Hindawi
Wireless Communications and Mobile Computing
Volume 2019, Article ID 7317019, 12 pages
https://doi.org/10.1155/2019/7317019

Research Article

The Last Man Standing Technique for Proof-of-Location in IoT Infrastructures at Network Edge

Marat Zhanikeev

School of Management, Tokyo University of Science Fujimi 1-11-2, Chiyoda-ku, Tokyo, 102-0071, Japan

Correspondence should be addressed to Marat Zhanikeev; maratishe@gmail.com

Received 22 February 2019; Revised 3 May 2019; Accepted 30 May 2019; Published 24 June 2019

Guest Editor: Ali Ebnenasir

This paper is divided in two parts. The first part is the story of a failure, where an attempt to generate verifiable Proof-of-Location from WiFi scan results has failed, even when scan results were compared with a small time shift within the same trace. In the second part, this paper proposes to conduct Proof-of-Location transactions in a peer-to-peer fashion. Each peer is assisted by the cloud side which plays the role of both the real-time mediator and public transaction ledger. This paper proposes the Last Man Standing (LMS) procedure which is both a means for ensuring a fair transaction and a natural way to close it. Each transaction results in a coin which can be either shared among transaction participants or owned individually by LMS. Analysis using real mobility traces from various types of urban locations shows that the proposal is valid and will ensure that all the locations within the city will gradually be claimed via the proposed type of transactions while providing independently verifiable proofs for each location. The distant goal of this paper is a next generation Location-Based Service (LBS) which takes the form of a location-based resource economy where each location is a coin compatible with traditional blockchain operations.

1. Introduction

There is a traceable development path for Internet of Things (IoT) starting from small-scale IoT spaces [1], then moving to the scale of a Smart City [2] and recently arriving at the topic of *security at network edge* where majority of discussion is in some way related to the blockchain technology [3]. When it comes to IoT and, broadly speaking, infrastructure at *network edge*, both the conventional form based on Proof-of-Work (PoW) [4] and lightweight methods better suitable for the resource-starved network edge [5] are discussed.

Amidst numerous literature, there is discussion on the fairly narrow topic of Proof-of-Location (PoL) [6–8] as a technology specifically useful at network edge, as it makes it possible for the various location-aware services to connect users to verifiable locations, while at the same time allowing users to retain their privacy. Note that PoL is not an alternative to Proof-of-Work (PoW) or Proof-of-Stake (PoS)—the two low-level approaches to attaining consensus in distributed multiparty transactions. The consensus in PoL is entirely about being able to identify a given location and generate a variable *hashkey* that can be confirmed by other parties once made available publically.

This paper treads along the same general line as in [6–8], which pack WiFi Access Points (APs) found in WiFi scan results into a spatial-temporal (trajectory of WiFi APs in time) trust structure and use it as PoL. However, this paper shows that, unless one assumes that WiFi APs play active role in supplying and validating locations, such a method cannot generate reliable hashkeys in practice. See Section 4 for details on experiments and reasons why this method cannot be used for PoL.

Note that [6, 7] do make this assumption, in fact assigning the Witness role to WiFi APs. To make this work in practice, one would need to force major upgrades on existing WiFi infrastructure. This paper assumes that traditional WiFi infrastructure is used. Instead, the PoL in this paper uses GPS to identify a rough location and then employs peers as Witnesses and cloud side as mediator (and public transaction ledger) during the process of generating a verifiable PoL.

The proposal in this paper has the following original features:

(i) this paper builds a bridge between wireless positioning (outdoors and indoors) and its applications as a PoL method, where the first step in this paper is to

show that wireless context generated by using WiFi signal is not sufficiently reliable to generate consistent hashkeys;

(ii) this paper proposes a method for generating (blockchain) blocks via real-time transactions performed by peers at network edge and works for any number of peers starting at two—the number of peers varies in a wide range in what is referred to as *dense wireless spaces* in literature [9] and this paper assumes that modern urban settings fall neatly into this category;

(iii) the core of the proposal in this paper is the *Last Man Standing* (LMS) procedure which describes the auction-like method for claiming ownership (*mining* in blockchain terminology) of a hashkey that uniquely identifies a given location;

(iv) various forms of ownership are formulated, supporting both single-peer and multipeer ownership models.

The main practical scenario in this paper is as follows. Urban (but basically the entire world) areas are split into a GPS grid by using rounded up numbers for latitude and longitude. Each grid cell needs a verifiable PoL assigned to it—the main assumption here is that publically verifiable hashkeys (the same transaction reflected in multiple transactions involving multiple people) are preferred to a centralized service issuing hashkeys based on an undisclosed internal logic. The LMS PoL proposed in this paper guarantees that procedures for discovering new/unknown locations can be defined as multiparty transactions with clear start and end points. Although this paper focuses mainly on the PoL function itself and leaves the blockchain part out of scope, this paper briefly discusses how PoL hashkeys can be linked into a chain (or even interconnected map) by linking together geographically neighboring locations. Note that this is not a traditional form of blockchain, but the basic blockchain mechanism, namely, hashing previous hashkeys as part of newly generated blocks, is retained.

Finally, this paper uses the term *coin* to describe PoL hashkeys, which is a link between the proposal in this paper and the various future practical applications for PoL. This paper discusses how coin/hashkey owners can collect micro-royalties from digital services running at that location. For example, micro-payments can be made to coin owners for each request to verify a given location—the coin in this case is both the ID and the record of how the location was verified originally. Naturally, secondary verification does not need to be based on multiparty transactions in the physical world—the proof in the original (first) PoL transaction is sufficient for all further verifications. Coins representing locations can also be treated as actual cryptocurrency in which case the system should allow for the transfer of ownership across users.

The proposal in this paper fits into the above scenario and offers the following unique advantages. There is no need for smart WiFi AP or any smart infra; in fact, users can communicate to the cloud only using 3G/4G/LTE connections. The proposal is completely distributed as it relies on peers to pose as Witnesses to a transaction. Hashkeys generated at the end of each transaction can be verified independently by each participating peer and, if made part of a blockchain, can be verified publically. Transactions are also private for each peer relative to other peers, while the cloud can choose to hide but otherwise has access to all the identities. Ownership of hashkeys (one coin per location) can be claimed either by the LMS or shared among all transaction participants—the proposal can accommodate both forms.

This paper has the following structure. Section 2 offers background on blockchain and explains PoL in detail. Section 3 discussed all the related work. Section 4 shares the experience from conducting experiments towards using WiFi APs as *wireless context* and failing in the end. This failure led to the proposal in Section 5, which shows how PoL transactions can be conducted in a cloud-assisted peer-to-peer (p2p) manner. Trace-based analysis of the proposal is offered in Section 6. Section 7 compares the proposal to three similar methods in recent literature. The paper is concluded in Section 8.

In terms of the data, mobility traces used for analysis are available at [10] while specifically the density map and other data structures generated on top of the raw traces in [10] are, along with detailed descriptions, are publically available at [11].

2. Background on Proof-of-Location

Within a larger area of Location-Based Services (LBS), this paper focuses on the problem of defining locations in such a way that they can be verified in a distributed and private manner. The problem itself is understood well in the current literature [4, 6, 7], yet, as this paper shows, there are still unresolved issues.

Before understanding the uniqueness of the Proof-of-Location (PoL) problem from other Po∗ acronyms, one has to understand the role of Proof-of-Work (PoW) and Proof-of-Stake (PoS) in mobile environments.

Traditional blockchain requires the use of Proof-of-Work (PoW) primitive by all network peers [12]. The "work" part of the name refers to heavy calculations that have to be performed in order to generate a single block. For Bitcoin, one block (= one coin) is generated once every 10 minutes, regardless of the number of participants in the peer-to-peer (p2p) network at the time. With active peers in hundreds of millions, it is obvious that the probability that a given user generates its own coin is very low. One can improve the odds by accumulating hardware resources and, thus, represent a larger portion of the network (in terms of computing power).

IoT devices at network edge simply cannot handle such heavy computations, even when a lightweight form of PoW is used [3]. Here, Proof-of-Stake (PoS) was proposed as an alternative to PoW [5] and normally needs only a very small fraction of power required by PoW. Even under PoS, proposals in literature make certain that blocks are connected into chains which can be publically verified by any other user or even made publically available.

In spite of its name, Proof-of-Location (PoL) is not a rival of PoW or PoS. Instead, PoL is simply defined as a

method that can successfully verify one's location. Verification is preferred in a blockchain-compatible way, which means that locations are encoded into blocks and blocks are linked to each other in chains. PoL exists in a larger context of Location-Based Services (LBS) where solid Proof-of-Location can be of value to both end users and services [6, 7].

Section 7 further in this paper offers a detailed comparison between this paper and three closest rival methods. All three methods used in comparison focus on PoL but only two of them use blockchain and explain how locations and the related local context are used to generate hashkeys. As a preview, it should be stated at this point that this paper has a unique blockchain-related feature where blocks are assumed to become *coins* with a clearly defined ownership (a specific client) and, as such, require consistent hashkeys for each unique location. The three rival methods, specifically ADGT [12], PPLV [6], and CLIP [7], do not impose such a requirement, instead discussing a loose form of PoL.

Although the proposal in this paper is based on a blockchain technology and explains the respective parts of the proposal, the issues related to PoW and/or PoS and comparison between the two are left out of scope. Instead, this paper focuses on only the PoL and spends majority of attention on how to build a single block and how its hashkey can be generated in such a way that it would be unique for any specific location.

3. Related Work

There are several years of discussion of the topic of IoT not as standalone devices but as part of *groups* of devices as *network edge* or, even broader, as part of a Smart City infrastructure [1]. Such literature normally discusses various sensor devices and protocols, how cloud services can become part of the communication chain, etc. Such literature normally does not cover the topic of *verifiable* security and/or privacy in such environments, instead focusing on technical implementation. However, it provides a solid proof that wireless p2p communications at network edge are feasible even at current level of technology.

When it comes to urban spaces, the topic of *dense wireless spaces* is an example of how multiple technologies and devices can become part of the same process. The term *mobile clouds* is often attributed to such processes. A good survey of the state of technologies and methods related to mobile clouds can be found in [9], which offers detailed definitions for all the terms used in this paper, such as *vehicular cloud*, *offload*, *network edge*, and others. Again, the subject of verifiable security, such as offered by the blockchain method, is not part of such literature.

Previous work by this author on the subject of grouping of devices at network edge can be found in [9]. Apart from the core proposal, [9] shows how P2P WiFi can be easily implemented in practice using the WiFi Direct technology available on all modern devices powered by Android.

As far as wireless communications go, all the above technologies are classified as *associative* because they require the communicating parties to authenticate (password, accept connection, etc.). On the *nonassociative* side, there is Beacon Stuffing [13] technology which is a separate form, distinct from Bluetooth- or ZigBee-based beacons.

Recent literature identifies the need for verifiable security and privacy at network edge, where a large portion of such literature is dedicated to the topic of Smart Cities [2]. Specifically, the blockchain technology is subject of active current research [4, 14]. There is, however, a major problem with blockchain at network edge. Traditional blockchain is based on Proof-of-Work (PoW) which consumes too much resources in terms of computation load, volume of message exchange, etc. Traditional PoW-based proposals for wireless network edge are in a minority [4] or are made possible by making sure that sufficient computing resources are made available (buses in [14]). Recent proposals offer less resource-hungry methods, often opting for much more lightweight Proof-of-Stake (PoS) [5]. Decoupling data from blockchain calculations is another way to allow PoW at wireless edge [3]. Various theoretical ramifications for consensus at network edge can be found in [4].

For the literature on PoL, the term *wireless context* is a shared feature. Depending on a method, WiFi APs, 3G/5G/LTE base towers, p2p signaling, etc. can be encoded as part of such context [6]. Note that the topic of *network positioning* (also known as network coordination, virtual network coordinates, and its wireless counterparts including *wireless localization*) is its own separate discipline with subdisciplines such as large-scale network coordination and small-scale indoors (normally wireless) positioning. Special WiFi APs which generate specific signal strength distributions were proposed in early days [15] and are found in recent literature as well when using directional antennas is proposed for higher precision [16]. Wireless positioning using only conventional WiFi signals suffers from low precision, which in recent literature is improved in several ways, including using ultrasound [17], physical maps/layouts to use as guidance when inferring positions [18], and others. Precision of inferred positions (2D or 3D coordinates) have drastically improved in recent years and is sufficient for gesture recognition [19], yet the problem of reliability still remains [20]. Section 4 in this paper agrees with [20] and shows via field experiments and data visualizations that generation of consistent hashkeys from wireless context is impossible to achieve in practice.

Some methods related to PoL rely solely on WiFi APs but require APs to play active role (Witnesses) during transactions [7]. WiFi APs combined with mobility data for each user can be used to create a verifiable signature in both space and along the timeline [7]. Note that all these methods assume the presence of *smart WiFi APs* which play active and important role during transactions.

This paper relates to the above topics in the following ways. In respect to associated/nonassociated wireless connectivity, P2P WiFi, Beacon Stuffing, etc., this paper opts for peers communicating via AP SSIDs (the names for APs that show up on WiFi scans). SSIDs broadcast unique IDs assigned to each peer by the cloud side. As such, this method belongs to the nonassociating class, as peers do not have to connect directly to each other.

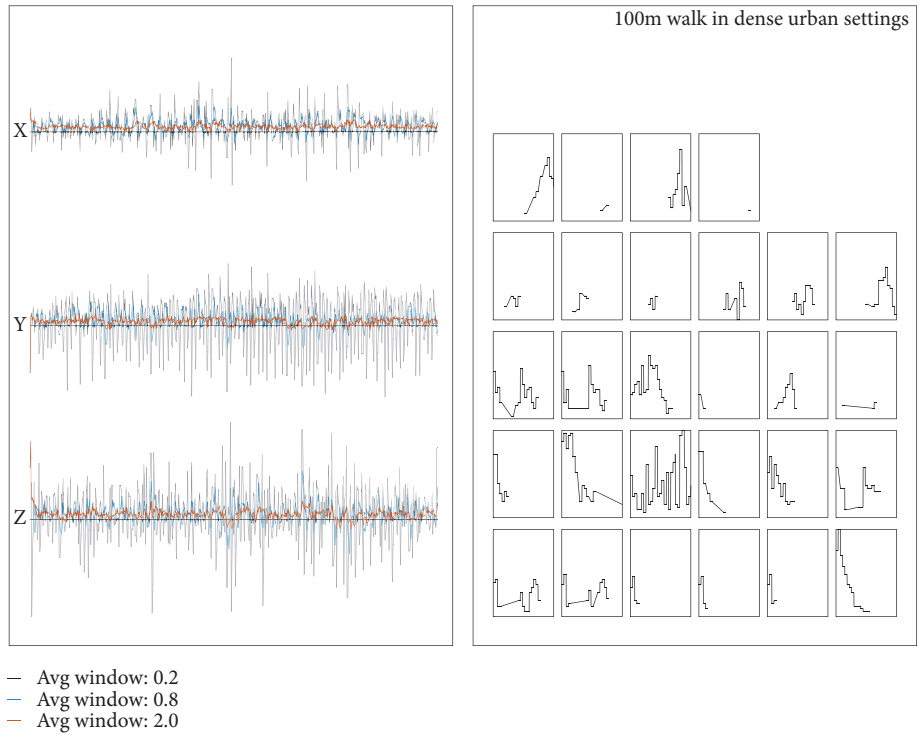

Figure 1: A randomly selected trace captured on a commodity smartphone, showing XYZ coordinates *(left)* passed through three separate smoothing windows, and curves for all the WiFi signals *(right)* captured during the walk.

In terms of PoL, WiFi APs in this proposal are considered to be passive (if used at all). The use of APs is not required—instead the proposal relies on GPS coordinates and APs provided by all peers within the transaction (and local vicinity). By this definition, peers are Witnesses, and the cloud side both is the mediator during the transaction and is used to store publically accessible transaction ledger.

Another distant relation in this paper is to auctions [21]. Related literature is introduced when proposing the Last Man Standing procedure in Section 4.

4. On the Failure of WiFi-Only Context

This section puts much effort into proving that WiFi-only context cannot be used to generate consistent hashkeys. Consistent hashing is a requirement to be able to generate blockchain blocks for PoL which would uniquely identify a given location. This is not necessary for a loose form of PoL which would tolerate minor errors in coordinates. However, this paper assumes that unique locations become blockchain blocks and, later, *coins* owned by uniquely identifiable clients, which imposes strict requirements on uniqueness on hashkeys representing locations.

4.1. Experimental Setup. Following proposals in [6–8], an experiment was conducted to find out whether reliable wireless context can be generated in a passive way ([6–8] assume active role) to be used as a basis for a verifiable PoL. The definition of *verifiable* is simple: we need a way to generate hashkeys in a consistent and verifiable manner. Two elements were used in experiments below: (1) WiFi APs and (2) mobility data generated by accelerometer (the same as in [8]). 10 Android devices of various models were used to collect traces analyzed in this section. The traces are publically available at [11]. Simultaneous capture of accelerometer data (XYZ quantization) and WiFi AP data is difficult since XYZ data can be captured at very short intervals (below 200ms) while Android devices try to avoid frequent WiFi scans (2-3-second refreshes). This is why traces for the two sources are dumped to separate files which are then merged (guided by timestamps) during analysis.

Figure 1 shows an example trace captured by an Android device. The left side shows the accelerometers data split into X, Y, and Z components, each smoothed with 0.2s, 0.8s, and 2s windows. The trace comes from a walk in a dense urban setting which translates into high variations in XYZ data. The right side shows all the WiFi APs encountered during the walk. Each plot represents the time trend of the strength of a given WiFi AP signal. The scales are not shown (to avoid a crowded plot) but they are all the same for all the plots (absolute, not relative frame). Note that only few curves in WiFi plots show the expected trend of growing and falling signal strength, which is to be expected for a WiFi AP that slowly approaches (stronger signal) and then fades away. Majority of trends show complex dynamics—see the discussion on context deltas below, which are the direct effect of this complexity.

4.2. Raw Traces to WiFi Context. To be able to generate a hashkey of the data in Figure 1, we need to parse the data

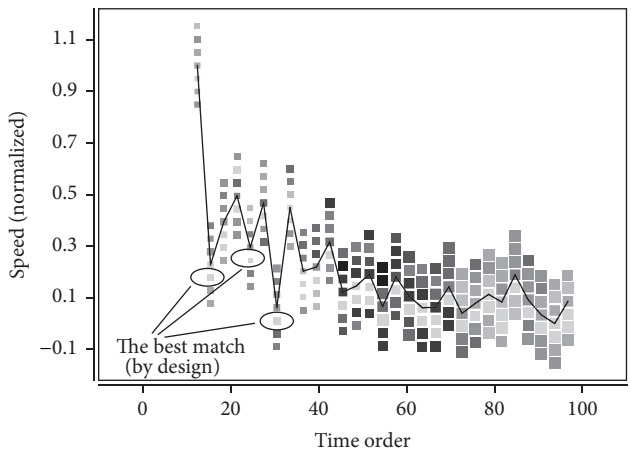

Figure 2: Comparing *context* within a WiFi trace to its own time-shifted copy.

into some kind of *context*. A *delta* between two contexts should ideally be zero for the same location, even if the two compared contexts come from different clients, using different hardware, etc. Failure to achieve zero delta results in different hashkeys, resulting further in different (blockchain) blocks and finally unrelated chains of blocks. It is unclear how to resolve situations when two or more hashkeys claim the same location. Note that the blockchain-based methods to which this proposal is compared [6–8] do not discuss either the subject of uniqueness or ownership of blocks. In fact, this paper offers this element as an original part of the proposal which defines how locations can be *mined* and converted into coins which can further be used as part of a LBS-related resource economy.

Let us first define the term *snapshot* to identify a single outcome of a WiFi scan, which produces a list of WiFi APs in descending order of signal strength. The term *snapshot context* then refers to the *delta* between two such snapshots—it can be binary when only checking for presence/absence of a given WiFi AP or numeric when comparing positions of WiFi APs in the ordered list. For the latter assessment, this paper quantizes the ordered list and uses numbers for order positions in visualizations. The full context then can be defined as such *deltas* plotted along timeline, additionally with speed (calculated as Euclidean distance from XYZ data with or without calibration) added to the plot. This method can both provide the visual image of the context and be used to calculate the numeric distance (aggregate delta) between two contexts.

Note that the term *context* here refers to the aggregate deltas; i.e., it is the visual or numeric representation of the discrepancies between two traces. It is an important distinction since, for reliable PoL, the context should ideally represent mostly empty spaces, that is, having zero or near-zero deltas across the entire plot. Analysis in this section shows that this is not the case in reality.

4.3. Same Trace Experiments. Let us convert the concepts of *delta* and *context* into visual form using Figure 2 as both learning and working example. Figure 2 shows context for the same trace (compared to its own shifted version) captured in a dense urban outdoor space while walking on a straight line for about 100m. For a valid context, we need to compare trace A to trace B. In Figure 2 both A and B are the same trace, but B has first 3s of data removed (generating a time shifted copy). Comparison is also done using the same time increment, which means that trace B is exactly one step ahead. For each 3s step, XYZ values are averaged and speed is calculated as Euclidean distance of these XYZ values, and WiFi APs are aggregated from all the scans (1-2 normally) within the interval.

In addition to the concept of context above, the following visualization method is applied in Figure 2. First, the curve represents the speed trend in time (normalized). At each 3s interval (at each bullet on the curve), vertical blocks represent deltas in the respective snapshots. On the vertical column, each block that further departs from the curve represents comparison with increasing *lag*. Specifically, when, say, the position of the snapshots is 5 (in both traces), vertical column uses lag within the range of -2..0..+2, which represents comparison between snapshots 5 and 3, then 5 and 4, and so on.

Each block encodes two metrics. Its *size* represents the average number of APs between the two compared snapshots, and its *color* encodes the delta itself (darker represents larger delta).

Note that by design (shifted copy of itself), the best possible performance should be expected in Figure 2 for lag=1, i.e., for boxes located directly below the curve (some are shown in the figure). The reality in Figure 2, however, is different. The best matches (by design) are relatively lighter (and shown in annotations), but even they have some nonzero delta. Farther from the curve (larger lag) is not reliably worse, while it should be expected to be. In fact, the overall outcome in Figure 2 is such that no reliable pattern exists in the figure to generate a reliable PoL.

The following reasons are identified for the poor performance above.

(i) As Figure 1 already showed, WiFi strength dynamics are extremely noisy; in the context the noise translates into unpredictable changes in strength-ordered lists, even between neighboring snapshots—this is a known fact and has been identified in recent literature on indoors positioning [20].

(ii) WiFi APs may appear and disappear in scan results interchangeably—now, this can be partially counteracted by keeping an active list and removing items on timeout but this creates a fairly complex logic prone to other types of noises (like too many WiFi APs).

(iii) Scan results are wildly different across different devices (absent/present APs, signal strength, etc.), which was confirmed in raw data (and supported by recent related literature [20]) and concluded that even within the same class of WiFi protocol (g/ac and others in 802.11) there are major differences in scan results across devices.

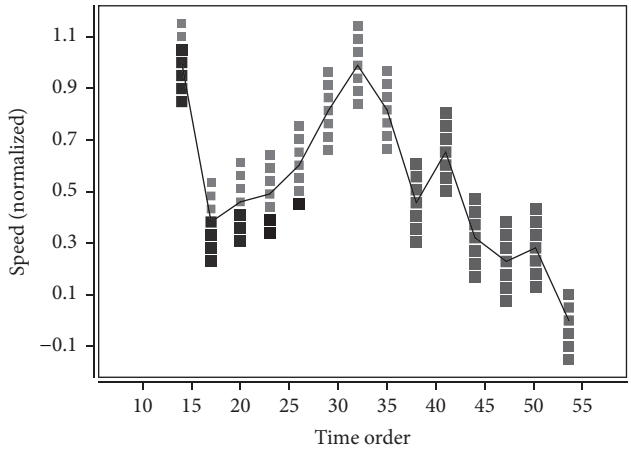

Figure 3: Context delta map for two devices separated by 10-20 meters. Delta is calculated based on relative positions of WiFi APs in signal-ordered lists.

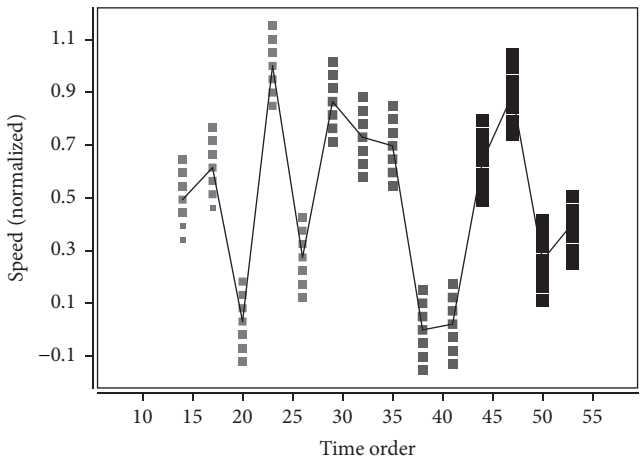

Figure 4: Context delta map for two devices separated by 10-20 meters, this time using binary match between two AP lists as delta.

4.4. Experiments with Multiple Devices. Figure 3 performs the same set of tests on various pairs of devices, separated by 10-20m distance, walking on the same path for about 100m. The same visualization approach is applied. It is immediately obvious that there is much higher delta all across the timeline. Moreover, there is no reliable pattern in vertical (lag) blocks, as well as in time. All in all, it is hard to believe that this context was generated by a pair of devices which were only 10m away from each other in an outdoors setting.

In an attempt to rectify the outcome in Figure 3, the context in Figure 4 uses binary delta that only accounts for presence or absence of APs in two compared snapshots. Another randomly sampled context is used to visualize the outcome. Yes, in some part of the context blocks are more opaque which signals a relatively small delta. But closer to the end of the timeline, there are several seconds of very high delta blocks. Again, however one tries to handle this context; it would be very hard to convert it into a hashkey that would stay reliable for that particular location.

The following conclusion on WiFi-only context can be offered. Using WiFi APs to generate a consistent hashkey which can be used as basis for PoL is not feasible at present time, especially so in dense urban spaces and when one cannot control the models of devices used by users. As a solution to the above problem, the next section proposes a pure p2p method to achieve the same goal.

5. Proposal: the P2P Proof-Of-Location

As in [8], this paper opts to make peers into Witnesses. However, a major difference from [8] is that this proposal does not depend on WiFi APs. The use of WiFi APs is not prohibited, in fact, one can imagine that some spaces would install the necessary WiFi APs that would mimic normal users and even initiate transactions. Note that, unless a single provider installs and monopolizes such infrastructure at a large scale, it does not introduce bias into consensus. Also note that it would be easy for an infrastructure node to become the Last Man Standing in a given transaction, so, some rules prohibiting this should probably be introduced. For the cloud side it is easy to distinguish such nodes with a quick look at their overall mobility.

5.1. Main Assumptions. The following are the main underlying premises for this proposal:

(i) the proposal has to solve the problem of *inconsistent hashkeys* generated by digesting local wireless context, specifically the proposal assumes that a given location should generate a consistent hashkey regardless of the rounding error in GPS coordinates, number and identities of wireless peers, timing, etc.;

(ii) this paper assumes that (blockchain) blocks are generated in real-time, i.e., taking the form of a *transaction* which has a clear start and end timestamps, records of participant peers, and other information, which, when the transaction has finished, are digested to become a (blockchain) block, and further as *coin* owned by one or multiple participant peers—this paper shows that such coins can become a valued resource and can even become part of a larger resource economy;

(iii) multiple aborted (cannot be witnessed and therefore cannot be verified) transactions are allowed but only one successful transaction can happen for a given location, resulting in a unique block describing each unique location;

(iv) this paper proposes a *distributed*, *verifiable,* and, otherwise, *blockchain-based* PoL method, which means that hashkeys are generated by digesting data which is contributed by clients at network edge via real-time local physical interactions; this means that hashkeys represented locations cannot simply be generated by cloud-side (as digests of GPS coordinates, for example); they have to be generated via a real-time transaction.

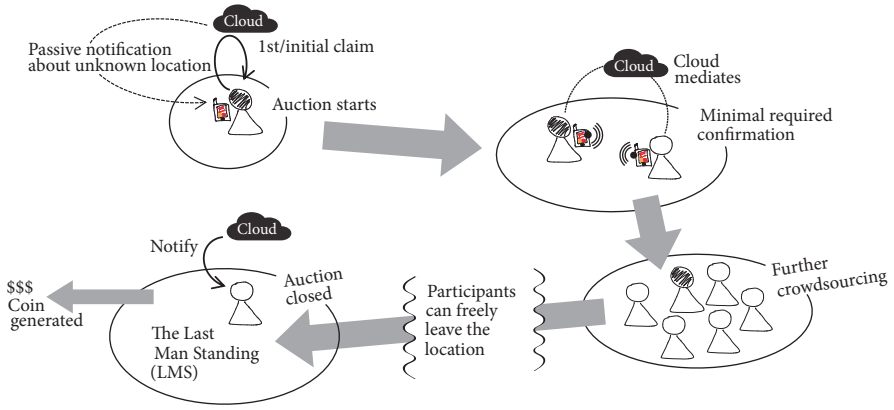

Figure 5: The sequence of actions that define the proposed Last Man Standing (LMS) form of Proof-of-Location (PoL).

The proposal assumes the following components to PoL transactions:

(i) GPS as a rough method to estimate location (rounded up longitude and latitude values) can be hierarchical by varying the rounding depth but this discussion is left for future papers on the topic;

(ii) peers are Witnesses, without at least a single Witness apart from the user that initiated the transaction; PoL is impossible; note that Witnesses naturally guarantee that GPS locations are not spoofed by malicious parties;

(iii) cloud is a mediator and also a location for public transaction ledger.

This paper focuses only on the PoL primitives and does not consider the ledger. In fact, while the block used to generate hashkeys is proposed, this paper does not go into the discussion on how such blocks can be connected into chains.

5.2. The Last Man Standing (LMS) Routine. Figure 5 explains the proposed LMS procedure. There are roughly 4 stages connected in a sequence in the figure.

Stage 1: Start of Transaction. This paper assumes that locations remain unclaimed unless a user initiates a transaction and claims (mines) it. Initiation can happen actively or passively. The most likely scenario is when a user gets a passive notification about an unclaimed location when passing through it and then chooses to act on it by placing the claim. The claimer then is forced to wait for at least one other Witness before he/she can depart; otherwise, the transaction is deemed null and is purged. One can imagine that, with low penetration of a smartphone application that implements the proposal, relatively low density locations will experience many aborted transactions, simply because no other Witness would be found within a reasonable span of time.

Stage 2: Second User, First Witness. It should be noted that at Stage 1, the first user (initiator) gets a unique ID from the cloud and sets it as its own WiFi AP SSID (it should be actively broadcasting). No one is expected to connect to this WiFi AP, but its ID can now be scanned by potential Witnesses. Similarly, the first user can now scan for other WiFi APs in the area and send the list to the cloud for verification. Matching between the lists from various users within a given transaction is a trivial task and can be quickly conducted at cloud side with minimal resource spending. Once the first Witness is confirmed—all the other users get notified on the number of Witnesses for the location and the current status—the first user can leave the area, unless he/she wants to obtain the Last Man Standing status, in which case such users should prepare for a longer wait. Note that, if both users leave the area at this point, the transaction is still successful and the user that leaves the last is declared the Last Man Standing. However, with increasing density, it is likely that the transaction advances to the next stage.

Stage 3: Other Witnesses. It is likely that more Witnesses arrive at the location, potentially being summoned by push notifications from the smartphone application. The transaction benefits from more Witnesses as this improves the consensus and makes it less susceptible to malicious attacks. There is no need for all the Witnesses to overlap in time. It is sufficient for at least pairs of Witnesses to overlap, while the cloud can figure out the timeline of encounters. Note that this forms an interesting arrival process which is already studied for online actions [22]—this discussion is offered later in this section. Regardless of the arrival process, all the confirmed Witnesses—confirmation requires that at least two users show up on each other WiFi scans—are included in the ledger for that transaction.

Stage 4: (Till) The Last Man Standing. The same as in online auctions [22, 23], arrival becomes less sparse and finally the number of active Witnesses should decline. The Last Man Standing (LMS) feature defines the point of time at which only a single Witness remains at (or departs the last from) the location. This is a natural closing point for the transaction. At this point, the coin is generated and represents the ownership of that location. Two ownership models are considered further in this section.

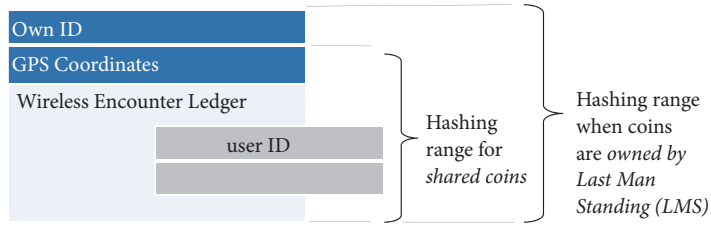

FIGURE 6: Data block which can be used to produce uniquely identifiable *hashkeys*, subject of further use a part of a blockchain-based security platform.

Note that the opposite extreme to the *aborted transaction* at Stage 1 can happen at Stage 4—this is when a location is too crowded for a transaction to end naturally. Two solutions can be offered for such situations. If coin ownership is shared—see ownership models below—the natural end of transaction can be dictated by how many decimal digits are allowed when partial coins are handled by the system. For example, for 2 decimal digits, only 100 people can share a coin.

This deadlock can be resolved even when coins are owned by the LMS person him/herself. For example, one can be required to linger in the vicinity of the location for a longer time (can be a parameter of local/current density of wireless devices) in order to be considered an LMS candidate (Witness role can still be offered to anyone without limit). This should remove majority of passing-by wireless traffic and guarantee that the transaction will end naturally at some point. Note that these solutions purposely avoid using *deadlines*—this is explained from the viewpoint of online auctions further in this section.

5.3. Similarities and Differences from Auctions. As was mentioned above, the arrival process of Witnesses at a location resembles that of bidders in online auctions. In fact, since ownership of a location (identify, hashkey, not the physical space) is a resource, the resemblance is especially strong.

Differences from online auctions are as follows. There is no clear deadline for LMS, while auctions have a clear deadline. In fact, LMS should not have deadlines as it will urge people to arrive at a location closer to the deadline, thus interfering with the LMS procedure. There is also no price competition in LMS at this point. Although, given the variable density, it is possible that either the coin value or location granularity is adjusted based on demand (future papers will look into this issue).

LMS and auctions are similar in the following ways. Both share a similar arrival process – some research on auctions looks specifically into the bid arrival process. Specifically, there is Barista model [23] which studies arrival in 3 stages. There is also a proposal to study arrival via a graph structure (nodes are users; links are bids-in-response) in which centrality parameter (two bidders competing) increases as deadline approaches [22]. Recent review of the various auction models can be found in [24]. There are also examples of auctions applied in clouds [25], which is the closest example to the application in this paper. This paper stops at specifying the connection to auction research, while future publications will look into the modeling details. It should only be mentioned that the graph-based approach in [22] is the most likely direction for such analysis, since LMS interactions can very naturally be described by a graph.

5.4. Practical Aspects of LMS. Figure 6 shows two ways of how the LMS transaction can be hashed uniquely. When a *shared coin* is assumed, ID of the initiator is left out (but is still part of the Encounter Ledger) when digesting the block into a hashkey—this way all the Witnesses can claim (partial) ownership of the same location/coin. When it is assumed that LMS him/herself gets the *entire coin*, the entire block is digested into a hashkey. The block is still verifiable using the Encounter Ledger, but the coin has only a single owner—note that it is the last Witness, not the initiator of the transaction (although the two may be the same user). Future publications will try to compare the two models both qualitatively and numerically.

Let us finally consider the various security and privacy concerns that can be raised in relation to the above proposal.

Can locations be spoofed? No. In fact, if the transaction is conducted via smartphone application, there are security measured in modern Android and other OSes that makes it nearly impossible to spoof one's location. However, to be successful, the spoofing needs to be done on multiple devices. Cloud side can easily monitor locations during transactions and raise alarms when potential malicious activity is detected. Note that, given the proposal, "spoofed location" means that the user him/herself is not at the location and will not show up on other Witnesses' scans—this can also be easily caught by cloud slide. Also note that ability to spoof diminishes naturally with increasing location density.

Can location ownership be aggregated en mass by a single party? It is not easy. Transactions require time. If a malicious user wants to participate in multiple transactions at the same time, he/she would need to quickly move from one place to another. Physical speed limit itself places a hard ceiling on the ability to aggregate transactions in such a way. Also, it is very easy for cloud side to prohibit users from participating in multiple transactions at the same time, thus making such malicious activity completely impossible.

Are transactions fair? Even if LMS-owned coins are used, fairness is retained—one has to put in much time to wait out other competitors, which is a price one pays for future ownership. With shared coins, LMS itself becomes a formality where the cloud side needs to close the transaction, while

Witnesses can leave immediately once an encounter with other Witnesses is confirmed. In both cases, transactions are fair to all participants.

Is there sufficient incentive for participation? Well, it depends on what one can do with the coin. Since there are few examples of cryptocurrency attached to physical resources (there are small but mostly unknown cases in farming, for example), this is terra incognita for future applications. This author can imagine cases when such crypto-ownership of locations can be used to pay micro-royalty to crypto-owners when other cloud services request verifiable proof for that location. The deeper discussion of this topic is left for future publications.

Is privacy ensured? Yes. IDs for each Witness are generated randomly by cloud side. While the cloud side does have access to the complete information, Witnesses retain privacy among each other during the transaction. With some effort, Witness IDs can also be anonymized so that cloud side would not be able to match IDs to identities. Again, this topic is too large for this publication.

6. Trace-Based Analysis

Analysis of the potential of the proposal is done on real mobility traces publically available at [10]. Additional data structures generated on top of the raw traces—such as density maps and Lucene databases—are offered at [11]. Since the latter has both the raw and additional data, using only [11] is sufficient.

The traces at [10] come from various locations including university campus (KAIST), a state fair at a stadium (given the shape of mobility traces), urban area (New York), etc. Such a diversity is helpful when analyzing the potential for crowdsourcing location proofing to end users. The traces themselves come in form of GPS values with timestamps captured at round 30s intervals.

Such traces have to be processed from the user-centric form into a location-centric form to be useful for analysis in this paper. Figure 7 is the first step in this direction. It is a density map for all the locations in the KAIST trace (university campus). Locations are grid cells of 50x50 meters. Distances are identified directly using the GPS coordinates in the traces (those are real/actual values). However, the time component is removed from the map, which means that locations do not count people that necessarily overlap in time. Note that the proposal in this paper does not require the strict time overlap and is instead satisfied with pairwise encounters.

Naturally, (relatively) denser locations are marked in darker tone in the map. If one searches for the map of KAIST online, one can see that the density map not only traces the contours of the campus but also mirrors the obviously dense locations—those around major buildings.

Figure 8 uses density maps for all the traces at [10] and infers how the population of 10k users/devices is distributed across the various 50x50m locations within each respective area. The two extremes in Figure 8 are KAIST at one end and Statefair at the other. KAIST represents a large area with many potential locations and, thus, a wide range of density values. Statefair trace, on the other hand, comes from a small area and has 8 times fewer grid cells. However, regardless of the fewer potential locations, distribution of the Statefair trace is much steeper, with only about 20% of the locations falling below 20 people. The KAIST trace, by comparison, is much flatter and, therefore, allocates fewer people to each location.

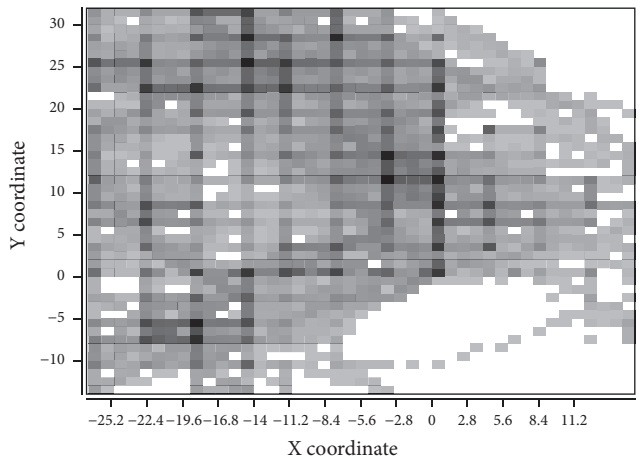

Figure 7: Density map created based on the KAIST trace from [10]. X and Y coordinates are rounded to the closest 50 meters, where the coordinates are shown as multiples of this unit and are relative to the geographical (GPS) center of the campus.

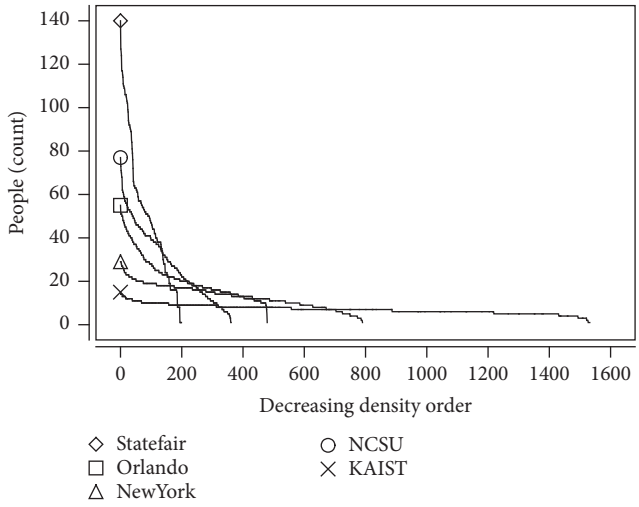

Figure 8: Comparing all the traces in the [10] dataset in terms of the distribution of density, calculated as the number of people relative to the total 10k population.

Figure 9 attempts to infer the case when notifications are sent to users who are not physically at the location but can decide to walk a small distance to participate in a PoL transaction. Only the New York trace is used—the actual values are represented by the baseline distribution. We can now clearly see that New York trace has 800 locations with only the last 100–150 locations falling below 10 people per location.

Now, let us see what happens when notifications are sent to all users within 100m range from each location. First, the

Table 1: Comparison with three main rival existing methods.

	PPLV [6]	ADGT [12]	CLIP [7]	LmsPoL (this proposal)
Requires new WiFi or other infra?	Yes	No	Yes	*No*
Unicast versus broadcast comm. (mostly the same as active vs. passive)	unicast	Unicast	Unicast	*Broadcast*
Based on blockchain?	No	Yes	Yes	*Yes*
Defines coin (or ownership)?	No	No	No (unit: trace/path)	*Yes*
Strong consensus? (protected from spoof attacks)	- (trusted infra)	Limited (sale for chain monopoly)	Yes	*Yes*
Privacy for clients/participants?	Limited privacy	Yes	Yes	*Limited (cloud-side knows)*
Fairness	Yes	No (resources = advantage)	Yes	*Yes*
Participation incentive	No	No	No	*Yes*

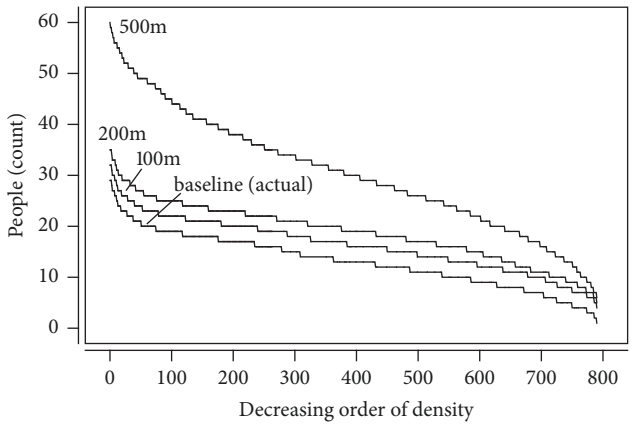

Figure 9: Visualizing how p2p crowds can grow when notifications are sent to devices within 100m, 200m, and 500m radius from each location. Only the NewYork trace is used—shown as *baseline* in the plot.

tail of the curve gets a substantial raise—this is because some low-density areas have high-density areas nearby which are to blame for the boost. The head of distribution gets a smaller yet a tangible boost. Moving from 100m to 200m range provides only a small further surplus. Finally, when notifications are sent to users within 500m area, the distribution experiences a major uplift—this is because, at this range, dense areas provide boosts for each other as well as all low-density areas in between. Considering 500m performance of an extreme case, it might be beneficial to set the range at between 100m and 300m to provide additional support for transactions but not overtax them with large crowds. Naturally, cloud side should maintain a separate frame of reference for each large city (or part of city) and vary the range for push notification, if used at all.

7. Comparison with Existing Methods

The three methods in recent literature which can be considered as rivals to this proposal are PPLV [6], ADGT [12], and CLIP [7]. Descriptions of these methods and the relation to this proposal were offered earlier in this paper. This section focuses on specific features which can be used for comparison.

First, let us establish the dimensions which are important for operation of a PoL method. Table 1 shows all the comparison analysis in a single spreadsheet, while the text below goes through the table feature-by-feature.

WiFi infrastructure upgrade is often mentioned in literature indirectly, but proposing message exchanges between clients and WiFi APs which lie beyond capabilities of conventional infrastructure. Often referred to as *smart WiFi APs*, they pose a problem since they require a major investment into the upgrade and may cause further complications when problems with compatibility and legacy support arise in the future. This proposal and ADGT work with the conventional infrastructure while the other two methods in Table 1 require smart WiFi APs.

Unicast vs. broadcast messaging, which can also be referred to as *passive vs. active*, since unicast messaging require active involvement, while broadcasts can be listened to passively. This proposal is the only method of the four in Table 1 which is based on the passive broadcasting.

Three of the four compared methods are based on the *blockchain technology*, while PPLV [6] depends on trusted WiFi APs. Note that the problem with a nonblockchain technology is that it cannot be verified outside of a centralized authority (mint) in a reliable way.

Whether a method *assigned as coin or ownership* for PoL transaction participants is important. Only this proposal makes this feature a key element of the proposal, where locations are linked to blocks and further to coins with a clearly defined ownership (can be private).

Strong consensus is important when judging how a given method can be attacked. Specifically to PoL, location spoofing is the immediate concern. In this respect, this proposal is at about the same level as the other three methods.

The same goes for *privacy* where this proposal offers a limited (but controlled) form of privacy (the cloud side knows all the identities), the same as some of the other compared methods.

Fairness is an important factor. For example, ADGT [12] gives the ownership of a hashkey to a user who submits the most number of Witness accounts (also p2p, like this proposal). This means that hardware or, broadly, resource advantage can help one collect more Witness accounts and thus win the ownership of the hashkey. The other two compared methods can be considered fair, meaning that all clients are equal regardless of available resources, but they also do not discuss ownership as part of their proposal.

The final aspect where this paper is unique is the *participation incentive*, where this proposal is the only one that considers this aspect. Setting PPLV aside as a nonblockchain method, ADGT and CLIP both expect local collaboration from peers who become witnesses during the transaction. However, both AGDT and CLIP fail to specify the incentives which would make such a collaboration feasible in practice. By contrast, this paper defines the ownership of the coin and even allows for shared ownership which is a sufficient incentive for peers to participate in a given local transaction. Quite literally, this paper draws a scenario in which people would walk a certain distance to a given unclaimed location in order to *invest* (their effort) in a coin which can become as source of income in the future when Location-Based Services at that location are monetized.

Table 1 summarizes the discussion in this section in a table form for easy visual comparison.

8. Conclusion

The starting point for this paper is placed at recent research on Proof-of-Location techniques, where the common feature is the assumption that WiFi APs play an active role in helping each user verifies the location. While there are also attempts to encode user-side mobility and WiFi scan results as a spatio-temporal trust structure which can be witnesses by other users (or WiFi APs) and independently verified, in the end, location *hashkeys* come from WiFi APs in a solid, precalculated form.

This paper started with the assumption that commodity WiFi APs are used to encode local wireless context. Unfortunately, even leaving mobility context (comes from own accelerometer) out of scope, this paper showed that one cannot encode WiFi scan results in such a way that they can be successfully *witnessed* by other nearby users. The problem lies in too much variation in both the signal level and presence of WiFi APs in scan results. Context hashing failed even for a single device when a trace was compared to a time-shifted version of itself. For multiple devices, the reliability of context hashing rapidly degrades further.

Having learned from this experience, this paper decided to employ the purely p2p version of the method. In it, peers, assisted by cloud side, broadcast their own unique IDs via their own WiFi APs (smartphones are assumed) and collect WiFi scan results with IDs of potential Witnesses which are within the WiFi range. The *WiFi range* is a natural way to confirm that users are at the same location. Note that WiFi range can also help deal with the problem of having unreliable GPS readings indoors—for example, a separate floor in the same building may share the same GPS coordinates but otherwise have a distinct and separate transaction from those on other floors.

The cloud side collects WiFi scans from all the users within the same transaction and gradually builds the transaction ledger to be later converted into a cryptocurrency coin. This paper proposes the Last Man Standing (LMS) procedure which helps transactions end in a natural way—with the departure of the last device. However, coins generated in the aftermath may be shared by all the participants in the transactions whose IDs are stored in the Encounter Ledger and maintained by cloud side. If the importance of LMS needs to be emphasized, the ownership of the coin can be assigned to the LMS him/herself. This paper does not favor either of the models; instead it was satisfied with providing detailed descriptions for both.

The proposal is analyzed both qualitatively and quantitatively. In terms of quality, the various security and privacy concerned are raised and dutifully answered. In terms of quantity, analysis was performed using real mobility traces captured at various types of locations, showing that, while density can vary broadly within the location, locations at the tail of distributions can still satisfy the necessary conditions for the proposal—having at least two users to witness each other's presence at the location. As a special case, this paper shows how the number of people can be boosted at the tail by sending push notifications within a limited radius from a given location. Note that the proposal has a natural fallback for the case when the first Witness (second user) does not show up for a given transaction—in this case the transaction is aborted and the location is deemed unclaimed subject for new claims in the future. Using this logic, with time, all the locations within a large urban area should gradually be claimed.

Several interesting subjects are left out of scope of this paper. For example, it was shown that the arrival process during transaction is similar to that found in existing literature on online auctions. Specifically, graph representation of the arrival and departure processes is an interesting venue for future research on the topic. Future work will also be conducted on connecting the individual blocks/coins in chains/trees, finally growing in scale to entire cities. Such research will draw parallels from literature on blockchain technology but should offer interesting features coming from the geographical nature of the considered resource. In fact, the link between cryptocurrency and physical resources is, but itself, an interesting subject for future study.

Data Availability

The data is made publically available.

Conflicts of Interest

The author declares that they have no conflicts of interest.

References

[1] O. Hersent, D. Boswarthick, and O. Elloumi, *The Internet of Things: Applications to the Smart Grid and Building Automation*, John Wiley & Sons, Ltd, 2012.

[2] J. Jin, J. Gubbi, S. Marusic, and M. Palaniswami, "An information framework for creating a smart city through internet of things," *IEEE Internet of Things Journal*, vol. 1, no. 2, pp. 112–121, 2014.

[3] R. A. Michelin, A. Dorri, and A. F. Zorzo, "SpeedyChain: A framework for decoupling data from blockchain for smart cities," *Computing Research Repository (CoRR)*, 2018, https://arxiv.org/abs/1807.01980.

[4] A. Kiayias, H.-S. Zhou, and V. Zikas, "Fair and robust multiparty computation using a global transaction ledger," in *Annual International Conference on the Theory and Applications of Cryptographic Techniques (EUROCRYPT)*, pp. 705–734, 2016.

[5] R. Lunardi, R. Michelin, C. Neu, and A. Zorzo, "Distributed access control on IoT ledger-based architecture," in *Proceedings of the 2018 IEEE/IFIP Network Operations and Management Symposium, NOMS*, 2018.

[6] G. Brambilla, M. Amoretti, and F. Zanichelli, "Using block chain for peer-to-peer proof-of-location," *Computing Research Repository (CoRR)*, 2016, http://arxiv.org/abs/1607.00174.

[7] Y. Li, L. Zhou, H. Zhu, and L. Sun, "Privacy-preserving location proof for securing large-scale database-driven cognitive radio networks," *IEEE Internet of Things Journal*, vol. 3, no. 4, pp. 563–571, 2016.

[8] C. Lyu, A. Pande, X. Wang, J. Zhu, D. Gu, and P. Mohapatra, "CLIP: Continuous location integrity and provenance for mobile phones," in *Proceedings of the 12th IEEE International Conference on Mobile Ad Hoc and Sensor Systems, MASS 2015*, pp. 172–180, USA, October 2015.

[9] M. Zhanikeev, "Opportunistic multiconnect with P2P WiFi and cellular providers," in *Mobile Computing and Communications: 4G and Beyond*, pp. 271–318, CRC Press, Boca Raton, Fla, USA, 2016.

[10] "CRAWDAD Repository of Mobility Traces," http://crawdad.cs.dartmouth.edu, 2014.

[11] "LmsPoL Project page with wireless context and other mobility traces," https://github.com/maratishe/lmspol, 2019.

[12] S. Nakamoto, "Bitcoin: a peer-to-peer electronic cash system," http://bitcoin.org/bitcoin.pdf, 2009.

[13] R. Chandra, J. Padhye, L. Ravindranath, and A. Wolman, "Beacon-stuffing: Wi-Fi without associations," in *Proceedings of the 8th IEEE Workshop on Mobile Computing Systems and Applications (HotMobile)*, pp. 53–57, 2007.

[14] P. Sharma, S. Moon, and J. Park, "Block-VN: a distributed blockchain based vehicular network architecture in smart city," *Journal of Information Processing Systems (JiPS)*, vol. 13, no. 1, pp. 184–195, 2017.

[15] D. Niculescu and B. Nath, "VOR base stations for indoor 802.11 positioning," in *Proceedings of the the 10th Annual International Conference on Mobile Computing and Networking (MobiCom)*, pp. 58–69, 2004.

[16] E. Soltanaghaei, A. Kalyanaraman, and K. Whitehouse, "Multipath triangulation: Decimeter-level WiFi localization and orientation with a single unaided receiver," in *Proceedings of the 16th ACM International Conference on Mobile Systems, Applications, and Services (MobiSys)*, pp. 376–388, 2018.

[17] V. Erdélyi, T.-K. Le, B. Bhattacharjee, P. Druschel, and N. Ono, "Sonoloc: Scalable positioning of commodity mobile devices," in *Proceedings of the 16th ACM International Conference on Mobile Systems, Applications, and Services,MobiSys 2018*, pp. 136–149, 2018.

[18] H. Tran, S. Pandey, and N. Bulusu, "Online map matching for passive indoor positioning systems," in *Proceedings of the 15th ACM International Conference on Mobile Systems, Applications, and Services (MobiSys)*, 2017.

[19] A. Virmani and M. Shahzad, "Position and orientation agnostic gesture recognition using WiFi," in *Proceedings of the 15th ACM International Conference on Mobile Systems, Applications, ad Services (MobiSys)*, pp. 252–264, 2017.

[20] A. Popleteev, "Impact of ground truth errors on Wi-Fi localization accuracy," in *Proceedings of the 15th ACM International Conference on Mobile Systems, Applications, and Services (MobiSys)*, 2017.

[21] G. Shmueli and W. Jank, *Modeling Online Auctions*, John Wiley & Sons, 2010.

[22] M. Dass, S. K. Reddy, and D. Iacobucci, "A network bidder behavior model in online auctions: a case of fine art auctions," *Journal of Retailing*, vol. 90, no. 4, pp. 445–462, 2014.

[23] G. Shmueli, R. P. Russo, and W. Jank, "The barista: a model for bid arrivals in online auctions," *The Annals of Applied Statistics*, vol. 1, no. 2, pp. 412–441, 2007.

[24] R. Vadovič, "Bidding behavior and price search in Internet auctions," *International Journal of Industrial Organization*, vol. 54, pp. 125–147, 2017.

[25] X. Zhang, Z. Huang, C. Wu, Z. Li, and F. Lau, "Online auctions in IaaS clouds: welfare and profit maximization with server costs," *IEEE/ACM Transactions on Networking*, vol. 25, no. 2, pp. 1034–1047, 2017.