When Your Sensor Earns Money: Exchanging Data for Cash with Bitcoin

Dominic Wörner
Department of Management
Technology and Economics
ETH Zurich, Switzerland
dwoerner@ethz.ch

Thomas von Bomhard
Institute for Technology
Management
University of St. Gallen,
Switzerland
thomas.vonbomhard@unisg.ch

UbiComp'14 Adjunct, September 13-17 2014, Seattle, WA, USA
ACM 978-1-4503-3047-3/14/09.
http://dx.doi.org/10.1145/2638728.2638786

Abstract
Bitcoin is an emerging technology which allows two entities to exchange value overt the Internet without trust. Embracing that those entities could well be machines we present a system that allows a sensor to offer its measurement data directly to a world-wide data market. Thus, we describe a prototypical implementation of the process of exchanging data for electronic cash between a sensor and a requester by leveraging the Bitcoin network and discuss its current limitations.

Author Keywords
Bitcoin; sensors; data market

ACM Classification Keywords
K.4.4 [Electronic Commerce]: digital cash, electronic data interchange.

Introduction
Bitcoin was introduced in 2008 as a peer-to-peer version of electronic cash and the first system for electronic transactions without relying on trust between transacting parties and without the need for a central authority[2]. This is achieved by ingeniously combining digital signatures, a peer-to-peer network using proof-of-work to record a public history of transactions, called the blockchain, and an incentive scheme alluring early

adopters. Six years later a bitcoin is worth around $500, there are around 60,000 transactions per day, and thousands of developers are contributing to the ecosystem. The latter being possible because Bitcoin is open source, and because Bitcoin transactions are inherently programmable due to a built-in scripting system. Bitcoin can be seen as an API for money and may allow machines to directly take part in an economy. For this reason, we expect that Bitcoin has the potential to stimulate the Internet of Things (IOT). A basic building block of the IOT is the deployment of sensors. However, most of the sensors deployed today are living in private sensor networks only used for single applications. This contrasts the vision of a true *Internet of Things*. Cloud platforms such as Xively[1], Thingspeak[2], and Thingful[3] allow individuals to share their sensor's data, but there is little incentive for sensor owners to provide well structured meta data and reliability. Therefore, there is no way for third parties to leverage today's sensor deployments considerably. Bitcoin as a frictionless Internet-native currency allows in principle for the first time to pay for individual sensor measurements. Attaching a Bitcoin address to a sensor could empower the sensor immediately to take part in a world-wide data market. There is no need for a bank or PayPal account which involves a legal entity. With Bitcoin in contrast, creating a private-public key pair is all that is needed. An example could be a former private weather station with air quality sensors now offers its data. A sports app provider like Nike could then buy this data to suggest its users a pollution-free running track. In this work, we introduce the concept and a prototypical implementation of exchanging measurement data for electronic cash using the Bitcoin blockchain.

[1] http://www.xively.com
[2] http://www.thingspeak.com
[3] http://www.thingful.io

Therefore, we first explain how Bitcoin transactions work and how they can be used to transmit measurement data. Thereafter, we present the system's building blocks and their tasks followed by our prototypical implementation. We conclude with a discussion of limitations and future work.

Bitcoin Transactions

A Bitcoin transaction consists of inputs and outputs. Inputs link to outputs of former transaction. Each entails a script. A script related to an output typically determines the requirements an input script has to fulfill in order to redeem that output. In the common case a transaction is used to transfer ownership of some bitcoins whereby ownership is defined by possession of a private key that is able to redeem the particular output.

We clarify this concept with an example. Let's examine the following output script which is called Pay-to-PubkeyHash:

```
OP_DUP OP_HASH160 <pubKeyHash> OP_EQUALVERIFY
OP_CHECKSIG
```

The amount corresponding to this output can be redeemed by concatenating (from left) an input script of the form:

```
<sig><pubKey>
```

which needs to evaluate to *true*. This is the case if and only if the public key `pubKey` hashes to the demanded `pubKeyHash` (`OP_EQUALVERIFY`) and the signature `<sig>` was created by the corresponding private key (`OP_CHECKSIG`). By interpreting `pubKeyHash` as a Bitcoin address, we now understand what is meant by sending bitcoins to an address. But more interestingly we see that

due to the scripting capabilities there are much more sophisticated transaction types possible. An important example is a multi-signature transaction which demands the signature of more than one private key. Recently, also the first implementation of assurance contracts[1], which can be understood as a trust-less decentralized way of implementing Kickstarter-like crowd funding with Bitcoin, was introduced[4].

In a first implementation of the data exchange, we aim for including the data directly into a transaction which will eventually be included into the blockchain. Notably, a Bitcoin transaction provides no field to add meta data. A loophole to add data nevertheless would be to add an additional output and encode the data in the form of a public key hash. However, this has the following drawbacks. First, the amount of data is strictly limited and second, since every node in the Bitcoin network thinks this is a valid unspent transaction output it will be held in its memory forever. We can evade the latter drawback using the following output script:

```
OP_RETURN <data>
```

The `OP_RETURN` operator assures a Bitcoin node that this output is unspendable and therefore doesn't need to be held in memory. Nevertheless, it will be stored on disc as part of the blockchain. Currently the Bitcoin reference implementation allows to add 40 Bytes of custom data.

System Overview and Requirements

Our system consists of three parts. A sensor client, a requester client and a sensor repository. How these parts interact with each other and with the Bitcoin network is illustrated in Fig. 1.

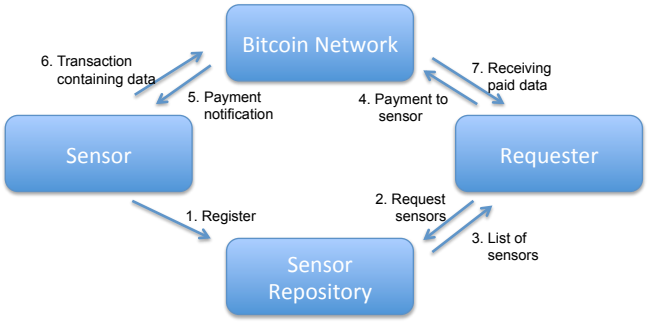

Figure 1: Process for exchanging data for bitcoin.

Sensor Client
The sensor[5] client needs to fulfill the following tasks. It needs to note a data request by receiving bitcoins and it needs to be able to create and publish a transaction containing the requested data.

Requester Client
The requester client needs to be able to send bitcoins to the sensor's Bitcoin address. Further, it has to note the subsequent transaction of the sensor containing the requested data.

Sensor Repository
In addition, we propose a sensor repository where sensors can be registered in oder that they can be found by requesters. An entry in the sensor repository should contain at least the Bitcoin address, which data it offers, the price, and additional meta data like location, tags, etc.

[4] http://blog.vinumeris.com/2014/05/17/lighthouse/

[5] In this work we understand as a sensor a complete system entailing a sensing unit and a computing unit with Internet connectivity. However, they may not be in a single device.

Prototype Implementation

Sensor Client

In order to note a payment and therefore a data request, we implemented a websocket client which registers with a websocket API. The websocket server relays transactions containing the sensor's Bitcoin address. Subsequently the transaction is parsed for the requester's Bitcoin address. In the general case, there might be multiple inputs with different addresses. However, in this first implementation we assume that the requester uses only a single bitcoin address for spending. Thereafter, a transaction with a Pay-to-PubkeyHash output containing the requester's Bitcoin address and a unspendable output containing the current measurement data is created and published to the Bitcoin network[6].

Requester Client

The requester client retrieves sensors from the sensor repository. The user then selects the desired sensor and a payment is made to the sensor's Bitcoin address. Another websocket client waits for the transaction from the sensor entailing the data. On arrival, the data is decoded and presented to the user.

Sensor Repository

The sensor repository is implemented as a database with a RESTful HTTP API and a web front end. Both can be used to register a sensor or to search for a sensor. An example for a sensor search would be to search for a keyword and a location. The response then entails a list of sensors meeting those criteria.

[6] In practice there is another Pay-to-PubkeyHash output containing the sensor's Bitcoin address to absorb the change since all inputs have to be spent completely.

Discussion

We presented a basic concept and its prototypical implementation for the exchange of data for electronic cash between a sensor and a requester. The vision thereby is to build a decentralized Sensing-as-a-Service infrastructure where a sensor can directly offer its measurements to a world-wide data market. Our first concept aiming for demonstrating the most simple process using the Bitcoin network has certainly limitations. First, the purchased data is publicly available in the blockchain which would stimulate free riding. This could be addressed by encrypting the data with the requester's public key whereon the requester client decrypts the data using its private key. Second, in order to provide the measurement data immediately we accept zero confirmation payments which a requester may be able to exploit by an attempt to double spend. Third, there are obvious scaling issues. On the one hand the blockchain would be bloated with transactions and exchanged data would be stored forever on every full Bitcoin node. On the other hand a sensor would have a large number of tiny unspent transaction outputs which are expensive to spend. However, cryptocurrencies, the concept of programmable money and blockchain technologies are still in their infancy. Developments like micro-payment channels, side chains or tree chains to name only a few promise to ease the issues with long confirmation times and scalability. Thus, we expect that it is only a matter of time until machines not only exchange data but also money. This opens up a whole new dimension for ubiquitous computing.

References

[1] Bitcoin Wiki. Contracts, 2014. [Online; accessed 26-May-2014].

[2] Satoshi Nakamoto. Bitcoin: A peer-to-peer electronic cash system, 2008. [Online; accessed 26-May-2014].