Towards a decentralized, trustless infrastructure for brokered data traffic metering to enable IoT data marketplaces

someone else
School of Computing
Newcastle University
Email: paolo.missier@ncl.ac.uk

Homer Simpson
Twentieth Century Fox
Springfield, USA
Email: homer@thesimpsons.com

James Kirk
and Montgomery Scott
Starfleet Academy
San Francisco, California 96678–2391
Telephone: (800) 555–1212
Fax: (888) 555–1212

***Abstract*—The abstract goes here.**

I. Introduction

Individuals who own and operate personal IoT edge devices (wearables, smart home etc) should be able to retain some control over the data that is continuously generated by those devices. Increasingly, such data are considered valuable digital assets that could be traded with third parties: There are third parties that could benefit from using that data, and the challenge is in allowing them to access it under the conditions that data owners find acceptable. [Misura, Kresimir, and Mario Zagar. 2016] Such value can be found both in aggregate form, through analytics over large number of data streams, on a per-individual basis. For example, the density of personal travel card swipes over time at metro stations in London will be of interest not only to the transportation authority, but also to taxi companies who need to decide how to position their cars outside mero stations. Similarly, aggregate smart metering of commodities (water, energy) may enable providers to optimise their delivery. On the other hand, an indviduals fitness data, as recorded by smart phones or dedicated fitness devices, may be of interest to health insurance companies {*[CITE the case of UK insurance and applewatch deal]*}.

In this paper, we explore the idea of a marketplace for IoT data which enables device owners not only to control the distribution of the data produced by their devices, but to trade them with third parties. Initially As in any marketplace, IoT data trading should be governed by an agreed-upon set of rules, set upfront by some trusted authority, which determine what kinds of contract and transactions are acceptable, and stipulate sanctions when the rules are not met. The main novel element of the proposed marketplace is its decentralized, authority-free, self-regulating nature. This is reflected in the way rules are automatically enforced, fairness of the transaction guaranteed, and sanctions imposed. This proposition is based on two technology elements. The first is an ehancement to IoT inrastructure, to enable the systematic monitoring of data streams as they flow from devices at the edge of the nework to analytics consumers, or Value Added Services (VAS), somewhere in the core cloud infrastructure. The second is the use of emerging consensus-based distributed transaction ledgers, specifically blockchain technology to create a transparent and open audit trail of granular data flow through the network, as well as of smart contracts to enforce marketplace rules, resolve disputes, and settle payments.

A. Related work

B. Contributions

Firstly, we propose a new model and software infrastructure to enable the next generation of IoT data marketplaces through granular metering of IoT data exchanges. These exchanges may occur either through a broker, using a publish-subscribe pattern, or through network servers. In traditional marketplaces, a trusted authority oversees data exchanges as well as guarantees the identity of the paritcipants. The key novel aspect of our contribution is that our model relies upon blockchain technology and smart contracts to remove the need for centralised trust. We discuss the challenges and limitations of using smart contracts for automatic dispute resolution.

Secondly, we present a prototype realisation of the marketplace for both brokered data exchanges, i.e., using the MQTT protocol, and non-brokered exchanges using core network servers and application servers. We use blockchain technology and smart contracts (Ethereum) for enforcing the contracts definition and providing dispute resolution capabilities.

Finally, we evaluate the flexbility of our solution, i.e., the ability for smart contracts to accommodate a variety of contract types, as well as its techical capability to handle a stream of blockchain transactions at varying arrival rate.

II. Brokered IoT data as tradeable assets

Initially, we are going to assume a baseline scenario in which ground rules for governance of a marketplace are set in advance, data exchanges are mediated by brokers, a brokered message constitutes a unit value of digital goods, whose value is determined solely by the type of the message, and the marketplace is controlled by a trusted authority that is able to ensure its fairness.

basic brokered data exchanges – fig may be redundant

Fig. 1. Brokered data exchanges.

Vaughn's arch figure

Fig. 2. Generating traffic cubes

Our reference architecture for brokered IoT data exchanges is sketched in Fig. {xx}

In line with common IoT network architectures {*[... on using brokers for data exchanges]*}, we assume that data produced by edge devices P are routed to consumers C, which we may refer to as *Value Added Services* (VAS), through message brokers. Concretely, in our prototype we have used MQTT {*[...standard etc.]*}

Data producers may be, for example, wearables (accelerometers, glucose monitors, heart monitors, smart energy meters in the home), personal or home monitoring sensors, etc. VAS are analytics applications, for example to detect activity levels of a cohort of individuals over time for health care monitoring or for commercial purposes, detect the current traffic within a building, etc.

Aa raw data stream consists either of individual data tokens, for example each individual swipe of every Oyster card on the London tube, or of windowed aggregations from a stream, for instance a few seconds worth of accelerometry data, or the average GPS position within the window.

VAS may aggregate data from multiple Ps over time. Although VAS may themselves produce new data that can potentially be traded, for the purpose of this paper we only consider the first level, from P to C. We assume that Ps are network edge devices, while the VAS are located in a cloud.

Gateways are responsible for forwarding multiple data streams generated by IoT edge devices to one or more brokers. We assume that participants in the brokered exchange will have agreed on a hierarchy of topics, and that each data type in a stream is associated with one topic. For example, leaf-level topics may be "heart rate","GPS track", "glucose reading", "energy reading" and so forth. Note that topics are the only type of descriptive metadata associated with the messages. The gateways generate a stream of messages from the raw stream of tokens, by encapsulating each token into a MQTT message payload and adding the topic corresponding to the stream type.

VAS (see right side of the figure) subscribe to topics, possibly just for a set time interval. In line with the pub/sub model, providers (the devices) are unaware of which consumers (the VAS on the right) subscribe to their message streams. This model results in a many-to-many broker-mediated message exchange, where each VAS may subscribe to data from multiple Ps, and each message from one PP or P may be delivered to multiple VAS. As MQTT supports multiple QoS levels, we are going to assume complete and accurate delivery of each message.

In this model, each individual message is a tradeable asset, with a value that is defined by marketplace rules. In this work our focus is solely on the enabling infrastructure to ensure fair trading, and we are not concerned with the mechanisms deployed to set and update the price of these assets. Instead, we are going to assume that a pre-defined unit value V_k is associated with each topic T_k. Thus, for n messages with topic T_k sent by producer P_i to the broker during time interval $[t_1, t_2]$, P_i should receive $n \cdot V_k$ units in compensation from each subscriber to T_k, to which the broker will have forwarded the messages. The reward model is therefore very simple:

{*FIXME*}

$$reward(P_i, t_1, t_2) = sum_{T_k, C_j} n \cdot count(P_i, T_k, C_j) \quad (1)$$

{*[note that there may be multiple brokers so this needs adjusting].*} Our goal is to monitor the traffic through each of the brokers, to the extent needed to keep track of, and settle, the balance owed by each VAS to each Pi.

In the next section we describe a centralised model where settlement is performed by a trusted element in the IoT architecture. We are then going to show how removing the assumption that the settlement element is trusted exposes the model to potential attacks, and how introducing settlement through smart contracts helps addressing such attacks.

III. Granular traffic metering through trusted brokers

A. Traffic cubes

As we have seen, in our model the reward for a producer P_i is determined only by the number of messages that flow from P_i to each of its subscribers C_j and each topic T_k. In order to ascribe credit to each producer for traffic exchanged in a window $W = [t_1, t_2]$ is therefore sufficient, in principle, to maintain a count N_{ijk} of such messages for each combination of provider P_i, consumer C_j, and topic T_k that are observed within W. Thus, for each W we want to generate a report consisting of a set of 4-tuples:

$$\langle P_i, C_j, T_k, N_{ijk} \rangle$$

We denote this set of counts as a *traffic cube*, borrowing terminology from OLAP database practice where a "cube" is a table with N attributes, in which the first $N-1$ attributes are dimensions in a database schema (in our case, these are the Producers, Consumers, and Topics) and the last is an aggregation over values in the database for each combination of the dimensions – in our case, a count(). Thus, a traffic cube contains a summary of all data flows observed by a broker. Notice that these cubes only contain metadata, ie the counts, but they disregard the content of the messages.

B. Basic architecture

Using this simple aggregation as a model, we extend a MQTT-based IoT platform to provide the traffic monitoring capability that underpins reward settlement and thus basic marketplace functionality, as shown in Fig. 2.

Firstly, we enhance the MQTT broker continuously logs a summary of each message it observes into a dedicated

Cubes in context

Fig. 3. REDO

database, which we call *TrackerDB*. In our initial prototyping we have used the open source Mosquitto MQTT broker and a Cassandra NoSQL database to ensure scalability.

Secondly, a traffic reporting service generates cubes on demand by querying the *TrackerDB*, in response to requests issued by client applications (including possibly independent third party clients). The service is accessible through a REST interface. A request that only specifies a window W will return the most general traffic cube involving all producers, consumers, and topics. Clients may optionally also define a slice through this potentially large cube, by specifying a subset of the available producers, consumers, and topics.

Finally, a settlement service uses the cubes provided by the traffic reporting service to create, for each window W, a set of payment transactions from each consumer to each producer, based on the reward model (1). In the baseline scenario we have been presenting, where we have assumed there is only one broker, which is a trusted component, setllement is straightforward, as the settlement service has complete visibility of the entire traffic over time, and it can assume that the cubes are complete and correct. Note that, under the same trust assumptions, settlement extends easily to a more realistic scenario where multiple brokers are deployed, each enhanced with the same logging capabilities and local traffic reporting service, as in Fig. 3.

C. Removing trust using blockchain and smart contracts

Any marketplace where the reward model is based on message counts is vulnerable to malicious behaviour by any of the participants. Specifically, the suppliers (the publishers) have an incentive to claim to have produced more messages than in reality, while conversely, subscribers have an incentive to under-report the number of messages they receive.

If we remove the assumption that the broker(s) are trusted, we must also accept that they may be prepared to collude with any of the participants, and thus deliver traffic cubes that may not be correct or complete. Discovering such collusions may not be possible when the broker is the only source of traffic counts available to the settlement service. At the same time, resolving any disputes amongst pairs of participants requires a public and irrefutable record of the reported traffic.

We propose a two-steps approach to address both these problems. Firstly, we remove the assumption that traffic cubes are generated by the broker alone, and instead require each marketplace participant to provide to the settlement service their own view of the data traffic they have sent (publishers) and received (subscribers). Note that each such view, even when it is faithful and not malicious, will not be complete, as it can only reflect the partial traffic information as viewed by each publisher and each subscriber. Secondly, we use Smart Contracts to realise the settlement service itself, in such a way that every traffic cube received by any of the participants is posted on a blockchain as part of a transaction.

This approach provides at the same time transparency and accountability, because the content of the blockchain is public and can be inspected, and a way to address disputes, because for each window W, multiple (partial) views of each cube are made available to the settlement service.

Our initial, and partial, Proof of Concept realisation of this approach aims to show the potential of smart contracts technology in this setting. Methods for addressing disputes are briefly discussed in the last section, but are still at the experimental stage.

{*[explain the concept of eth transactions as extension of bitcoin transaction]*}

{*[say what is a transaction in this setting – ie between a marketplace coordinator and the settlement service, how it works in practice, how much it costs. Traffic cube are the message payload]*}

IV. Initial Proof-of-Concept Realisation

{*Proof of Concept using Eth*}

[PM] We need reference to ETH architecture, with APIs call and a snippet of the code. We can add some numbers on how much this will cost to run and consequently set up a minimum cost for each data in order to keep infrastructure sustainable. Logic: data is a count cube. The platform generates a stream of these cubes at a certain rate, which is tunable using the window size on the TrackerDB. The arrival rate of the cubes determines the frequency at which contracts are executed, and therefore the cost over time.

V. Evaluation and Lessons Learnt

[PM]

- Are smart contracts an adequate implementation model to realise a fair marketplace?
- Are there limitations in the reconciliation phase?
- Cost of operating and marketplace: executing transactions and how to control them – contract activated in an adaptive mode. Who owns the contracts? (ideal answer: nobody. Participants share the cost of transactions)
- Scalability: how the cubes decouple the data flow rates from the transaction frequency
- Data marketplace model is preliminary and not validated on real world use cases. It is based on minimal data semantics (ie the topic) and has no notion of more sophisticated contract models.

VI. Conclusion and Future Work

Acknowledgment

The authors would like to thank...

References

[1] H. Kopka and P. W. Daly, *A Guide to LaTeX*, 3rd ed. Harlow, England: Addison-Wesley, 1999.