Decibel.LIVE: A Decentralized Noise Pollution Monitoring and Incentive Platform *

https://www.decibel.live

Vijay Kandy and Shane Loomb

{vijay,shane}@decibel.live

March 25, 2017

ABSTRACT

Noise pollution is one of the leading causes which adversely affect the well-being and quality of life of residents in urban areas. In this paper we present Decibel.LIVE, a blockchain based decentralized platform to monitor urban noise and compensate residents living around the noise sites. With the Decibel.LIVE smartphone app, citizens can measure their personal exposure to noise using smartphones as noise sensors and report excess noise to business owners responsible for the noise sites over private encrypted channels. The platform also implements a settlements infrastructure based on blockchain technology which enables business owners to compensate residents affected by noise in real-time. The platform aggregates noise levels in a given region and presents visual noise propagation models which can be used by communities and local noise enforcement officers to enforce noise guidelines.

1. INTRODUCTION

Noise is often described as *unwanted sound* which is unpleasant and disruptive to hearing. It affects quality of life because it interferes with activities like sleep, relaxation, conversation or listening to music, TV, radio or natural sounds.

In recent decades, increases in population have also led to increases in traffic, construction, demolition and other noise Noise pollution is a common problem in urban environments, affecting human behavior, health and even children's cognitive development.

Typically, cities and towns have established guidelines on noise levels allowed in residential areas. Local laws and bylaws may restrict general construction and other sources of noise depending on the neighborhood, commercial zones, time of day and days of week such as weekends and holidays. Cities may employ noise abatement inspectors to enforce noise guidelines. However, proving noise variance when code violations occur isn't easy with existing methods. Frustrated residents end up calling local bylaw enforcement or noise control officers. However, without any proof of code violation there's not much a noise control officer can do.

There is both objective and subjective evidence for the evidence of non-auditory effects of noise on health and well-being which are caused by exposure to noise [13]. Exposure to levels above 40 dB(A) begins to influence well being and levels above 60 dB(A) are considered detrimental to health. There is good evidence, largely from laboratory studies, that noise exposure impairs performance [8]. Noise exposure during sleep may increase blood pressure, heart rate and finger pulse amplitude as well as body movements. There may also be after-effects during the day following disturbed sleep; perceived sleep quality, mood and performance in terms of reaction time are all decreased following sleep disturbed by noise [14].

In this paper we present Decibel.LIVE, a novel approach to noise pollution monitoring using blockchain technology and smartphones. Leveraging the decentralized nature of the blockchain and the continuous climb in smartphone ownership and internet usage, we present an open and low-cost platform to monitor noise pollution.

2. NOISE ASSESSMENT

Lisez Euler, lisez Euler, c'est notre matre tous.

— Pierre-Simon Laplace

This section summarizes noise measurement techniques particularly environment noise. We review noise measurement techniques and tools to capture noise data.

Sound or noise is the result of pressure variations, or oscillations, in an elastic medium (e.g., air, water, solids), generated by a vibrating surface. Sound propagates in the form of longitudinal waves, involving a succession of compressions and rarefactions in the elastic medium [6]. *Appendix A* gives a glossary of acoustic terminology.

As sound waves travel outward from the source, their intensity (loudness) expressed as watts per square meter (W/m^2) steadily decreases due to several natural phenomena such as spreading with distance, absorption by the atmosphere and wind and temperature changes. In a free field, the intensity and sound pressure at a given point, at a distance r (in meters) from the source, is expressed by [6]:

*This is a draft release. A final version will be published when the application is deployed to Ethereum mainnet.

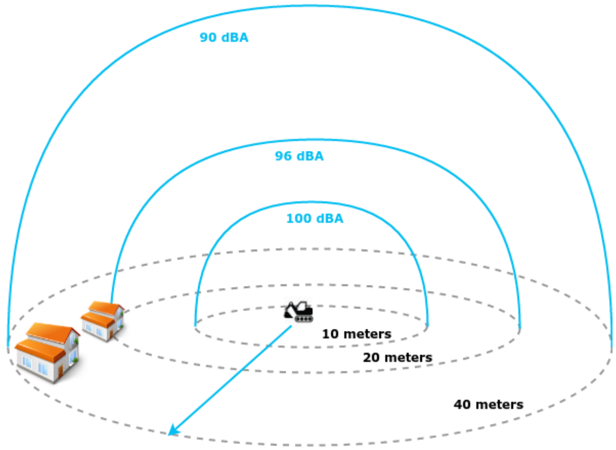

Figure 1: Attenuation rate at 6dBA per doubling of the distance.

$$p^2 = \rho c I = \frac{\rho c W}{4\pi r^2} \quad (1)$$

where ρ and c are the air density and speed of sound respectively.

From the preceding equation, if the sound pressure level, L_{m}, is measured at some reference distance, r_{m}, from the noise source then the sound pressure level at some other distance, r, may be estimated using [1]:

$$L_{\mathrm{p}} = L_{\mathrm{m}} - 20\log_{10}(\frac{r}{r_{\mathrm{m}}}) \quad (2)$$

It follows that in free field conditions, the noise level decreases by 6 dB each time the distance between the source and the observer doubles (See Figure 1).

However, free-field conditions don't exist in urban environments so, in general the equation relating sound pressure level and sound power level is modified to account for the presence of reflecting surfaces.

In the near field where distance from the source is equal to about a wavelength of sound or equal to three times the largest dimension of the sound source (whichever is the larger), the sound field does not decrease by 6 dB each time the distance from the source is increased (as it does in the far field).

Our focus is the far field of a source which begins where the near field ends and extends to infinity. This is the field where the citizens affected by noise pollution reside. Note that the transition from near to far field is gradual in the transition region. In the far field, the direct field radiated by most machinery sources will decay at the rate of 6 dB each time the distance from the source is doubled. For line sources such as traffic noise, the decay rate varies between 3 dB and 4 dB [6].

2.1 Measuring environmental noise dB(A)

The basic unit of measurement for environmental noise, as recommended by ISO 1996/1, is the dB(A). Noise emission limits are also usually expressed in dB(A). Noise levels may be measured at any moment, but noise levels vary widely with time, e.g., traffic noise is higher at noon than at midnight and higher still during the morning and evening rush hours. Therefore a single dB(A) measurement isnt a very good indicator of noise pollution. A much better indication is a logarithmic average, $L_{\mathrm{Aeq,T}}$ which is the metric of choice for measuring exposure to noise.

$L_{\mathrm{Aeq,T}}$ is the equivalent continuous dB(A) level which has the same energy as the original fluctuating noise for the same given period of time T and is defined as [12]:

$$L_{\mathrm{Aeq,T}} = 10\log_{10}(\frac{1}{T}\int_0^T \frac{p_A(t)^2}{p_0^2}\,\mathrm{d}t) \quad (3)$$

where $p_A(t)$ is obtained from applying A-weighting filter to the raw acoustic signal $p(t)$ and p_0 is a standard reference value corresponding to the minimal audible acoustic signal ($20\mu Pa$ in air). The raw acoustic signal $p(t)$ is filtered (A-weighting) to compute the relative loudness of sound as perceived by the human ear. A weighting is (formally defined standards IEC 61672-1 and ANSI 1.43) in the analog or frequency domain and converted to a time domain digital filter, normally done with z-transform or bilinear transform.

Zero dB(A) is considered the point at which a person begins to hear sound[3]. A soft whisper at 1 meter measures at 30 dB(A), a busy freeway at 15 meters at around 80 dB(A), and a chain saw can reach 110 dB(A) or more at operating distance. Brief exposure to noise levels exceeding 120 dB(A) without hearing protection may even cause physical pain.

Decibel.LIVE smartphone app implements filtering to produce dB(A) events.

2.2 Techniques for capturing noise levels

Techniques to monitor noise in urban areas vary based on jurisdictions. Typically, noise measurements in urban areas are carried out by designated officers or consultants who collect data at a particular location using a sound level meter or similar microphone-equipped device. However, the measuring sessions take place only at few accessible spots and during short time intervals (e.g., 30 minutes).

There are a number of limitations with such approaches. Noise mapping campaigns are expensive due to need of expertise and human resources, the deployment of sound level meter devices and the processing effort. Also, data collection at sparse locations and at specific times often produce computational models with an unknown error margins.

2.2.1 Alternative Approaches

Over the last few years, smartphone adoption has been rising worldwide[10]. Modern smartphones have significant computational power, always-on internet connectivity and built-in sensors such as microphones, cameras, global positioning system (GPS) receiver, accelerometers, gyroscopes, and proximity and light sensors making them as sensors carried by humans rather than placed at static locations. Smartphone developers now offer many sound measurement applications that are accurate and reliable to be used to assess occupational noise exposures[7]. This has given rise to a low-cost alternative to traditional large-scale, costly and difficult to manage sensing infrastructure based on sensor networks.

One such novel implementation is NoiseTube[9] which is a mobile application platform that monitors noise pollution involving the general public. Citizens measure their personal exposure to noise in their everyday environment by using GPS-equipped mobile phones as noise sensors. VANET[11] takes a different approach by using a mesh network composed of mobile phones for detecting noise and transporting noise data. These projects show that smartphones can be used effectively as noise sensors to monitor noise pollution.

3. ETHEREUM BLOCKCHAIN

> *TCP implementations will follow a general principle of robustness: be conservative in what you do, be liberal in what you accept from others.*
>
> — Jon Postel, *RFC: 793, Transmission Control Protocol*

In the strictest sense, a blockchain [1] is a type of database that is shared across nodes geographically distributed in a network. No single entity owns the data on a blockchain and the design of a blockchain makes it infeasible to modify data once it's published on a public blockchain like Ethereum.

We have chosen Ethereum blockchain as the platform for Decibel.LIVE service.

3.1 Why Ethereum?

For Decibel.LIVE, the primary use of blockchain is as a shared public database. We want any user of Decibel.LIVE to be able to submit a `transaction` to the blockchain. The `transaction` propagates throughout the network and is eventually committed into a consistent ordered ledger.

Any party can view all the transactions committed on the blockchain. If any party's view includes a transaction `tx`, then every party's view will also include `tx` after a maximum time bound. Therefore we need a time bound, Δ, which bounds the maximum delay round trip time for a blockchain transaction such that if a party P_1 submits a transaction `tx` at time `T`, then all parties have the same view of confirmed transaction `tx` within time $\texttt{T} + \Delta$.

Our choice of Ethereum as blockchain platform is due to the following:

1) Ethereum [15] is one of the biggest public blockchains with over 6,000 full nodes running as of this writing [5].

2) Blocks are confirmed in a few seconds, typically around 15-17 seconds [2].

3) Ethereum also provides a Turing-complete language called Solidity which is used to write *smart contracts*.

Smart contracts are programs which consist of data declarations and high-level functions. Users interact with a smart contract by publishing a transaction containing a procedure call, including the address of the contract, the name of the function to invoke, and arguments to pass.

A user submits a transaction `tx` by paying a *gas* fee. The gas price is the cost of executing the smart contract code on Ethereum blockchain.

3.2 State Channels

While blockchains as distributed ledgers have proven to be successful at low transaction volumes, scalability is a concern. As blocks are added to Ethereum blockchain every 12 seconds on average, the Ethereum transaction rate is slower than that of a typical centralized database.

Also, storing every transaction in a smart contract can be expensive due to the gas price.

State channels are a promising approach[2] addressing the scaling problem by moving some of the transactions *off chain*, while preserving the systems decentralized structure and strong integrity guarantees.

Instead of submitting every transaction to the blockchain, state channels employ the blockchain to first establish a shared deposit between two parties. The parties interact directly and communicate with the blockchain only to close channels or to resolve disputes between the parties. Therefore state channels dramatically increase the transaction rate while preserving all the benefits of the blockchain.

A state channel $\chi^{P_1 \to P_2}$ between two parties P_1 and P_2 is created on chain using a hashed time locked contract $\mathcal{C}_{Preimage}$ which acts as an escrow. A channel is initialized when party P_1 creates a conditional *commitment* (a micropayment or a bid etc.) by sending a hash h to preimage contract $\mathcal{C}_{Preimage}$ such that $h = \mathcal{H}(\sigma)$ is the hash of a secret string σ only known to party P_1. The channel is valid until time T_{Expiry}.

The state of the channel is concluded in one of three ways:

1) When party P_1 closes the channel by sending a final *commitment* and publishing the preimage σ to $\mathcal{C}_{Preimage}$.

2) When party P_2 closes the channel by publishing the latest *commitment* it received to $\mathcal{C}_{Preimage}$.

3) Or in case of a dispute when one or both parties are not in agreement of the latest state of the commitment. Typically parties establish dispute resolution policies and may employ a arbiter contract.

We denote a state channel between two parties P_1 and P_2 as

$$P_1 \xrightarrow[\mathcal{C}_{Preimage}[h, T_{Expiry}]]{commitment} P_2 \tag{4}$$

where party P_1 makes a commitment to party P_2 an action denoted by *commitment* (a micropayment or a bid etc.) by publishing the preimage σ of hash $h = \mathcal{H}(\sigma)$ before time T_{Expiry}.

4. DECIBEL.LIVE ARCHITECTURE

> *I often say that when you can measure what you are speaking about, and express it in numbers, you know something about it; but when you cannot measure it, when you cannot express it in numbers, your knowledge is of a meagre and unsatisfactory kind; it may be the beginning of knowledge, but you have scarcely, in your thoughts, advanced to the stage of science, whatever the matter may be.*
>
> — Lord Kelvin, *Electrical Units of Measurement*

Decibel.LIVE is a synthesis of ideas presented in the previous two sections. The platform consists of two distinct components:

1) **Smartphone app** to capture noise levels and store compensation transactions with the built-in wallet.

2) **DApp** (short for decentralized application) consists of:

[1] The terms *blockchain* and *distributed ledger* are often treated as synonyms in the industry even though blockchain is a specific type of distributed ledger

[2] https://blog.ethereum.org/2015/09/14/on-slow-and-fast-block-times/

Table 1: Noise Collection Fields

Field	Type	Description
beneficiaryId	string	represents a resident
lat	decimal	latitude
long	decimal	longitude
db	integer	db(A) level
timestamp	integer	UNIX timestamp

a) a smart contract $\mathcal{C}_{DecibelLive}$. [3]

b) *Bel* (ß), an ERC20 token built on top of Ethereum.

c) an API to interact with the smart contract on the blockchain.

d) a mesh routing network to link and manage secure off-chain channels.

4.1 Smartphone app

The Decibel.LIVE app is available for free and can be installed on GPS equipped smartphones. A `beneficiary` uses Decibel.LIVE app to monitor noise and send noise events in JSON format to the `initiator` over the secure Telehash channel.

4.1.1 *Gathering Noise Data*

The Decibel.LIVE app collects five fields as shown in table 1 including *beneficiaryId* representing a `beneficiary`, geolocation (*lat*, *long*) where noise sample was captured, the noise level in dB(A) (*db*), and the time when the noise was captured (*timestamp*). No additional information is captured.

The app measures the loudness level of the captured environmental sound at 22500 Hz, 16 bits/second at a chosen interval. The app contains a real-time signal processing module which applies infinite impulse response (IIR) and computes $L_{\mathrm{Aeq,T}}$ in dB(A) as the primary indicator of noise pollution. IIR filters are used as they tend to mimic the performance of traditional analog filters and make use of feedback.

The app also takes into account two common noise indicators, L_{night} and L_{den} which represent the equivalent sound levels averaged over the night only (to assess sleep disturbance) and over the whole day, respectively.

These values on the A-weighted dB scale are used in the payout settlement algorithm (see Algorithm 1).

4.1.2 *Storing Noise Data*

Noise events collected on all secure channels are stored in a local database and persisted on IPFS[4] periodically. Events are organized by noise contract and beneficiary:

```
/DecibelLIVE/{:deedId}/events/{:beneficiaryId}
```

At any time, payout settlement can be verified using the noise events and policies in any noise contract $\mathcal{C}_{Noise}$ even after the contract becomes inactive.

4.2 The DApp

[3]Currently the contract is deployed on Ropsten Testnet at: 0XBD29DD4534F2DD0A8B5F4AA366A02451795FF09D

[4]https://ipfs.io/

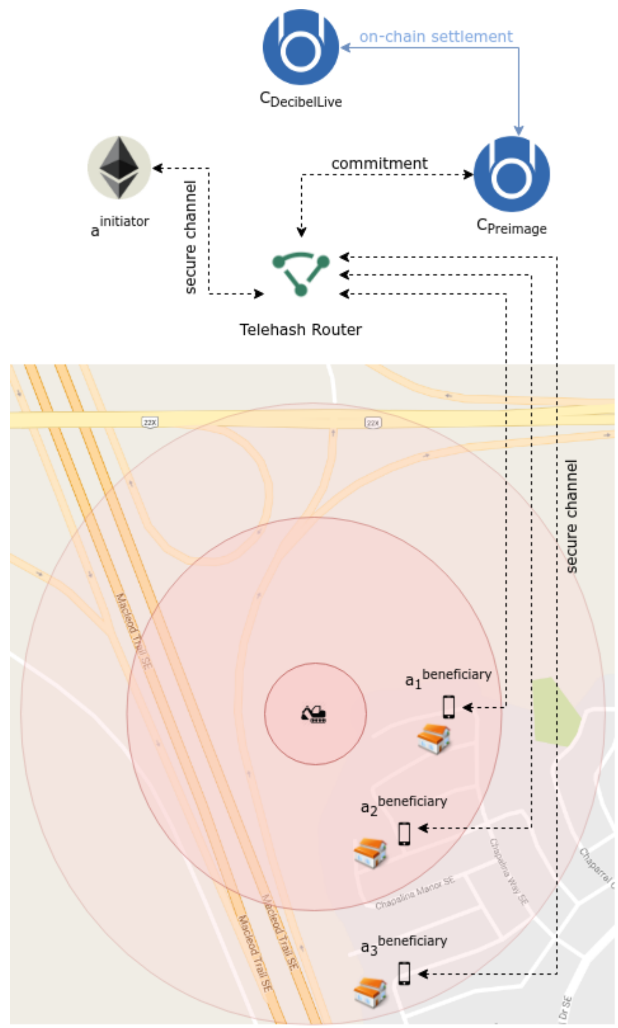

Figure 2: Mesh network of secure channels connecting initiator, beneficiaries, and Preimage contract.

The dApp consists of a smart contract which we denote here as $\mathcal{C}_{DecibelLive}$ and a web based API[5] to provide read and write access to the contract.

There are four types of identities in the Decibel.LIVE platform that interact with $\mathcal{C}_{DecibelLive}$ contract – *Initiator*, *Beneficiary*, *Noise Smart Contract* and *Contract Monitor*.

4.2.1 *Initiator*

An `initiator` is an individual or business responsible for the noise site. Every initiator is assigned a *wallet* within the Decibel.LIVE platform. The wallet consists of a secret-key sk, a public-key pk derived from the secret key, and unique Ethereum address $a^{initiator} = (pk, sk)$ constructed from the public-key.

An `initiator` creates a smart contract for monitoring noise at a noise site.

4.2.2 *Beneficiary*

[5]The live site is at https://www.decibel.live and connects to Ethereum Mainnet. The test site is at https://test.decibel.live and connects to Ethereum Testnet

A `beneficiary` is any party (i.e., residents, other businesses) affected by noise originating from `initiator`'s noise site. Every beneficiary is also assigned a wallet consisting a secret-key sk', a public-key pk' derived from the secret key, and unique Ethereum address $a^{beneficiary} = (pk', sk')$ constructed from the public-key.

4.2.3 *Noise Smart Contract*

$\mathcal{C}_{DecibelLive}$ creates and deploys new noise smart contracts on behalf of initiators. A noise contract which we denote as $\mathcal{C}_{Noise}$ encapsulates information about the noise site such as the location (latitude and longitude coordinates), Ethereum addresses of N beneficiaries $a_1^{beneficiary}...a_N^{beneficiary}$ confirmed by Decibel.LIVE as residents living within an area of radius r meters from the noise site. Following are the attributes of a noise contract.

```
// Reference to Decibel.LIVE system
address decibel;

// Defines a location reference. Location
// coordinates are stored multiplying by
// 10^6 (i.e., coordinates have 6 digits
// after the decimal point)
int32 lat;
int32 long;

// contract valid from (Unix timestamp)
uint8 beginsAt;

// contract valid till (Unix timestamp)
uint8 endsAt;

// Refers to the initiator
InitiatingParty initiatingParty;

// contract id
bytes32 deedId;

// A dictionary of beneficiaries with
// beneficiaryIds as keys
mapping (bytes32 => BeneficiaryParty)
  public beneficiaryParties;

// Store keys separately
bytes32[] beneficiaryIds;

// Current state of the contract
State public state;

// payout policies per currency and base unit
// set in the deed
PayoutPolicy[] payoutPolicies;
```

An `initiator` creates a noise contract by filling out a contract template from the web or mobile application. The platform uses an implementation of smart contract templates[4] to create and deploy noise contracts. In essence, $\mathcal{C}_{DecibelLive}$ uses a creational pattern and acts as a factory to create new instances of $\mathcal{C}_{Noise}$ on behalf of an `initiator`. Following method in $\mathcal{C}_{DecibelLive}$ creates a new noise contract.

```
function newDeed(
    bytes32 _deedId,
    address _initiatingParty,
    Holding.Currency _currency,
    uint256 _initialBalance,
    int32 _lat,
    int32 _long
)
    public
    onlyDecibel
    returns(address)
{
    Deed deed = new Deed(_deedId, _initiatingParty,
    _currency, _initialBalance, _lat, _long);
     deeds[_deedId] = deed;
    deedIds.length++;
    deedIds[deedIds.length - 1] = _deedId;
    return (deed);
}
```

Table 2: Example payout policies

Day	During	Threshold dB(A)	Payout
Sat	6AM - 5PM	60 - 70	ß1/min
Sun	6AM - 5PM	60 - 70	ß1.50/min
Sun	7PM - 11PM	60 - 70	ß5/min

The modifier `onlyDecibel` ensures that new noise contracts can be created through $\mathcal{C}_{DecibelLive}$ only by the Decibel.LIVE platform.

4.2.4 *Contract Monitor*

A `monitor` is a local authority that wishes to monitor noise at a particular site in real time. A `monitor` is also assigned a wallet consisting of a key pair and unique Ethereum address $a^{monitor} = (pk'', sk'')$ constructed from the public-key. However, a `monitor` is a read-only observer in the system.

4.2.5 *Bel token (ß)*

The DApp platform uses *Bel* token (ß), an ERC20-compliant token built on top of Ethereum, as the unit of compensation on the blockchain-based platform. Within Decibel.LIVE, `initiator`s define compensation policies using *Bel* tokens. A `beneficiary` receives compensation in the form of *Bel* tokens. *Bel*s can exchanged on a crypto currency exchange or can be reused on the platform.

The distribution of *Bel*s and the token supply is determined by a smart contract deployed on Ethereum.

4.2.6 *Payout Policies*

An `initiator` also declares incentives in the form of compensation when noise exceeds a certain threshold at certain times during the day (or night) by hour(s) of day and day(s) of week. Such *payout policies* are encapsulated in $\mathcal{C}_{Noise}$ by an `initiator` when a noise contract is created (see Table 2). When the contract becomes active, the policies can not be changed.

The payout settlement algorithm (see Algorithm 1) evaluates every incoming noise event as per the payment policies in a noise contract.

4.2.7 *Noise Contract Expiration*

A noise contract expires when $endsAt \geqslant block.timestamp$. *block* is a built-in Solidity property and *block.timestamp* gives the Unix timestamp of the latest block. Following method in $\mathcal{C}_{Noise}$ expires a noise contract [6].

```
function expire()
    onlyDecibel
    inState(State.Active)
{
    if(endsAt >= block.timestamp) {
        expired();
        state = State.Inactive;
    }
}
```

4.2.8 Noise Contract Cancellation

An `initiator` can abort a noise contract before it becomes active. Both web and mobile applications provide means to abort a noise contract if it's not yet in *Active* state. The modifier *inState*(*State.Created*) ensures such guarantees.

```
function abort()
    onlyDecibel
    inState(State.Created)
{
    aborted();
    state = State.Inactive;
}
```

4.2.9 State Channel

Decibel.LIVE uses Telehash[7], an encrypted mesh network protocol for communication between the `initiator` and each `beneficiary`. Telehash enables a completely distributed, decentralized communication mechanism that doesn't rely on on centralized or federated servers.

When a noise contract $\mathcal{C}_{Noise}$ becomes active, the `initiator` and every `beneficiary` of the contract is assigned a unique Telehash address known as a *hashname*. A hashname is a 32-byte string that corresponds to the SHA-256 hash of one or more public keys. A link or a *channel* is created connecting the `initiator` to every `beneficiary` of the contract. All messages exchanged on a channel are encrypted end-to-end all of the time using forward secrecy.

$\mathcal{C}_{Preimage}$ creates a new channel on behalf of an `initiator` with an opening balance of ßb and hash $h = \mathcal{H}(\sigma)$ where σ is secret chosen by the `initiator`. ßb is the maximum amount the `initiator` is willing to pay while the contract is active. The `beneficiary` joins the channel after confirming the owner of the channel using *ecrecover* function in Solidity.

```
address initiatorPK = ecrecover(h, r, s, v)
```

where r, s, v represents the signature of `initiator`'s transaction.

$$a_1^{initiator} \xrightarrow[\mathcal{C}_{Preimage}[h, T_{Expiry}]]{ßb} a_2^{beneficiary} \tag{5}$$

The Decibel.LIVE app captures loudness of noise on a `beneficiary`'s smartphone, computes and sends $L_{\mathrm{Aeq,T}}$ events in JSON format over this channel to the `initiator`. Each noise event from a `beneficiary` is evaluated and if eligible, a payout commitment is sent from the `initiator`. The `beneficiary`'s balance in the wallet is updated in real time.

A `monitor` may observe noise events events originating from a site in real time by establishing a channel to the `initiator`.

$$a_0^{monitor} \xrightarrow[\mathcal{C}_P[h', T_{Expiry}]]{ßb} a_1^{initiator} \xrightarrow[\mathcal{C}_P[h'', T_{Expiry}]]{ßc} a_2^{beneficiary} \tag{6}$$

where $\mathcal{C}_P$ is the preimage contract.

4.2.10 Compensation Settlement

The $L_{\mathrm{Aeq,T}}$ events are input into the payout settlement algorithm (Algorithm 1). Each incoming event is checked against the payout policies declared in the associated noise contract. If an $L_{\mathrm{Aeq,T}}$ event is eligible for compensation, the `beneficiary`'s balance is updated on the channel in real-time.

When the contract expires, the balances on the channel are published on $\mathcal{C}_{DecibelLive}$ permanently recording transfer of funds.

Algorithm 1 Payout Settlement

```
[C_1...C_n] ← GetNoiseContracts(C_DecibelLive)
for i ← 1 to n do
    while C_i is active do
        PolicySet_i ← GetPolicies(C_i)
        Evaluate f(PolicySet_i)
        for all noise events ϵ in ξ do
            if f(ϵ, PolicySet_i) is payable then
                payout_i ← Eligible(ϵ, PolicySet_i)
                TransferFunds(C_DecibelLive, C_i, payout_i)
            end if
        end for
    end while
end for
```

4.3 Veracity of Data

The degree to which data is accurate and reliable is the biggest challenge of the application. To that end, Decibel.LIVE aims to ensure high accuracy of sensor data and low probability of incorrect inputs from a dishonest actor.

4.3.1 Dishonest Actors and Gaming the System

There are a few ways in which dishonest actors might try to game the system for their own financial benefit.

a) **Joining the contract as beneficiaries when they don't reside within the contract zone** It is feasible that individuals might try to temporarily position themselves within the approved contract zone for the purpose of joining and thus benefiting from the contract. The incentives for doing this are minimal given that the individual(s) would need to be present within the contract zone for a reasonable amount of time to benefit in any substantive way. However, this possibility does need to be considered and will be mitigated through authentication i.e. pre-issuing a unique passcode to each nearby resident which must be entered to join the contract.

[6] Currently the Decibel.LIVE platform invokes *expire*() method using a scheduled job. In future versions, we plan on using an Oracle service to execute methods at a scheduled future time.

[7] https://github.com/telehash

b) **Artificially distorting noise readings to increase payouts** Individuals might try to increase the noise readings through a variety of techniques like pre-recording the noise closer to the sound source and playing it back, generating another noise source within their residence, blowing into the microphone on their smartphone to increase the sound pressure etc. These techniques will be mitigated through sophisticated analysis of sound snippets (short duration audio recordings of $< 1sec$ taken at periodic intervals) for those noise readings that are identified as statistical outliers. Options such as banning the user for the duration of the contract or a fixed period of time (e.g. 30 days) may also be considered. It is eventually planned that much of this process will automated using AI.

c) **Joining the noise contract multiple times** This will be resolved by allowing each resident to join the contract once only using pre-authentication techniques described above. In addition, GPS locations for noise readings will need to correspond to the beneficiaries location essentially eliminating the ability for a resident to join/monitor the contract on behalf of another nearby resident but all from their within the bounds of their own residence.

4.3.2 Calibration

Prior to capturing noise levels, the smartphone app performs a set of calibrations. Data from an improperly calibrated smartphone is misleading and not useful. GPS modules in modern smartphones have a very high positioning accuracy sufficient enough to calculate the distance from a noise site. The app correlates GPS data with known information to determine the veracity of noise events.

5. CONCLUSIONS

In this paper we presented Decibel.LIVE, a noise monitoring and incentive platform which enables citizens monitor noise and report noise events to noise site owners on a private encrypted channel. Site owners can define compensation policies based on the loudness of noise. Decibel.LIVE helps citizens receive instant compensation from noise site owners when noise exceeds thresholds defined by site owners.

6. ACKNOWLEDGMENTS

We express our gratitude to Dr. Ralph Muehleisen for help with signal processing and IIR filters. Thanks to Steve Anderson for long discussions on many of these ideas and thanks to members of the Ethereum community who have been generous with their knowledge.

7. REFERENCES

[1] D. A.Bies and C. H.Hansen. *Engineering Noise Control: Theory and Practice*. Spon Press, 1996.

[2] R. K. Andrew Miller, Iddo Bentov and P. McCorry. Sprites: Payment channels that go faster than lightning. February 2017.

[3] R. Chepesiuk. *Decibel Hell: The Effects of Living in a Noisy World*. National Institue of Environmental Health Sciences, January 2005.

[4] L. B. Christopher D. Clack, Vikram A. Bakshi. Smart contract templates: foundations, design landscape and research directions. 2016.

[5] Ethernodes.org. Ethernodes.org. `https://www.ethernodes.org/network/1`. Accessed February 25, 2017.

[6] P. C. H. Hansen. Fundamentals of acoustics. pages 24–27, 1996.

[7] C. A. Kardous and P. B. Shaw. In *Evaluation of smartphone sound measurement applications*, March 2014.

[8] M. Loeb. *Noise And Human Efficiency*. Chichester: Wiley, 1986.

[9] N. Maisonneuve, M. Stevens, M. E. Niessen, P. Hanappe, and L. Steels. In *Citizen Noise Pollution Monitoring*, 2009.

[10] M. Meeker. Kpcb: Internet trends 2016 - code conference. `http://www.kpcb.com/internet-trends`. Accessed February 25, 2017.

[11] D. Radu, C. Avram, and J. Y. Adina Astilean. In *Acoustic noise pollution monitoring in an urban environment using a VANET network*, May 2012.

[12] S. Santini and A. Vitaletti. In *First experiences using wireless sensor networks for noise pollution monitoring*, April 2008.

[13] S. A. Stansfeld and M. P. Matheson. Noise pollution: non-auditory effects on health. Department of Psychiatry, Medical Sciences Building, Queen Mary, University of London, 2013.

[14] C. J. e. a. Vallet M, Gagneux J. Heart rate reactivity to aircraft noise after a long-term exposure. in: Rossi g (ed) noise as a public health problem. milan: Centro recherche e studioamplifon. pages 965–75, 1983.

[15] G. Wood. Ethereum: A secure decentralised generalised transaction ledger. 2014.

APPENDIX

A. GLOSSARY

Blockchain A blockchain is an append-only distributed ledger of transactions. Transactions confirmed and verified by multiple nodes (computers connected forming a blockchain network with the task of validating and relaying transactions) are recorded as "blocks" which are added in a linear, chronological order with each blocking linking its predecessor forming a chain from the very first block. Each node gets a copy of the blockchain, which gets downloaded automatically upon joining the network.

Direct Field A direct field of a sound source is the area of the sound field which has not suffered any reflection from any surfaces or obstacles.

Elastic Medium A substance or material (e.g., air , water) that can change its form as a result of a deforming force but can return to its original form when the deforming force is removed. For example, a gas is an elastic medium as it can be compressed under pressure but will regain its former volume when the pressure is released.

ERC20 Token A *de facto* token standard to create digital assets one can purchase and own on the Ethereum blockchain. The standard defines a set of function signatures such as `send`, `register`, `delete`, in the Ethereum contract ABI language. Using the ERC20 token standards, crypto currency exchanges, users and applications easily interact with each other exchanging currency units, registering names, making offers on exchanges, and other similar functions.

Free Field Free field is a region in space where sound may propagate free from any form of obstruction.

Frequency Analysis Frequency analysis is a process by which a time varying signal in the time domain is transformed to its frequency components in the frequency domain. There are two ways of transforming a signal from the time domain to the frequency domain. The first involves the use of band limited digital or analog filters. The second involves the use of Fourier analysis where the time domain signal is transformed using a Fourier series. This is implemented in practice digitally (referred to as the DFT - digital Fourier Transform) using a very efficient algorithm known as the FFT (Fast Fourier Transform).

Hash function A hash function is a mathematical function that takes an input of any size and produces an output of fixed size.

Impulse A signal can be decomposed into a group of simple additive components called *impulses*. Generally, an impulse is a short-duration time-domain signal. A *unit* impulse is a normalized impulse where sample number zero has a value of one, while all other samples have a value of zero. A unit impulse is also known as a delta function and is denoted by $\delta(t)$ for continuous systems and $\delta[n]$ for discrete systems.

Impulse Response Impulse response is the output signal of a system when a delta function (unit impulse) is applied as the input.

Hashlock Hashlock is a type of restriction placed on a transaction such that a user must provide the input string used to produce the hash which is part of the transaction.

Oracle Oracles are special smart contracts that connect a contract on the blockchain with outside datasources.

Signals and Systems In digital signal processing, a signal is a description of how one parameter varies with another parameter. For e.g., voltage changing over time in an electronic circuit, or brightness varying with distance in an image. A *system* is any process that produces an output signal in response to an input signal. A system is called *continuous* if its input and output are continuous signals. For e.g., a *voltage* depending on *time*: $v(t)$. A system is *discrete* if its input and output are discrete signals. For e.g., a stock *price* measured by *day*: $p[d]$.

Transaction A message from a user to be recorded on the blockchain.