Smart Contracts for Incentivizing Sensor Based Mobile Smart City Applications

John Lindsay
Dallas, TX
johnl@stopclusters.com

Background

"One of the issues that arise when developing a crowdsourcing monitoring application is the fact that we rely on the measurements taken by the users, but, *how to engage users to use this application* and provide observations?"[1].

Smartphones have become an indispensable item for people and are carried with people throughout the day. They are often embedded with various sensors such as accelerometers, magnetometers, barometers, microphone, humidity, light sensors, cameras, along with GPS. "Crowdsourced" smartphone sensor based applications have been developed that detect potholes[2], cluster areas of harsh braking while driving[3], provide input for contagion spread risk[4], and audio mapping[5].

In order for some applications to be effective and provide a return on investment, they need a threshold level of usage. Unfortunately, a 2016 article states that approximately ¼ users abandon an application after one use and users who open an app more than eleven times is in the 30% - 40% range[6]. Apps and Smart City apps have relied on engagement/retention approaches such as citizen loyalty, enticing content such as news feeds, and gamification. Early results of a recent study suggested that loyalty points appear to have driven participation in usage of an app that sought to influence health behaviors with the ROI being lower health care costs to the government[7].

This paper proposes consideration of paying users to use "smart city" apps in circumstances where the data would have some economic value to the municipality, in proportion to that economic value. The proposal focuses on incentivized usage of the apps with minimal modification from their current state and minimal additional operational overhead for the municipality. As such, sensor data and reporting would still be stored at its current location, not on blockchain.

Smart Contracts Background

Ethereum, the platform on which this was tested, is a blockchain-based distributed computing platform and featuring "smart contracts." Effectively, Ethereum may be viewed as a decentralized currency with an associated Turing complete scripting language centering around storage and transfer of the cryptocurrency. The "smart contracts" are programs executed by nodes on the blockchain in the Ethereum Virtual Machine.

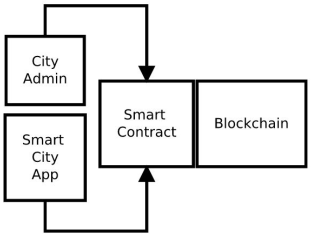
Ether is unit of currency, serving as to compensate the nodes for storage and processing of Smart Contracts as well as a transfer of value among users. Programmatic transfers of ether is supported, which enables reduction of payment infrastructure overhead.

978-1-5386-5959-5/18/$31.00 © 2018 IEEE

As of the time of this writing, one Ether equals $430.46 USD. Further, the smallest denomination of Ether, a Wei, is 1x10^{-18}, or one quintillionth of an Ether, supporting microtransactions. Ethereum blocks are of variable size and added in ~15 second intervals, and supporting ~15 transactions per second. Data not written in a current block is queued for recording in subsequent blocks.

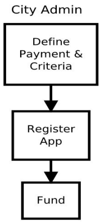
A complete application incorporating a “Smart Contract” is often referred to as a “Decentralized App” or “dApp.” The general application architecture for a Decentralized App is a backend Smart Contract, which contains the core logic, and a front end, such as a web browser or mobile client, which interacts with the smart contract[8].

Proposed Application Architecture



A smart contract is developed and deployed to the blockchain. The smart contract holds the logic for administration of the smart city apps that would participate in the user incentive program, administration of the interface for the smart city app, storage of the app and user events data to the blockchain, and disbursement of cryptocurrency to app end users.

Initial Steps for City Administrator



An interface is provided for the city administrator for the “Smart City Platform” for the initial setup for the integration of the smart city into the payment platform. The first step is to input the payment amount and payment basis. The city administrator would decide on the payment parameters for the particular smart city app, with the key parameters being the amount of payment and the payment basis. Ostensibly, the amount of payment and payment basis would be based on the potential return on investment of the nature of the problem address in the app and the criteria would be based. Expected and tested app usage payment bases would be on a per event basis, time basis, and direct or indirect quantity of data basis.

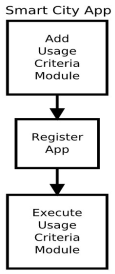
For example, the above mentioned app that employed loyalty points to influence health behaviors references risk factors with modifiable behaviors of not getting a flu shot last year and physical inactivity[9]. Not getting a flu shot is a (lack of a) discrete behavior with a discrete, verifiable solution (ie a receiving a flu shot) whose impact has been quantified in the aggregate (eg in the form of absenteeism from work or university). A payment amount might be some currency on a per event basis (ie a receiving a flu shot). Physical inactivity is a (lack of a) discrete behavior with a solution of periodic exercise. A payment amount might be some currency on a time basis (ie per hour of activity).

For example, the ROI of the app that monitors drivers and maps clusters of location with harsh braking[10] is less collisions and, in turn, less police and EMS deployment. A time based payment basis might be appropriate. For the mentioned audio mapping application[11], amount of data points might make sense.

After a payment amount and payment basis are input into the smart contract, the identifiers for the app are created, so that the app can can be uniquely identified by the smart contract. The identifiers for the app are stored in the smart contract and available for the city administrator to provide to the smart city application developer.

To complete the setup from the city administration end, Ethereum is transferred to the smart contract for funding application usage.

Initial Steps for Smart City Application Owner

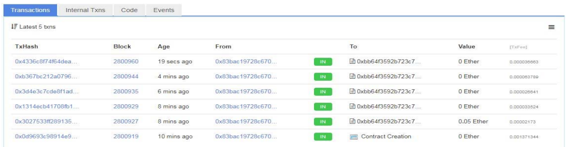


The city administrator relays the usage criteria and application identifiers to the application developer so that the application may be updated for the payment incentive. The application is updated to provide direct or indirect information to support the usage criteria and payment basis. For example, where the payment basis is an event basis, the smart city app should be modified to submit the event occurrence information to the smart contract. Where the payment basis is a time basis, the smart city app should be modified to monitor usage and submit periodic time usage information to the smart contract. Where the payment basis is a data basis, the smart city app should be modified to submit data and/or metadata to the smart contract.

The relayed application registration information is used to identify the app to the smart contract for interaction. In usage, the application is also updated with a “Settings” screen to receive a user’s Ethereum wallet address and submit that address to the smart contract. When the updated application is deployed and executed, it interacts with the smart contract, providing users' usage information, where the smart contract evaluates the usage information and transfers ether when the usage criteria is met and the smart contract is funded.

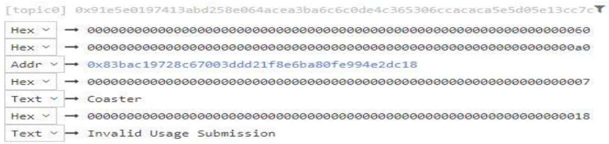
Proof of Concept

A rudimentary smart contract and display interface supporting basic functionality described above supporting input for receipt of funds, adding eligible users, setting an application name/description/key, payment basis, and application usage events from the smart city app was deployed to the Rinkeby test network. Acting as a city administrator, an application was defined and a payment basis and amount was setup. As the author is the developer of the above referenced harsh braking map clustering application, acting as an application administrator, rudimentary modifications were made to a version of it to submit simulated user usage events. Acting as an end user, the app was used, with a usage event submitted to the smart contract. The smart contract history supporting the below screen captures can be viewed on the test network at the URL in the footnote[12]. The first figure below shows a sequence of actions in the smart contract, with the most relevant being the smart contract creation, funding it, adding authorized users, and disbursement of funds to a user for a usage event.



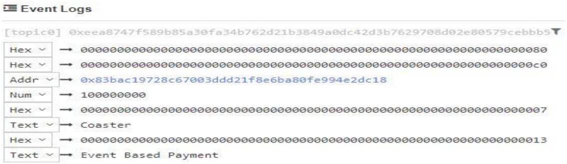
TxHash	Block	Age	From		To	Value	[TxFee]
0x4336c8f74f64dea…	2800960	19 secs ago	0x83bac19728c670…	IN	0xbb64f3592b723c7…	0 Ether	0.000036663
0xb367bc212a0796…	2800944	4 mins ago	0x83bac19728c670…	IN	0xbb64f3592b723c7…	0 Ether	0.000063789
0x3d4e3c7cde8f1ad…	2800935	6 mins ago	0x83bac19728c670…	IN	0xbb64f3592b723c7…	0 Ether	0.000026641
0x1314ecb41708fb1…	2800929	8 mins ago	0x83bac19728c670…	IN	0xbb64f3592b723c7…	0 Ether	0.000033524
0x3027533ff289135…	2800927	8 mins ago	0x83bac19728c670…	IN	0xbb64f3592b723c7…	0.05 Ether	0.00002173
0x0d9693c98914e9…	2800919	10 mins ago	0x83bac19728c670…	IN	Contract Creation	0 Ether	0.001371344

The second figure shows an attempted invalid user submission.



```
[topic0] 0x91e5e0197413abd258e064acea3ba6c6c0de4c365306ccacaca5e5d05e13cc7c
Hex  → 0000000000000000000000000000000000000000000000000000000000000060
Hex  → 00000000000000000000000000000000000000000000000000000000000000a0
Addr → 0x83bac19728c67003ddd21f8e6ba80fe994e2dc18
Hex  → 0000000000000000000000000000000000000000000000000000000000000007
Text → Coaster
Hex  → 0000000000000000000000000000000000000000000000000000000000000018
Text → Invalid Usage Submission
```

The third figure shows a record of a usage event on the test network blockchain, with the most relevant usage information including the payment basis, the users wallet address, and the amount transferred.



```
Event Logs
[topic0] 0xeea8747f589b85a30fa34b762d21b3849a0dc42d3b7629708d02e80579cebbb5
Hex  → 0000000000000000000000000000000000000000000000000000000000000080
Hex  → 00000000000000000000000000000000000000000000000000000000000000c0
Addr → 0x83bac19728c67003ddd21f8e6ba80fe994e2dc18
Num  → 100000000
Hex  → 0000000000000000000000000000000000000000000000000000000000000007
Text → Coaster
Hex  → 0000000000000000000000000000000000000000000000000000000000000013
Text → Event Based Payment
```

Based on the usage event payload being ~62 bytes, possible user monitoring between usage payload submissions being a similar size, and the nature of usage events or use monitoring not requiring low latency indicate that scalability is not a likely issue.

[1] “Uptake of an Incentive-Based mHealth App: Process Evaluation of the Carrot Rewards App” available at https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5470010

[2] Wired, “'Street Bump' Android app detects and reports potholes” available at http://www.wired.co.uk/article/street-bump-app

[3] Smooth Driver Monitoring App, available at https://stopclusters.com/ (developed by the author)

[4] Wello, “Humidity Impacts Infection Spread and Susceptibility” available at http://www.welloinc.com/wp-content/uploads/2014/06/Our-Health-and-Indoor-Weather.pdf

[5] Entertainment Computing - ICEC 2012 : 11th International Conference, ICEC 2012, Bremen, Germany, September 26-29, 2012. Proceedings, “Mobile Application for Noise Pollution Monitoring through Gamification Techniques” available at https://www.researchgate.net/publication/259052847_Mobile_Application_for_Noise_Pollution_Monitoring_through_Gamification_Techniques

[6] “23% of Users Abandon an App After One Use” available at http://info.localytics.com/blog/23-of-users-abandon-an-app-after-one-use

[7] “Uptake of an Incentive-Based mHealth App: Process Evaluation of the Carrot Rewards App”

[8] See the dApp Front End Steps diagram in the “A 101 Noob Intro to Programming Smart Contracts on Ethereum” post, available at https://medium.com/@ConsenSys/a-101-noob-intro-to-programming-smart-contracts-on-ethereum-695d15c1dab4

[9] “Uptake of an Incentive-Based mHealth App: Process Evaluation of the Carrot Rewards App”

[10] Smooth Driver Monitoring App

[11] “Mobile Application for Noise Pollution Monitoring through Gamification Techniques”

[12] https://rinkeby.etherscan.io/address/0xbb64f3592b723c7f83b2faad158a50282683a787