

ARCHITEKTUR ZEIGEN

II. Results

ERGEBNISSE DARLEGEN

References

[1] Cisco, “Internet of things,” 2016. [Online]. Available: https://www.cisco.com/c/dam/en/us/products/collateral/se/internet-of-things/at-a-glance-c45-731471.pdf