

Swupel PMS

A blockchain based real-estate solution

Contents

1. Abstract	4
2. Introduction to blockchain technology	4
2.1 Possible attacks and problems	4
2.1.1 Double spending	4
2.1.2 Forking.....	4
2.1.3 51% attack.....	5
2.1.4 Sybil attack	5
2.2 Consensus mechanisms	5
2.2.1 Proof of work	5
2.2.2 Proof of stake	5
3. Introduction to the Swupel PMS Project	5
3.1 Problems to solve.....	6
3.2 First world inconveniences	6
3.3 Solutions to mentioned Problems	6
3.4 Solutions to first world inconveniences.....	6
4. Block structure	6
4.1. Index:.....	7
4.2 Previous hash:	7
4. 3. Transactions:	7
4. 4. Locdata	7
4.5 Signature	7
4.6 Timestamp.....	7
5. Transaction structure	7
5.1 Sender	7
5.2 Recipient	7
5.3 Amount.....	7
5.4 Signature	8
5.5 Properties.....	8
5.6 Computational data	8
6. Smart contracts	8
6.1 Code structure.....	8
6.1.1 Transaction returning contracts.....	8
6.1.2 Domain smart contracts.....	9
7. Validation process	10
7.1 Validating Transactions	10

7.2 Artificial intelligence in the validating process	10
7.2.1 PMS map API	10
7.2.2 Input production	10
7.2.3 Input structure	10
7.2.4 Neural Network Model	11
7.2.4 Interpreting the Predictions.....	11
7.2.6 Handling improper zoning.....	11
7.3 Handling reported properties	12
7.4 Further steps in validating a block	12
8 Choosing a validator.....	12
8.1 Laying the fundament for biased randomness	12
8.2 The need for a sigmoid function	12
8.2.1 Logistical function overview.....	12
8.2.2 Usage of the before defined logistic function.....	13
8.3 Final decision.....	13
8.4 Oracle	13
8.5 Comparing the logistic function to a linear equivalent.....	13
9 Reward calculation.....	14
9.1 Reward factors for different types of Validators	14
9.1.1 Transaction shard validator	14
9.1.2 Property shard validator	14
9.1.3 Advanced zoning validator.....	14
10 Government incentives.....	14
10.1 Taxation.....	15
10.2 Zoning.....	15
10.2.1 Nature preservation.....	15
10.3 Fighting corruption.....	15
10.4 The power doesn't end up in one hand.....	15
11 Conclusion	15
Table of figures.....	16
Table of equations.....	16
Bibliography	16

1. Abstract

With further development in the blockchain space, consensus mechanisms have become the focus of many research papers, but no one of these innovations has been able to provide a decentralized solution with minimal electricity wasted. Comparing proof of work, and proof of stake, the environmental advantages of proof of stake are overshadowed by its lacking decentralization, problems such as the nothing at stake problem and manipulation possibilities of the validator choosing algorithms. Due to the nature of compound interest the buildup of whales is a big problem especially for proof of stake, while proof of work is struggling with big mining pools and mining farms. In the paper below an experimental consensus mechanism is explored which was developed to power a real estate blockchain with two native currencies. Proof of loc aims to be decentral, reduce energy consumption, while providing speed to currency transactions.

2. Introduction to blockchain technology

As a decentralized data storage technology, blockchains enable the creation of digital finance goods such as currencies, certificates of ownerships and other fungible and non-fungible tokens. The security of such blockchains is guaranteed by participants who agree to save a copy of the chain and check the validity of its new transactions, in turn for sacrificing their hardware these so-called miners or validators receive a reward, usually in form of crypto currency. This process of validating transactions usually involves checking the senders balance and subtracting the sent amount to check if the sender can afford said transaction. After validating a certain number or timeframe of transactions, the valid transactions get grouped together in a so-called block, these blocks of transactions, usually also holding some extra data such as indexes, timestamps or other technicalities make up the blockchain.

2.1 Possible attacks and problems

Like every other software, blockchains also suffer from possible attacks and problems. These are, compared to other ways of storing data or handling transactions, rather rare and often harmless but have to be addressed when building a new consensus mechanism.

2.1.1 Double spending

Double spending is one of the fundamental problems blockchain tries to combat, as a user should not be able to send more money than they have as this would undermine the system in a major way. The main idea in prohibiting this type of failure is to only add valid transactions to new blocks as described above. Double spending can occur due to major bugs, bad consensus architecture and if the system has been centralized by an authority (due to an attack or centralized blockchain architecture).

2.1.2 Forking

A fork describes a change in the blockchain network, similar to a software update, these might provide new features or improved security. There are two kinds of forks: soft forks and hard forks, the first refers to a backwards compatible change in where both versions of the blockchain can still communicate and work together as one blockchain, if this is not the case the fork is referred has a hard fork and the blockchain split into two new blockchains. Forks, especially soft forks, are a big part of improving a blockchain and should as such not be seen as a problem in general. However due to bugs or bad consensus architecture, hard forks can occur accidentally, effectively splitting the blockchain into two or more independent versions. If two versions become independent, sending money from one version to another is impossible.

2.1.3 51% attack

Should a party be able to seize control of more than 50% of the systems validators, validating invalid transactions would become possible as well as generating huge amounts of coins through double spending. An attack like this would require huge amounts of resources and wouldn't be profitable in most cases as the price of the currency would start to drop extremely fast after the attack becoming public, but could absolutely devastate a currency.

2.1.4 Sybil attack

As an approach to achieve a 51% attack the Sybil attack utilizes low mining costs to generate a huge number of validators under their control. These fake many validators then act in unity effectively achieving a consensus majority. With an increase in mining cost sybil attacks get less profitable, or as Julian Hosp stated: "The reason why mining can't be free is this [Sybil] attack." (Hosp, 2018).

2.2 Consensus mechanisms

The protocols which manage how new blocks are created, transactions are validated, and validators/miners are chosen are referred to as consensus mechanisms. These protocols form the backbone of every blockchain project as they are the main factors in shaping the end products performance, security and applications. Due to different demands and applications, a wide variety of mechanisms have been developed, although only the most popular 2 approaches are listed below.

2.2.1 Proof of work

As the first ever described consensus mechanism (nakamoto, 2009), proof of work is centered around the idea of solving complex mathematical puzzles (mining) to claim the chance to add a new block of validated transactions to the chain. These puzzles usually get solved via a central processing unit in a computer, resulting in high energy consumption, which is often deemed proof of works biggest weakness. Although many new approaches try to combat this inefficiency by designing new puzzles with different requirements such as Ethereum's GPU mining and different forms of storage-based mining.

2.2.2 Proof of stake

By using a locked-up amount of coins as "stake", proof of stake offers a higher chance of getting to validate the next block to those who have a bigger amount of coins locked up while punishing the validation of wrongful transactions by taking away part of the validators stake. Cutting the need for energy intense puzzles, this consensus mechanism is much more energy efficient and often much faster than its proof of work counterparts, while incentivizing holding the native currency to profit from compound interest. These advantages tend to come at the cost of decentralization as wallets with bigger balances experience exponential growth massively contributing to the formation of whales with extreme influence on supply and demand. Commentaries such as "In addition to the moral concern, that staking rewards the rich evermore, the possibility of constant forking represents a drastic technical drawback" (Hosp, 2018), depict proof of stakes main flaws, which are, even in the eyes of crypto enthusiasts, still prevalent.

3. Introduction to the Swupel PMS Project

The Swupel PMS blockchain, which is to be powered by proof of loc, brings real estate onto the blockchain, by, working together with local governments, providing a property register which can be accessed via an app/website, enabling common people to sell their properties and make smart contracts such as renting or usage agreements. Every property transaction

requires a smart contract to be signed, coin transactions can be made without such contracts. Coin based contracts can be found in wireless payment systems or agreements in between two private persons. The main coin chain is on a different shard than the property-based chain, this is to ensure maximum transaction speed for everyday transactions. Property transactions must be validated by an artificial intelligence, this consumes much more electricity and takes significantly longer, but as property transactions don't have to be as fast anyways, a slower block time on the property shard is to be established.

3.1 Problems to solve

With an ever-growing wealth and power difference between rich and poor individuals/communities, zoning violations and land robbery committed by big cooperation's or corrupt government officials are hard to combat as many developing nations do not offer an easy access land register, access to legal infrastructure tends to be near to nonexistent in periphery regions and centralized corruption works in favor of the criminals.

3.2 First world inconveniences

Having to deal with high fees, long waiting times and a lot of legal documents most analog land registers are an absolute disaster when it comes to efficiency and user friendliness. Working with staples of documents just to prove ownership of a property, while making renting out said property almost impossible without legal help.

Some countries try to battle these problems by digitalizing the analog systems, often resulting in very insecure office software for the same people who would otherwise work with the analog land register, not creating much of an improvement after all.

3.3 Solutions to mentioned Problems

Decentralizing the land register and making it public effectively fights the 2 most prevalent reasons for growing corruption: "A system in which making money is the biggest priority with minimal incentive to behave correctly" and "laws attracting money laundering whilst shielding foreign criminals from prosecution" (Milanovic, 2019), while ensuring the same legal treatment for every participant. Implementing the checking of zoning into the consensus mechanism even protects those, who cannot directly utilize the legal tools offered to them, from industrialization, illegal buildings, deforestation, mining, plantations and any other zoning violation.

3.4 Solutions to first world inconveniences

With the ability to buy and sell properties by accepting standardized contracts almost instantaneously while paying minimal fees and requiring no legal assistance, the joy of buying a new home, forest or any other property will not get overshadowed by complicated and expensive legal processes. Renting out a flat can be as easy as filling out a contract layout and pressing sent.

Being blockchain based hacking is an unreasonably small threat to such as system. Making a difference for the end user and not the unnecessary legal experts between the property and its owner.

4. Block structure

A block is made up of the following elements:

4.1. Index:

Marks the index of the block in the blockchain. The genesis Block represents index 0, every following block adds to this number.

4.2 Previous hash:

When creating a new block, the validator takes all elements of the block before and hashes them with sha256, this hash then forms the second element of the new block, this is crucial in validating the chain as any change in the previous blocks would result in a completely different hash.

4. 3. Transactions:

Containing all the transactions of the given block the transactions list will be described closer in chapter 3 Transaction structure.

4. 4. Locdata

The locdata represents the geographical location of the miner, which does not have to be the location of the actual device on which it was validated but is rather the average location of all the user's properties. Depicted in a set of latitude and longitude this will be used to calculate the mining reward.

4.5 Signature

As a validator validates a block, he generates a hexadecimal signature with his keys. This signature proves that said validator confirmed that every transaction in this block was valid. If a validator validates wrong transactions, he is penalized and has to pay a fee of his stake.

4.6 Timestamp

The last element of a block is the timestamp, this timestamp is to be ever increasing otherwise the block gets deemed invalid. This makes manipulating past blocks much easier and helps to maintain an instance of order, as well as in combination with minting timestamp can provide a second level of safety.

5. Transaction structure

As a key part of two blockchains the structure of its transactions often plays a big role in performance and mining efficiency. The following information will be centered around general transaction structure and offer less coverage of smart contracts.

5.1 Sender

The hexadecimal public key of the transaction sender.

5.2 Recipient

The hexadecimal public key of the transaction recipient.

5.3 Amount

The amount represents the number of fungible coins send in the transaction. Every transaction has to involve coins in order to pay fees. These fees will be calculated before and automatically added to the original transaction amount. A target of about 2 % fees is to be chased, although the exact amount of fees might vary a bit depending on system usage and many more factors.

5.4 Signature

A hexadecimal signature made by the sender of the transaction to testify being the sender of the transaction. This signature is key to prevent sending transactions in the name of other people.

5.5 Properties

The sent properties are represented by all their edge points, these shape a polygon, which can then be projected onto any map. When a property gets minted, zoning, time and further data is saved on the blockchain. This data can be accessed via an inbuild blockchain explorer, but does not get sent in further transactions.

5.6 Computational data

Smart contract code can be saved to the blockchain in these sections even if the contract in question is a non-repeating contract. (See 6. Smart contracts)

6. Smart contracts

Enabling the creation of multiple different types of smart contracts, the Swupel PMS blockchain can support a multitude of complex features, some of which will be elaborated on below.

6.1 Code structure

Although prone to changing with different features added to the blockchain a smart contract always consist of a string which is passed to the computational list of a transaction and will be computed by the next miner validating the transaction.

The first letter of the multi-line string that will be passed to the computational data list in the transaction determines how the smart contract will be processed. These are the currently available indication letters:

n: Builds a domain for your public key (e.g. nbob), this means all transactions sent to bob will be forwarded to the public key which launched the corresponding smart contract.

c: Builds a one-time smart contract which adds a transaction to the open transactions of the next block. This can be combined with checking balances, time or fee prices.

r: Builds a repeating, transaction returning, smart contract which can be used to build decentralized infrastructure such as money distribution, insurances and many more (comparable to Ethereum's smart contracts).

6.1.1 Transaction returning contracts

These contracts (initiated with the letters c and r) are to be programmed to return a Transaction object with the variable name t. This variable will then be appended to the list of open transactions. This enables the creations of complex smart contracts such as the example below.

(The green letters are comments explaining the code below)

```
#initialising the repeating smart contract with the letter r and making
the variable t global
rglobal t
#Initialising the blockchain object used to access functions and data
(sc is the address of the contract, '' is the private key, 5000 is the
node id)
blockchain = Blockchain('sc','',5000)
#Checks if own balance is beigger than 10
if blockchain.get_balance('sc') > 10:
#    if its bigger than 10 it sends 10 coins to bob and prints the
    remaining balance
    t=Transaction('sc','bob','',10,[''],[])
    print(t, blockchain.get_balance('sc'))
else:
    #if not it just prints the balance
    t=None
    print(blockchain.get_blanace('sc'))
```

6.1.2 Domain smart contracts

A domain enables user friendly sending of funds to a short address in string form similar to dns servers on the internet. Eliminating the need for 324 characters long addresses greatly improves user friendliness.

Initialized with an n these are the simplest contracts by simply just stating the desired domain behind the n. In the example below a contract for the domain name of bob is shown:

```
#n initialises the domain name contract and bob represents the desired
address
nbob
```

6.1.3 Primains

Although they should not be considered a classical smart contract Primains as in Pri-(mus) which is Latin for the first and (Do)-main relating to the well known term in computer science and web development mostly used to connect to DNS servers.

These Primains are similar to crypto domains in the sense that they enable user friendly sending of transactions without the need for copy and paste. As every user gets their own Primain without the need for any action on the user end, the Primain acts as a first, uncustomized domain, similar to an IBAN in centralized banking.

Normally providing every user with their own domain would result in huge strain on the chain as even unused wallets would acquire quite a lot of storage space. To combat this, the Primains get generated in reference to the public keys of the users. This means a function is capable of turning any public key into a sequence of three words connected by dots. As the word database a file of the 3000 most used English words can be used as this number cubed results in 27 000 000 000 possible combinations. The other way round a function can also determine the public key of a user by inputting their Primain.

7. Validation process

In order to participate in the network and start making money by helping to ensure all transactions and zonings are valid, a potential validator has to lock up a part of his coins. This is to ensure that validating wrongful transactions has consequences and can be punished by taking a part of the locked-up stake. Owning a property is the second requirement for starting to validate, as the average location of all the users' properties counts as their location.

7.1 Validating Transactions

Before a new block can be appended to the current blockchain, all the transactions inside the new block have to be validated. The first part of the validation process is, by subtracting the transaction amount from the users' balance, checking if the transaction sender has enough coins to complete the transaction. Validating transactions on the property shard involves, after to checking the coin balance and confirming all sent properties are owned by their senders, checking the zoning via an artificial Intelligence.

7.2 Artificial intelligence in the validating process

Checking zoning via an AI provides protection to common people by ensuring equal chances for every participant in the system. As a part of validating transactions, every change in property ownership also equals to a zoning check via AI. To ensure also checking properties which do not frequently change owners, after validating every sent transaction, the system randomly chooses 0,00015 % of all minted properties to have their zoning validated as well. This is to ensure every property gets validated, on average, every 7,5 Months.

7.2.1 PMS map API

In order produce reliable inputs for the neural network, an API has been developed which enables inputting the coordinates of the edge points and getting an image output of the area inside the property from a bird's eye view. These satellite or aircraft captured images are always going to slightly vary in quality, color and texture, combined with the fact that most properties are very different in shape, the returned images cannot be directly inputted into the neural network.

7.2.2 Input production

After the property coordinates have been passed through the PMS map API an image of only the land in between the edges of the property gets returned. Everything outside the property is blacked out to prevent influencing the AI in any way.

7.2.3 Input structure

As the neural network, which forms the judging AI used in the validating the zoning, requires a standardized input to make reliable guesses, the before accumulated data has to be formatted into an appropriate array, which can be processed by the ai. The big picture gets split into multiple small pictures with a minimum resolution of 500 pixels. These then get downsized to 250x250 pixels. Non quadratic property shapes producing black edges protruding into the image, has been accounted for when training the model. These small images have to be turned to grayscale and then converted to an array. For every property and their arrays an according zoning law has to be collected and saved as such that it can then be compared to the prediction results. After accumulating as many of these arrays as the ram on the machine allows, these can then be forwarded through the following model.

7.2.4 Neural Network Model

Providing an input in the shape of an array, with each value of the array corresponding to pixel on an image, a sequential, convolutional neural network provides the best training rate and energy efficiency of all possible network types. After thousands of epochs and hundreds of possible network structures, the following architecture has proven to be the best ratio in between energy efficiency, speed and accuracy.

Starting with 24 neurons, each with a rectified linear activation formula, the first layer has a filter size of 2 by 2, then adds a padding of zeros on the edges of the image in order to fight shrinking the image when passing through layers. After this the picture gets passed through a layer of Max Pooling which has kernel size of 2x2.

The second hidden layer is the exact same of the first.

Layers 3-5 are also made up of 24 neurons and operate with a filter size of 3x3. Padding and activation formula remain the same.

The last convolutional layer is identical to the first. After passing through this Layer the data gets flattened, a dropout of 30% is added to combat overfitting and passed to the non-convolutional hidden layers of this model.

The non-convolutional part of the network is made up of 2 identical Layers and one very similar. Featuring a neuron count of 28, the first two Layers have a slightly elevated dropout rate of 50% and the through the network present rectified linear activation formula. As the third layer only features 24 neurons the dropout rate is slightly lower at 30%, otherwise it is identical to the two Layers before.

The last Layer of the network features a softmax activation function to not only make binary predictions but also calculate the accuracy of these predictions. With a neuron count of 4 the system can distinguish housing, farming and industry land from nature.

Using categorical crossentropy while compiling the model enables effective learning even with standard optimizers such as adam, as well as making accurate predictions with certainty ratings.

7.2.4 Interpreting the Predictions

After passing the batches of arrays through the neural network and comparing them to the official zoning register, those properties who did not match with the official records have to be forwarded through the network again, although this time the accuracy results are also evaluated on their own rather than just taking the highest score as the final prediction. The threshold which defines at which uncertainty a property gets reported strongly depends on the type of zoning.

Taking a farming permit as an example, an ai prediction of natural forest should not be considered a problem, especially if the ai returns similar percentages for both options.

Should a farming ground be deemed an industrial area, a prediction very much unlikely to be made by accident, the property gets reported.

7.2.6 Handling improper zoning

If a property has been convicted of violating zoning permits, the transaction, if the property got validated while being in an open transaction, gets cancelled and put on hold. After this process has been completed the property gets reported to a special type of validators which

use bigger, more complex, neural networks, higher image resolutions and a broader spectrum of satellite bands as well as a history of older satellite imagery, enabling much more accurate predictions. These new predictions are now analyzed and depending on their results either get reported for a zoning violation or the transaction gets marked as complete and then added to the next block.

7.3 Handling reported properties

Should a property, after all zoning checks, still violate zoning it gets reported to a council of human moderators working for the Swupel PMS company. These moderators, equipped with better software and more cartographical material, judge if the property really commits a violation, should this be the case, an official notice to the government is filed with all the available data from property and owner. How many of these notices end up being prosecuted is up to the local government, although we highly recommend a very high prosecution rate as every reported property has been checked multiple times by advanced AI and even a highly specialized human moderator with fitting tools, reducing the chance of a wrongful accusation way below 1%.

7.4 Further steps in validating a block

After all transactions have either been deemed valid or thrown out of the block a new block has to be formed by hashing the previous block, updating the index, putting all valid transactions in the transactions list, updating the locdata to the average location of all your properties, signing all the before mentioned elements of the new block, adding the before produced signature and finally finishing the new block by adding a timestamp. After this process has been completed, the new block can be appended to the blockchain and all transactions which have been validated in the new block, can be cleared from the open transactions. After all these tasks have been completed the validator has access to his coin reward and the new block gets broadcasted to the other nodes. With this, all duties of the validator are completed and the blockchain has gained a new block.

8 Choosing a validator

The validator choosing system is based on randomly electing a participating validator, considering the validators locked up coin stake and the sum of owned property areas. Formatting this process into a sequence of simple mathematical processes:

8.1 Laying the fundament for biased randomness

First a list of all active validators is created, then, for every element of this list, the balance of coins and square meters of owned property gets determined. These two values then get passed on to the next step.

8.2 The need for a sigmoid function

Most biased randomness is linear. This means stacking 2 coins in a conventional staking system usually equates to twice the chance of being elected as a validator for the next block compared to only staking 1 coin. Implementing such a staking system usually leads to a few whales controlling most of the coins in circulation, as, although the chance to be elected rises in a linear fashion, the actual wealth accumulates, due to compound interest, exponentially. To combat this proof of loc utilizes a sigmoid logistical function, instead of a linear equivalent.

8.2.1 Logistical function overview

A logistical function follows an S like shape being very flat at the top and the bottom. Setting 100 as the maximum and 0 as the minimum and taking the owned number of coins as an input,

the function is going to return a value between 1 and 100. Taking the average amount of coins (o) and the actual number of coins in the user's balance (x) and defining the steepness of the curve (k), the following function, seen in equation 2 can be defined. Setting the average amount of coins (o) as 500, the steepness (k) as 0.01 the equation will look as depicted in equation 1, producing the, in figure 3, shown graph.

$$f(x) = \frac{100}{1 + e^{-k(x-o)}}$$

Equation 2

$$f(x) = \frac{100}{1 + e^{-0.01(x-500)}}$$

Equation 1

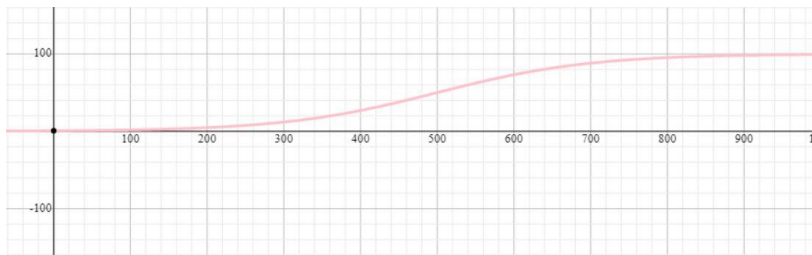


Figure 1

8.2.2 Usage of the before defined logistic function

Passing the number of owned coins through the logistic function will return a value between 0 and 100. The same is true for passing the owned land area in square meters. Through the same function. Both these values are then added, divided by 2.

8.3 Final decision

The higher the resulting number, the lower the yearly rate on the stake will be, this creates an incentive even for poorer users to stake part of their coins and gain interest on them.

8.4 Oracle

Taking away validator choosing calculations from the validator ensures to not only secure the system against possible attacks involving actively influencing who gets to validate the next block, but also minimizes energy usage by outsourcing complex logistical calculations.

The oracle tasked with the above-described process of determining the next validator also has to ensure a minimum of 50 backup validators of which everyone has to obey a 1 second responding time in order to claim the block, otherwise the next backup in the list will be able to claim the block. If a validator has been determined the process gets repeated.

8.5 Comparing the logistic function to a linear equivalent

The biggest difference between using a linear function, usually very simple, like $f(x)=x$ and implementing a more complex logistic function similar to the before mentioned one (see 5.2.1 Logistical function overview Figure 1) is its influence on coin distribution. With linear functions, coins tend to accumulate in a small selection of the richest wallets. This effect is also extremely prevalent in non-proof of stake systems such as Bitcoin's proof of work, as mining equipment price decreases exponentially with bigger orders, which makes for an even worse coin distribution. Nonlinear functions like logistic functions are much more complex than their linear counterpart, but effectively combat an indifference in coin balance, while still

incentivizing locking up bigger stake to get bigger returns. The key to this system is that being a validator is most efficient for the average person as rates decrease with every increase in the users coin balance.

9 Reward calculation

Aiming for a maximum in decentralization in not only an economical perspective, but also from a geographical standpoint, the validation reward gets calculated taking into account the distance to where the last block was validated. When calculating the distance between two sets of coordinates, due to earths spherical nature, the below shown haversine formula is applied with ϕ_1, ϕ_2 as the latitudes from both points and λ_1, λ_2 their longitudes.

$$d = 2r \arcsin \left(\sqrt{\sin^2 \left(\frac{\phi_2 - \phi_1}{2} \right) + \cos(\phi_1) \cos(\phi_2) \sin^2 \left(\frac{\lambda_2 - \lambda_1}{2} \right)} \right)$$

Equation 3

This calculation returns the distance between the location of the last miner and the average location of your properties in kilometers. Depending on the type of validator the result of the haversine equation will then be divided by a number between 100 and 1000.

9.1 Reward factors for different types of Validators

The reward factor refers to the number with which the distance between the last validation location and the location of the current validator gets divided. These factors are in place to limit inflation to a certain degree.

9.1.1 Transaction shard validator

Tasked with simple calculations and minimal energy effort these validators receive the least reward for each added block. Only needing to verify if transactions were sent from wallets with correct balances and checking signatures their needed resources are the lowest too.

9.1.2 Property shard validator

Having to handle much more complicated types of validations such as zoning checks, these validators need to sacrifice bigger amounts of electricity and have bigger demands from a hardware perspective. Combined with a lower block time a higher reward factor of 250 has been settled on.

9.1.3 Advanced zoning validator

Due to high hardware demands, less frequent usage, and more electricity consumption, being an advanced zoning validator offers a reward factor of 100.

10 Government incentives

Needing the support of local governments to be implemented, the question arises, why would any Government give away power from their institutions? Implementation concerns are extremely important when aiming to make a difference, after all the best blockchain is useless if no one uses it.

10.1 Taxation

Being able to collect tax on every property purchase, rent or any other smart contract on the blockchain would result in big increases in tax collection. Even if participants try to trick the systems by setting extremely low prices on the chain and paying for the property in local currency an ai could easily be trained to monitor transactions and report suspicious ones.

10.2 Zoning

As zoning violations are a crime and often result in tax fraud and harming of nature and local population, governments would profit massively if such violations could be detected with minimal spending. With a zoning validation system built into the consensus mechanism, violations and crime in this area could be combated with no extra cost from the government.

10.2.1 Nature preservation

With many economies being based on tourism, farming or fishing the preservation of nature is not only a morally important goal, but also serves as the backbone of most economies. Shielding national resources from international companies and corrupt officials also results in there being more resources for the own nation and enabling proper taxation of different industries.

10.3 Fighting corruption

As a public system with no hidden identities all transactions can be traced. This means receiving land or coins as a reward for a political favor can now be proven much more easily, enabling non corrupt politicians to persecute those who are corrupt and prove their innocence.

10.4 The power doesn't end up in one hand

Giving away power is dangerous for a government as in what the receiver could potentially do with the power, being a decentralized blockchain protocol Swupel PMS could however in no way shape or form use this power as a company or in any centralized way. A democracy would rather give the power to their very fundament, the citizens.

11 Conclusion

Written as an introduction to the Swupel PMS project, its underlying consensus mechanism and blockchain technology, this paper was meant to spark interest in ways to combat serious problems by the means of technology.

Utilizing AI to check zoning and rewarding validators based on geographical location, utilizing sigmoid instead of linear or exponential functions for calculating mining efficiency and chance to be chosen as the next miner, this blockchain aims to be truly innovative on a fundamental level, rather than just offering miner improvements in speed or smart contract usage.

By providing incentives for becoming a validator as well as for becoming a user of this system a maximum of decentralization, offering security for all users is achieved, while giving governments the benefits of reduced zoning violations and increased tax collection, without any additional cost.

Making owning and managing land possible for everyone, everywhere without any legal assistance, middlemen or any other complication.

Because properties represent homes.

Table of figures

Figure 1.....	13
---------------	----

Table of equations

Equation 1	13
Equation 2	13
Equation 3	14

Bibliography

Hosp, j. (2018). *Cryptocurrencies*. FinanzBuch Verlag.

Milanovic, B. (2019). *Capitalism, Alone*. Belknap Press.

nakamoto, S. (2009). *Bitcoin: A Peer-to-Peer Electronic Cash System*. Retrieved from
www.bitcoin.org: <https://bitcoin.org/bitcoin.pdf>