Aalto University

OPEN ACCESS

This is an electronic reprint of the original article.
This reprint may differ from the original in pagination and typographic detail.

Paavolainen, Santeri; Nikander, Pekka

Interledger Demo: IoT Integration

Published in:
IEEE International Conference on Blockchain and Cryptocurrency

DOI:
10.1109/BLOC.2019.8751399

Published: 01/01/2019

Document Version
Peer-reviewed accepted author manuscript, also known as Final accepted manuscript or Post-print

Please cite the original version:
Paavolainen, S., & Nikander, P. (2019). Interledger Demo: IoT Integration. In *IEEE International Conference on Blockchain and Cryptocurrency* (pp. 7-8). IEEE. https://doi.org/10.1109/BLOC.2019.8751399

This material is protected by copyright and other intellectual property rights, and duplication or sale of all or part of any of the repository collections is not permitted, except that material may be duplicated by you for your research use or educational purposes in electronic or print form. You must obtain permission for any other use. Electronic or print copies may not be offered, whether for sale or otherwise to anyone who is not an authorised user.

Interledger Demo: IoT Integration

Santeri Paavolainen* and Pekka Nikander*
*Department of Communications and Networking, School of Electrical Engineering, Aalto University, Finland
Email: {santeri.paavolainen,pekka.nikander}@aalto.fi

***Abstract*—With the spread of pragmatic approaches to blockchains and distributed ledger technologies (DLTs), the need to operate securely across multiple DLTs has become apparent. The class of operations that span two or more DLTs is generally referred to as interledger. Various interledger approaches have been proposed, researched and taken into use, including for example, atomic cross-chain transactions, sidechains, bridging approaches, ledger-of-ledger structures, and layered value transfer protocols, such as the W3C Interledger Protocol (ILP).**

In this demonstration, we exemplify ongoing work in our research group at Aalto University. We demonstrate how a resource-constrained IoT device may be made able to interact with an "IoT-friendly" ledger, with an interledger gateway service bridging to a public permissionless ledger. The demo use case reflects a scenario where an IoT device advertizes a service that is available for anyone on the Internet, pending a successful and validated payment has been made on the Ethereum ledger. This demonstration shows how this can be accomplished in an auditable and secure manner across ledgers using hashed timelock agreements (HTLAs) to provide payment escrow.

***Index Terms*—Interledger; Distributed Ledgers; Internet of Things; IoT; Hashed Timelock Agreement**

I. Introduction

While public permissionless blockchains, such as Bitcoin [1] and Ethereum [2], are often difficult to interact directly from IoT devices due to the large computing, storage or network resource requirements imposed by the blockchains [3, Table I], they also are very enticing for their economic properties. Bitcoin and Ethereum, the two largest public blockchains, have a total market capitalization of over $83 billion.[1] Given the way these blockchains are structured, receiving a payment does not require the device to host sensitive information, such as signing keys. Hence, in theory, a device may instruct a user to perform a payment to a specific address (the owner of the device), and monitor for such a payment to take place. However, such a simple scheme has firstly the drawback of requiring the device to interact directly with the public blockchain, with potentially severe resource requirements, and secondly, the payee has to blindly trust the device to detect the payment and perform the service as paid and requested.

In this demonstration, we show how both of these challenges can be addressed through the use of *interledger protocols* and *hashed timelock agreements*.

This work has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 779984.

[1] https://coinmarketcap.com/, as of 12 March 2019.

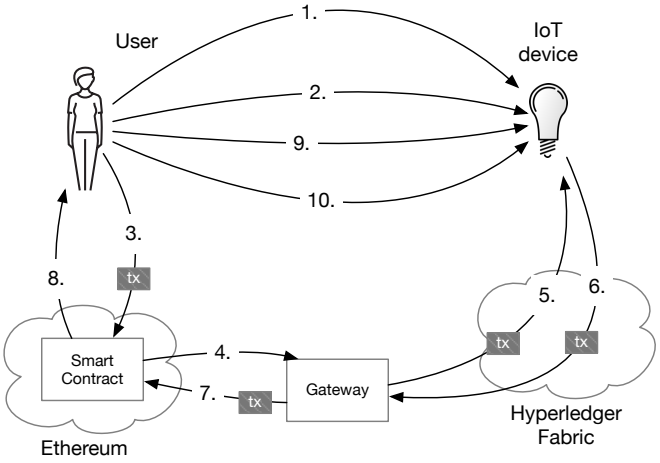

Figure 1: Components and their interactions in the demo.

II. Design and Implementation

The demonstration scenario is based on simple IoT devices that provide device-specific services. In our specific demo setting, we use an RGB lamp and an Internet radio based on Raspberry Pi. The goal of these devices is to be able to publicly sell *time-limited access* to their services. The user of the system, naturally enough, wants to gain access to the service(s). The system also consists of two ledgers — a public one and a private one — and an interledger gateway service that mediates the payment and the access grant protocol between ledgers and the different parties. These are shown in Figure 1. The interactions between the different entities are as follows:

1) During a **discovery phase**, the user contacts the IoT device using a normal web browser, and for example, inspects the service description, terms and conditions, and pricing.
2) Upon deciding on a purchase, the user starts the **payment setup phase**, where, using the user-supplied parameters (e.g. "reserve service for 30 minutes starting at 11:30") the IoT device provides the user with detailed payment information, including the actual fee, a smart contract address to use as the payment escrow service, and the hash of an access token, among other operational parameters.
3) The user will then **initiate a payment** on the Ethereum

blockchain to the escrow service smart contract.[2] This is a transaction on the the Ethereum blockchain, including parameters from the previous step.

4) The smart contract then runs, and **escrows the payment**. The cryptocurrency is effectively now held by the smart contract, and will not be released unless specific conditions are met (see below).
5) The gateway service detects that an escrowed payment was made, and **forwards the payment information** on the private ledger[3] to the IoT device.
6) The IoT device receives a transaction on the private ledger addressed to it, and **validates the payment** information with the payment parameters (from step 2).
7) If the requested resource is still available, the device marks the resource as reserved and sends back a **confirmation transaction** on the private ledger containing the access token matching the hash from step 2.
8) The interledger gateway now checks that the hash of the access token matches the hash it received in step 4, and initiates a **payment release** with the smart contract.
9) The smart contract, receiving a payment release transaction, will check that the agreement deadline has not been exceeded, and compares the hash of the access token with the hash it received in step 3, and **releases the payment** to the IoT device beneficiary (while keeping a cut for the interledger gateway operator.)
10) The user, at this stage, will be able to **extract the access token** from the public ledger.
11) The user signs the access token with their own private key, and use it to **gain access** to the device. In our demo, the access token is used to establish access permissions for the web session only.
12) Finally, the control of the actual service is performed over the regular HTTPS protocol directly between the device and the user. The device will use the reservation information generated in step 2 and marked as confirmed in step 5, and with the per-session access granted in step 9 to enforce the access time limits of the purchase.

The demonstration uses a web application for user interaction, using a Web3-enabled browser set up to operate on an Ethereum test network. The user interface for controlling the RGB lamp, and the RGB lamp itself are shown in Figure 2. The user interaction elements (steps 1–3 and 10–12) are demonstrated through the user interface, while the other steps can be inspected from the gateway console and from an interactive blockchain explorer.

III. Application and Conclusions

In several industrial fields, such as logistics, energy trading, facilities management, and even manufacturing, there is a growing interest to automate currently manual processes with the help of ledgers and IoT sensors and actuators. In many cases, there is a need to a) sense that an event is taking place and recording that in an irreversible way, b) to trigger a contractual transaction as a result of that event, and — often but not always — c) trigger another real-life event. In an automated supply chain, for example, the first event could be a container being lifted from a ship, the transaction recording that the responsibility for the container is being transferred from the shipper to the harbor operator, and the second event allowing the crane to release the container only after the transaction has been recorded.

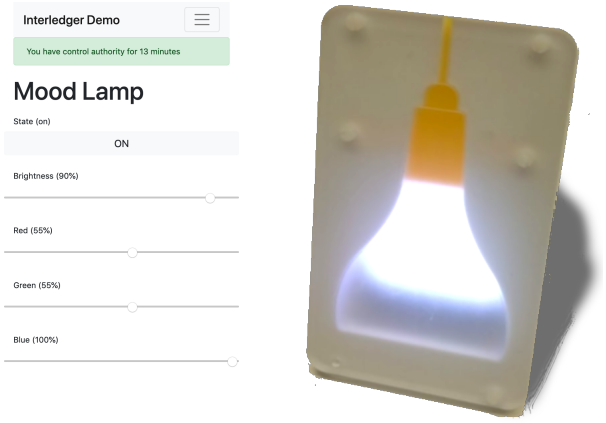

Figure 2: Browser user interface for controlling the RGB lamp, and the actual RGB lamp.

From the business point of view, it is often desirable to record the transactions into a public blockchain so that they cannot be later refuted. However, in many cases, the events are taking place in relative real time, i.e. within a few seconds, while the public public blockchain operations may take minutes or even hours. Furthermore, the companies or consortia involved may want to run their own ledgers, to reduce the risks involved with the public blockchains forking or even crashing.

The present demo has been developed to show how these kinds of aspects can be addressed using interledger technologies to securely enter transactions across public and private ledgers. In the future, we will further generalize and develop the techniques, applying them to the three pilot areas in the EU H2020 project SOFIE.[4]

References

[1] S. Nakamoto, *Bitcoin: A peer-to-peer electronic cash system*, 2008. [Online]. Available: https://bitcoin.org/bitcoin.pdf.

[2] G. Wood, "Ethereum: A secure decentralised generalised transaction ledger," *Ethereum Project Yellow Paper*, vol. 151, 2014.

[3] D. Leung, A. Suhl, Y. Gilad, and N. Zeldovich, "Vault: Fast Bootstrapping for the Algorand Cryptocurrency", presented at the Financial Cryptography 2019, 2019, p. 15.

[2] The user, at this or an earlier stage, could audit the smart contract to verify that it conforms to the payment escrow protocol.

[3] We use Hyperledger Fabric as the private ledger.

[4] See https://sofie-iot.eu.