Zillerium - A Decentralised Supply Chain Architecture with Multi-variable Optimization

Trevor Oakley GDL (Law), BSc (Physics), MSc (Control Systems)

Abstract

We propose implementing key advances in Contract Theory, and to apply multi-variable optimisation methods to find self-sustainability in trade and hence enabling global omnipresent trade via blockchain technology with an ERC20 token called the Zillcoin. This proposal is so far reaching it would change millions of lives globally when implemented and dramatically change the world economy.

Keywords: ERC20, ERC223, Ethereum, Smart Contracts, Oracle, Supply Chain, Zillcoins.

1. Introduction

Supply chain inefficiencies are well documented [8]. The original development of Supply and Demand economics was based on work in the 1970s and has evolved into a array of complex theories relating to multi-variable optimisation. One of the more important set of theories comes under Contract Theory which last year saw a Nobel Prize being awarded to two academics [6]. Advances in economics have run in parallel to the development of blockchain technology ([14], [13]) which allows many of these economic principles to be implemented and also in parallel has been the evolution of devices which have factored into what is now referred to as Supply Chain 4.0 [9]. This document proposes a new architecture called Zillerium to implement modern economic principles on the blockchain and to meet standards set out in SC 4.0.

A joke once circulated that economics could be summed up in three words - "supply and demand." Many serious points are made in jest. Supply and Demand has one fundamental characteristic - the equilibrium point when demand meets supply. Trade outside of this point is unstable. Stability is one of the key concepts in all multi-variable optimization problems [12].

What we aim to achieve in Zillerium is the determination of Supply and Demand equilibrium points for markets and then to facilitate trade within those markets. This will enable especially trade to grow globally, extending to every country and leaving no one excluded. The key drive behind Zillerium is self-sustainability ([15]): we want to introduce trade under equilibrium conditions everywhere in the world.

Up until even a few years ago, such a proposition would have been impossible to take seriously. There were incalculable constraints: corruption, no bank accounts for consumers, illiteracy, dominance by corporations, imperfect markets (market prices don't control the market), and a culture that just large companies could trade effectively everywhere and sell a range of products.

Times changed as the blockchain technology showed it was far more extensive that just making a payment. This technology is able to extend so far that even a consumer without a bank account and even someone who has limited literacy can trade. With trade comes self-sustainability.

Zillerium intends to implement several key engineered products as follows:

- Product Oracle
- Pricing Oracle
- Risk Oracle
- Marketplace for Sale of Goods
- Aggregation prices in a mutli-sided market allowing advertising
- Payment protection via insurance
- Accounts management between traders in real time

With these core components we believe we will implement all the branches needed to enable trade everywhere and for every person. Such an innovation would dramatically change lives. In a world which throws away 30% of all food produced [5] and watches 50,000 people starve to death every day, there is a clear opportunity to reform supply: something which is predicted academically and in market research [4].

2. Definitions

For clarity we define some key terms.

e-commerce: Electronic commerce operates in all four of the major market segments: business to business, business to consumer, consumer to consumer and consumer to business [24]. E-commerce is far more than simply using shopping carts and websites. It is a broad term which refers to EDI, paymnents via an electronic network, and websites [25].

An oracle, in the context of blockchains and smart contracts, is an agent that finds and verifies real world occurrences and submits this information to a blockchain to be used by smart contracts [11].

3. Platform Model

3.1. Layer One: Zillerium Core

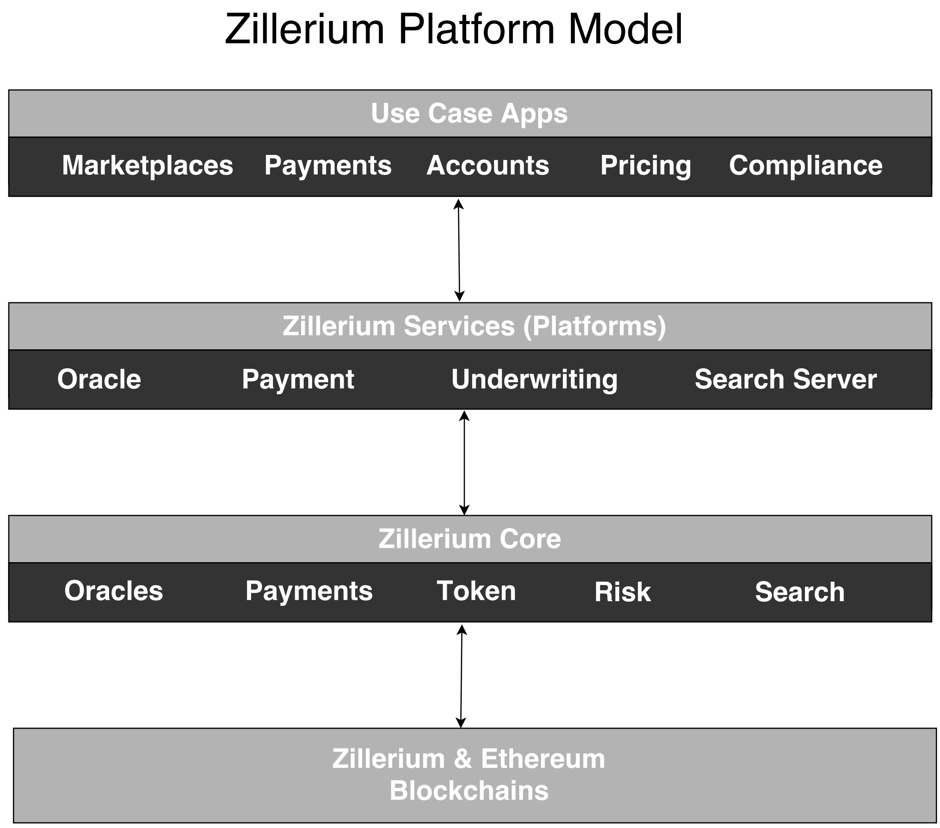
The Core layer has the fundamental smart contracts which create tokens, settle payments, manage risk marketplaces, and manage oracles.

Core	
Contract	Description
Settle Payments	Transfer Funds
Manage Risk	Ledger of Risk Classes
Manage Oracles	Smart Contract

3.2. Layer Two: Zillerium Services

The Service layer operates the underwriting marketplace, payment processor integrations, search, and oracle management.

Services	
Service	Description
Risk Markerplace	Listings of Risk
Payment Processor	API interfaces
Search	API to Elasticsearch
Oracle Management	Event Manager



3.3. Layer Three: Applications

The Application Layer is for front end apps primarily. The apps may be Zillerium ones or third party. An example could be a corporation which is self-branded and wishes to use pricing models from Zillerium based services (based on empirical evidence of market prices). The company could produce an app to read Zillerium and determine prices.

Other apps could be for small companies wanting to handle payments with some protection, and hence payment processing via Zillerim could be done.

Compliance officials could use their own apps to confirm goods are legal.

Apps	
Service	Description
Marketplaces	API Calls
Compliance Checks	API Calls
Pricing Models	API Calls

3.4. Blockchains

The Zillerium blockchain is permissioned based and store data which is not confirmed as being valuable by users. This is to reduce costs of maintaining Ethereum transactions. Confirmed data is inserted into Ethereum.

4. Zillerium Apps - App Layer

The following provides details about the use cases.

5. Core functionality

The design of Zillerium is that apps interact with the Zillerium Service Layer (ZS) and the Core (ZC) to get results and to add data into Zillerium. This section details some typical apps.

6. Advertising

6.1. Business Background

Pay per click models are well established online. They have been the main source of income of many large online companies. They depend on users being attracted to a website based on some third party interest in what is called a multi-sided market [7].

In the Zillerium design, the reason for the user visiting would be to find out the best market prices, this would akin to Google Shopping.

Sellers	Base Price	Total Price ▴	
Bradford Bathroom Company	£49.75 +£4.99 shipping	£54.74	Shop
British Bathroom Company	£49.75 +£4.99 shipping	£54.74	Shop
Taps Empire	£55.00	£55.00	Shop
eBay - chssupplies	£64.77	£64.77	Shop
eBay - plumb2u	£64.99	£64.99	Shop
PlumbArena.co.uk	£70.80 +£5.92 shipping	£76.72	Shop
eBay - elenor-liam	£84.92	£84.92	Shop
eBay - freefrance2015	£84.93	£84.93	Shop
eBay - tzahi.b	£88.17	£88.17	Shop
PlumbNation.co.uk	£84.36 +£7.20 shipping	£91.56	Shop

Google Shopping works by listing products based on seller input. This information is naturally biased in what is termed information asymmetry in economics [10]. We can introduce market pricing based on user events and hence someone can visit Zillerium to find out the market price.

A PPC event then can take a user to the seller's own site under a PPC model

In the overall market, PPC conversion rates are under 4% because the user has little detail about the market value of the price, and abandoned carts are commonplace online (60 %).

We propose to radically improve those conversion rates by adding more value to the transaction through market pricing.

6.2. Implementation

The app would call ZS via an API call and that in turn would return data via the ZC. A typical call might provide a unique part identifier which Zillerium then processes.

API Call:


```
{"product": {
 "partnumber": "36292000",
 "brandname": "Grohe",
 "apikey": key-value,
 "action": "allprices"
}}
```


API returned data:


```
{"product": {
 "partnumber": "36292000",
 "brandname": "Grohe",
 "description": " F-digital Veris F-Digital Digital controller
 for bath or shower",
 "currency": "GBP",
 "datetime": "Thu Sep 21 06:30:04 BST 2017",
 "pricevariation": "low"
 "pricing": {
   "sellers": [
     {"sellername": "BSO", "sellerprice": "530"},
     {"sellername": "Discounted Heating", "sellerprice": "604"},
     {"sellername": "Superbath", "sellerprice": "779"},
     {"sellername": "200arstore3010", "sellerprice": "993"},
   ]
 }
}}
```


7. DaaS

Under Data as a Service, a user will pay for data records only. There are licensing restrictions for most data and hence DaaS could only be done when the user has a legal purpose for the data (eg statistical analysis).

Some apps may want to determine trends in the market for products, eg the number of showers sold in a given period, or price movements. This is a statistical type of data service which would require an API call specifying what is delivered and Zillerium would return relevant data files. These could be downloaded from a dashboard.

8. Marketplace

The marketplace business model is well known and widely used. Companies such as ebay and Amazon have marketplaces.

We offer a distinction in that Zillerium is fundamentally decentralised as a market. Therefore data and pricing all originates from user activities. Also we propose lower of fees compared to established brands. Amazon can charge up to 25% as a fee.

These would be one of the commonest uses of Zillerium. This app extends the Advertising app use case already described. The user app would get all the data needed to list a product in a marketplace hosted by a third party.

9. Contract Quotes- SaaS

9.1. Business

There are several use cases in which contract quotes are worked out. For example if a hotel is being built, the architect will specify various goods to be installed, such as bathrooms, and the construction management will specify various goods such as wiring.

The subcontractors will buy the goods by getting various competing quotes. The companies quoting for the contract have to make up a quote listing all the goods and the costs. This is a very labour intensive process which can take hours for one quote.

Under Zillerium, the specifiers can state what the contract should have using the Product Oracle. The Pricing Oracle then provides a market quote.

Individual contract suppliers then can quote using the specification and applying their own cost model.

The specification would be entered into a permissioned based blockchain and then be accessible to the suppliers. They would quote using Zillerium and the buyer would then decide.

Zillerium would charge an operating fee in Zillcoins for this service - SaaS.

9.2. Implementation

A contract requirement would be stored in IPFS and accessible via the Ethereum blockchain. This would be accessible in turn via Zillerium. The user app would make an API call specifying a contract quote id and then the API returned data would list all the requirements for that contract with optional pricing.

10. Compliance

10.1. Requirement

There is extensive regulation regarding the sale of goods. This is magnified when exporting and importing. Presently to confirm approvals is very costly and non-compliance is commonplace.

We propose the Product Oracle using the Ethereum blockchain that compliance data would be visible to officials. For many instances in export, certain proofs are needed and this could be provided by Zillerium. For this use case, the seller of the goods would pay a fee which then registers credits for officials to use. This would in Zillcoins.

Officials would not pay to confirm compliance. They would scan a QR scan or an RFID tag and the event would enter Zillerium. Zillerium would identify the seller and the credit on the ethereum shared ledger for that seller and for that purpose (ie compliance) and then seller would be charged the usual product oracle fee.

This would be far less than current costs in obtaining certificates. Also legal sellers would have a huge competitive advantage which today is lost because the cost of verification that goods are legal is too high and hence lower cost goods stay in the market (Lemon Theory in economics [1]). This in economics is called signalling (communication of relevant sales information [2]).

For product listings at Zillerium a report button would allow some products to be flagged as unsuitable. Following such an action, the local community (ie for that seller) would judge the product listing and vote on its suitability. If the product is deemed illegal or unsuitable, the TV value would be adjusted to reflect that, eg -1.

Zillerium would not stop data being presented but simply flag data as unsuitable. Users would be able to filter out automatically unsuitable data. Zillerium is a self-governing platform and automated.

10.2. Implementation

An API call would list all compliance details in the returned data.

11. Payment Processing

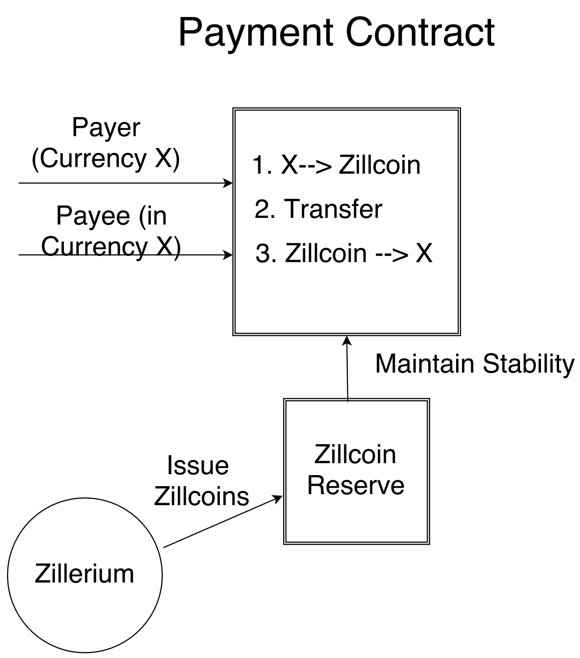
11.1. Payment

Third parties may use Zillerium to make payments in any currency which has a wallet connected to it and a cryptocurrency, eg a third party token. We propose using Shapeshift's API or a similar API service to convert currencies [26].

Hence any payer can pay to any payee in any currency. We convert the payment currency to Zillcoins to make the transfer and then convert to the paymnent currency to deliver to the payee.

The transaction is recorded in zillcoins and then insured as explained in this document.

A reserve fund ensures that the conversion does not lose any money, ie the time taken for conversion from currency X to Zillcoin and then back to X is done so that X is unchanged.



11.2. Payment Protection

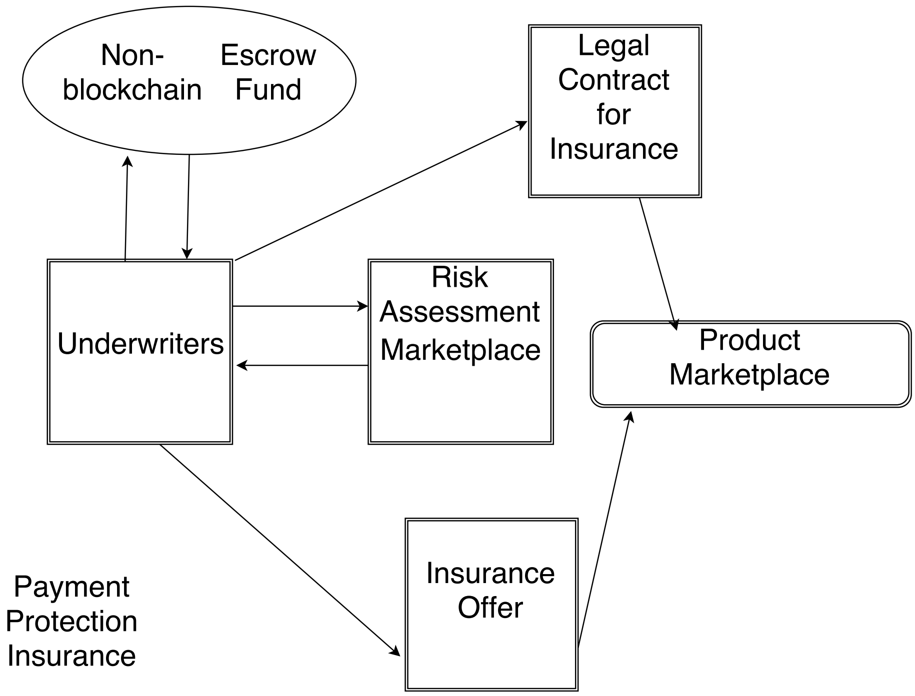
Today there is no payment protection for cryptopayments for the buying goods. Some companies have ventured escrow, [16], will work - we reject the proposal that escrow will work. Escrow will stifle cash flow and cause incalculable damage to trade. If escrow were implemented on a global scale which is suggested by some ventures, the economic devastation which would follow would be on a scale not seen for several decades.

We propose that sales transactions are underwritten by insurers. Insuring payments online is done already by paypal, ebay and Amazon. We extend their ideas so that the scope can be globalised. Our proposal is that peer-to-peer insurance is used.

This would work by underwriters contributing funds to an insured account (which is in ring fenced). This account is then made liquid in the market by insuring transactions based on risk and the underwriter will specify the risk they want to take.

Therefore, we would create a marketplace which had risk categories as follows:

- Country
- Product Type
- Price Band
- Reputation



As with all of Zillerium, this is all empirically based. Underwriters will determine by a marketplace action and by pricing options what is the market price to for example insure a transaction in Angola for the sale of a computer costing 100 dollars. If the market price to insure that transaction is 10 dollars, then the seller has to pay 10 dollars or reject the offer and not offer insurance.

Hence a risk assessment would be made via a Risk Oracle. The RO would produce a final risk level and underwriters would decide whether or not to accept it. They would fund a general insurance fund which is then allocated according to smart contract decisions based on the risk accepted and the pricing agreed in the marketplace.

The general insurance fund would exist off-chain and be protected by regulated services to stop theft and fraud. That fund would be used as security and the protection and payments in zillcoins would happen outside of that fund. In effect Zillerium would make insurance payments for claims based on refunds and then that liability would be placed on the balance sheet of the general insurance fund.

The original seller would be pursued for refund monies automatically, ie they would be required to pay zillcoins back to Zillerium or face account restriction and legal action. In any case the insured fund would cover the loss finally and not Zillerium.

At the time of contract formation, an arbitration process would be agreed. This could include the newly judicially accepted methods such as ODR (online dispute resolution, [21]) or the more established ADR (alternative dispute resolution, [22]) methods. These also would a jurisdiction of choice ([23]) to be chosen by the parties to the contract.

It would be unlikely for buyers to buy without insurance.

In extreme cases, escrow could be used, for example if no one would insure the transaction.

11.3. Implementation

A smart contract would make a transfer between to addresses on the blockchain. Zillerium Services would accept the payment request and then a smart contract would be accessed via Web3 to make the actual payment.

12. Accounts

12.1. Requirement

E-commerce is far broader than simply buying on websites. Many traders today pay via e-banking but order goods on the phone and they also get paper invoices.

We propose that traders using B2B methods could use Zillerium to order goods and get realtime invoicing based on the Ethereum blockchain.

A buyer would use the Product Order to create a purchase order which would then be entered into Ethereum against their user id held on Zillerium (ie stored at IPFS). Then the seller would invoice against that Purchase in real time. The payment would then be made using zillcoins or a cryptocurrency.

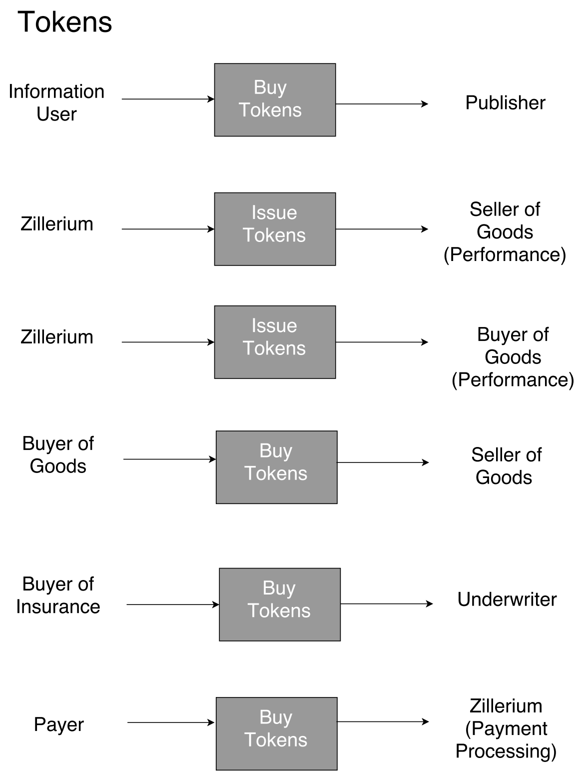
12.2. Implementation

Users would receive accounts based data after making an API call.

13. Zillcoins

The primary focus of Zillerium is the sale of goods. To sell goods an infrastructure is needed to ensure payments are protected, goods are legal to be sold, and there is market based pricing. The Zillcoin is the token to manage the financial aspects of the infrastructure so goods may be sold.

The Zillcoin would be ERC20 and ERC223 compliant [20].



13.1. Publisher

To sell goods, a lot of information is required. This includes product details, compliance details, pricing information (market based), and risk management details.

Mostly this data is all supplied via publishers into Zillerium who are then paid Zillcoins when data is used. For example if a seller wants to build a catalogue of products held at Zillerium, the user could make API calls to get the product details, or perform searches via Zillerium. Searches via a Zillerium app for end users are free but API based ones are charged with payments being paid to Zillerium and also partly to the publishers of the original data.

Data simply uploaded and not used does not earn any payment.

Payments are all made in zillcoins.

13.2. Rebates

In normal sales, rebates are very common. These encourage people to buy and sell. We have the equivalent in Zillerium. People can earn zillcoins for buying on a regular basis (similar to airmiles) and for selling under competitive conditions (eg selling regularly).

13.3. Sale of Goods

Zillerium by design is global and in many countries people have no bank accounts. In some countries more than 50% of the people have no bank accounts. We propose offering various currency options for payments one being zillcoins.

13.4. Insurance Premiums

Zillerium has to arrange payment protection and this is done via a payment protection marketplace. Underwriters will fund a ring fenced fund which is then held in escrow.

Zillerium will offer the insurance services based on a marketplace and zillcoins are paid to cover insurance, and zillcoins will be converted and then used to cover claims.

13.5. Payment Processing

A fee akin to VISA fees will be charged for processing payments.

13.6. Denominations

Goodscoin - that is equivalent to the Wei in Ether. The goodscoin is 10 to minus 18 of the Zillcoin.

Denominations		
Unit	GC Value	GC
Z-Goodscoin (GC)	1	1
Z-CentralCoin (CC)	1e3	1,000
Z-DewCoin (DC)	1e6	1,000,000
Z-JumboCoin (JC)	1e9	1,000,000,000
Z-ShowCoin (SC)	1e12	1,000,000,000,000
Z-Dia	1e15	1,000,000,000,000,000
Zillcoin Z	1e18	1,000,000,000,000,000,000

We would optionally offer a fiat backed Zillcoin in later versions of Zilerium called the F-Zillcoin (ie pegged to fiat). Depending on the currency, it would then be FGBP-Zillcoin for example or simply ZILL-GBP as an abbreviation.

14. Oracles

Smart contracts, by their nature, are able to run algorithmic calculations and store and retrieve data. Because every node runs every calculation, it's not practical (and presently impossible) to make arbitrary network requests from an Ethereum contract. Oracles fill this void by watching the blockchain for events and responding to them by publishing the results of a query back to the contract. In this way, contracts can interact with the off-chain world [17].

We propose using oracles to watch for user generated events to obtain product or product pricing information. The following outlines how we use the consensus aspects of the blockchain to measure the level of value, we term the measurement of these values as trust value (TV), and price value (PV).

These values form the basic value proposition between the smart contract supplying data from the oracle and the user smart contract which requests that data.

15. Product Oracle

The purpose of the product oracle is to provide the following -

- Product Identity (PI)
- Key Product Details (KPD)
- Trust Value for Information (TV)

15.1. Product Identity

Almost all parts or products in the supply chain have a unique part number to identify them. Supply chains by their nature are broad and across the supply chain the product number is often mis-written. This can take the form of typos, abbreviations, alternatives, or just being erroneous. Therefore in practice, retailers, distributors, and other players rely on their product knowledge when placing and receiving orders. Mostly orders are placed by fax and phone. This increases operational costs and reduces the scope of expansion as knowledgeable staff are needed.

Below is an example taken from Google Shopping ([18]), a platform for advertising goods, which depends on user uploaded data, and for PM BAS C (a Bristan product), six combinations of the product number are shown: C44914, JTAkinsonSons1223, PM BAS C, PMBAS, PMBASC, PMEBASC.

Any automatic processing of sales, orders, accounts, cost analysis, will require that a product is uniquely identified and under current systems that is impossible without considerable human involvement which increases costs.

We propose an oracle which returns product identity which is then subject to a Trust Value (TV). This TV is found by measuring how data is used. Therefore a key concept in Zillerium is that we measure data usage and that has a potential reward; simply creating data has no reward under Zillerium.

To measure TV, an empirical approach is taken. Therefore if 100 users use PMBAS as the part number for PM BAS C even though that is not correct from a manufacturer standpoint, that becomes the correct number or trusted through usage.

The feedback establishing the TV can be obtained from the following means:

- Completing a sale at Zillerium
- Listing a product at Zillerium
- A call for the product data using that identity via Zillerium

A sale at Zillerium, means a user will use the product oracle to list and sell a product. At the Point of Sale (POS) the TV for the product is increased. Therefore a sale is seen as a sort of vote for the correctness of the data.

Listing a product can be used for Pay-per-click (PPC) business models and then similarly the TV is increased when someone clicks on the link.

A simple call will also increase the TV for the item.

A graduated scale will be implemented so that a sale will weight the increase in the TV for a product more than simply calling it.

We define a weight (W) which increases the TV.

$$TV = TV + W \tag{1}$$

W is worked out based on the following factors:

- Actual successful usage of the data
- Reputation of the data user
- Distribution of usage
- Affiliated Data

In version 1 of Zillerium, advanced machine learning methods ([19]) will not be used due to complexity of such methods. But a simple feedback will be used. We would weight initially weights as follows:

- W=10 when a sale completes
- W=3 for a PPC event
- W=1 for a product oracle call

We would further adjust the weight as follows:

- W=W * R where R is a normalized reputation of the user.

R would be calculated by association of trade on the platform to the user. There would not be manual voting or reviews. So if a user uploads data of a high TV then there reputation is adjusted to reflect that (a simple average which is normalized across all TVs). Later when we explain PVs then the same principle applies to the sale of goods. Users who sell well, will have their reputation enhanced.

A bonus system will pay users who have good reputations (ie token payments).

In later versions of Zillerium, we would implement more advanced feedback systems using machine learning.

15.2. Key Product Details

The KPD will vary in composition a lot for different parts. Usually product details are extensive. They can include size, weight, finishes, composition of other products, description, installation details, how to use, and a lot more. This can also include images, technical images, and other visual information.

The primary focus of Zillerium is to assist in sales. Therefore certain details are more relevant in the sales process. Usually description and size information are very important.

KPDs are assessed with PIs in the same way. It is an empirical feedback which produces a TV. The TV hence would apply to the PI and the KPD.

15.3. Data Source

Data may come from uploaded data from users (manually created data). It can also come from a TTP (Trusted Third Party) via for example an API. The Appendix lists one data return example for FDA data via the FDA API.

The users who create data or provide data via API usage are termed publishers in Zillerium. They are rewarded with zillcoins as data is used. The value of the payment is related to what the data user is paying minus a Zillerium operating fee. If the data user pays P for using the data, and there is a cost to get the data C then the gross profit (GP) in Zillerium is:

$$GP = P - C \tag{2}$$

Zillerium charges a fee F (normally a small percentage), and hence the publisher commission (PC) is:

$$PC = GP - F \tag{3}$$

The value of P itself is determined by a marketplace. Prospective data users will see a list of possible data sources and a cost to using them. They will decide on what to use and then pay in Zillcoins. The publishers will set their own minimums which will factor into the PC. Other than that a normal market value would be obtained.

The TV is used to determine the value of data and not the origin. We do not distinguish between a TTP data source and data uploaded by a user. All the values are empirically calculated as outlined in this document.

When the user calls the product oracle, the user can specify a data source, eg FDA for drug related data. But that does not improve the TV for the data by itself.

Manually uploaded data will be stored in IPFS [3]. IPFS is a content-addressed file system and the hash of the data is the content address.

Below is an example of a manually created record in IPFS. The publisher's details are stored in IPFS to show their key user information.


```
 {"product": {
  "partnumber": "36292000",
  "brandname": "Grohe",
  "description": " F-digital Veris F-Digital Digital controller
  for bath or shower",
  "publisher": "0xb5a50d5fb5bD377dd84e6D39c65F076c75081782",
  "currency": "Zill"
  "price": "market"
  "components": {
    "compbrandname": [
      {"compdesc": "MOTOR", "comppn": "42337000"},
      {"compdesc": "retaining ring", "comppn": "47765000"},
      {"compdesc": "Thermostatic compact cartridge 3/4",
      "comppn": "47881000"}
    ]
  }
}}
```

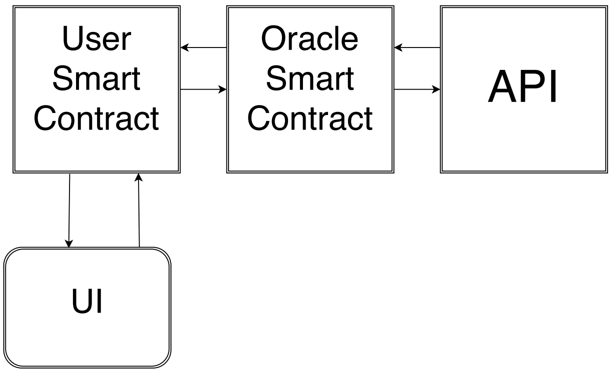
15.4. Storing Trust Values

Data comes from a number of different resources, but we need to store a ledger of TVs and publisher reputations.

IPFS will be used to store a lot of key data and in cases of confidential data, the data may be encrypted.

We also need a ledger of payments due to publishers. When data is available but not used, we would keep any records in a permissioned based blockchain called Zillerium. This would hold the publisher IPFS address, and the content-address data. As zillcoins are earned, the ledger in ethereum will be updated for the publisher's zillcoin payment.

Also the TV for the product identifier will be stored in Ethereum, the cost of this will be financed by the payment for acquiring the data. If the cost of using Ethereum is too high, then aggregated values will be written into Ethereum and a Zillerium permissioned based blockchain will be used to store interim TVs.



16. Pricing Oracle

Zillerium will implement a pricing oracle. This is based on market pricing as determined by Zillerium's own implemented marketplace for product sales.

As with product information, we produce a Price Value PV to determine the value of the price. As prices vary with time, the PV for price will have a time component. Usually product details do not change with time.

Once a product is listed on the Zillerium marketplace, the item can have the following events:

- Listed but not seen
- Listed viewed and not more action
- Listed, viewed, added to cart, but no further action
- Listed, viewed, added to cart, purchased, no refund
- Purchased and then refunded

If the user buys the item that is seen as statement that the price was a market one, otherwise not. A refund is interpreted as a price being against a market value during the sale. Hence we have:

- PV = +10 for a purchase

- PV = -20 for a refund
- PV = -3 for a abandoned cart
- PV = -1 viewed and not more action

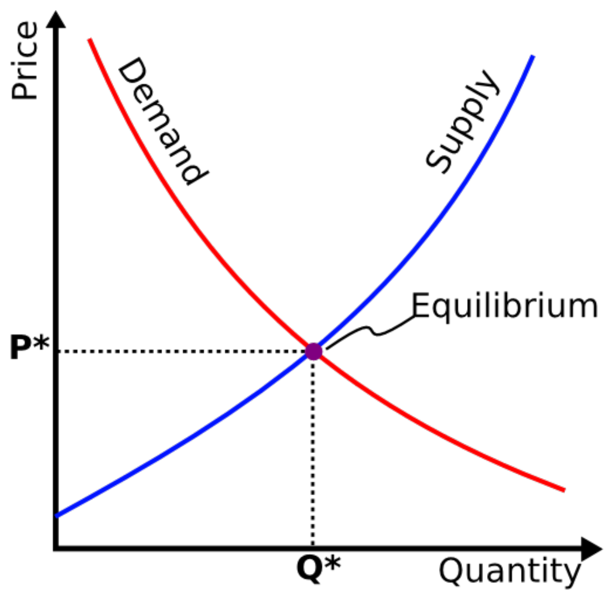
PV is then aggregated to get a total based on a time period τ, which is measured in terms of market conditions. Some goods are fast moving (FMCG) and a PV will change every few minutes, other goods are very slow moving (furniture) and a PV might not change for weeks or months.

We want to find the stable price for that market and hence on a standard supply and demand curve we have a supply price and a demand price, with an equilibrium shown. When goods are not sold that implies the supply and demand characteristic is not at an equilibrium point. If we take the highest PV for a price then that is most likely to be an equilibrium price under market conditions. If this is termed P_m for market price, at time T_i then we can observe the market prices over time and their variation. If these market prices are plotted for a time window τ from T_1 to T_{10} the dynamic of the price would be clear.

It would be a horizontal line for a stable price and not one which is unstable there would be a scatter type of diagram. Many prices would be expected to rise over time.

We would calculate the average price for the period τ and work out the standard deviation from the mean. A new market price would then have a certainty value based on previous values. If the price is more than 1 standard deviation from the average then the price would have a weight applied of 50% and more than 2 sd would apply a weight of 0.1 to the PV: w=.5 when P>1sd, w=.1 when P>2sd, P is the price.

$PV = PV * w$



17. Risk Oracle

The purpose of this oracle is to provide risk assessments based on proposed sales transactions. We would in effect record risk categories. These would be fairly broad relating to country; price band; reputations of the buyer and seller; and product type.

Underwriters would be presented with proposed transactions and they would state the price at which they would insure the transaction, which would lead to Zillerium classifying types of risks.

Hence a typical transaction might list a computer in a certain country and a certain price. The underwriting marketplace could classify that in a risk band by the bidding. Many aspects of the bidding would be automated. Also as the system acquires more data, the number of refunds would be recorded.

Hence a underwriter would see the number of likely refunds for a certain set of conditions.


```
{"risk": {
  "product type": "computer",
  "sellerReputation": .5,
  "buyerReputation": .5,
  "country": "Zambia,
  "priceband": .8,
  "previousClaims": 0.01,
  "risk": 0.7

}}
```


The reputations are generated based on the details stated already in this document. The country and product type would come from the Product Oracle. Priceband is worked out based on what else is for sale in that class of products, previousClaims is the normalised number of claims and the risk is an assessment based on the likelihood of a refund. This is worked out empirically based on previous claims.

When Zillerium has little data to make a risk assessment, then the risk will default 1 and will be ignored in decision making.

The risk assessment data would be stored in elasticsearch via a cloud solution. The reason for this is that searching is needed. Also it is likely that some Big Data methods would apply such as with HDFS, Hbase, and Hadoop. This does not dilute the decentralised concept because the decentralization refers to market dynamics and insurance risk cannot logically be put into a blockchain until search technologies are advanced enough.

We have to for risk assessment potentially engage a number of searches.

18. Comparison to Alternatives

Major sites such as ebay, Amazon, Google (Google Shopping) are far reaching. But these sites all depend on seller input. This is fundamentally biased under theories related to information asymmetry (the seller knows more than the buyer and uses than advantage against the buyer - this fundamentally damages the market).

What Zillerium does is enable publishers to publish data about products and this can be extended to data about product deals (ie special offers). This combined with a trust value system which we have explained, will reduce or remove bias in the sales process.

Also major sites such as those mentioned are governed by a small management teams at the board level. They adopt American values. Zillerium by its nature localized and not dependent on any one culture or country. The allows considerable growth in regions which presently are not served by major American brands.

Ventures such as OpenBazaar have entered into the sales of goods market but their perspective is entirely different to the one held at Zillerium. Zillerium is about market dynamics not the sales of goods. OB is mainly concerned with connecting a buyer to a seller. We are concerned with establishing market conditions and then enabling growth.

At the time of publication no venture is fundamentally addressing as a project which can be implemented the issues raised by Zillerium.

19. Summary

This document has outlined how an architecture could be built to measure value of product, pricing, and risk information via data usage. We have described how data itself can be obtained via oracles and valued data is rewarded to the publisher with a zillcoin payment.

If implemented, Zillerium would allow significant growth in the economy as it would address market instability problems. This is done by decentralizing publishing, risk, and pricing formation. Such an action would allow a far greater transparency which would apply supply and demand economics to prevail to form a stable and expanding market.

We would implement decentralization via a permissioned blockchain (Zillerium), and a permissionless one (Ethereum) which would confirm trust values, price values, and risk values.

We also address the unsolved problem of payment protection for sales of goods transactions via decentralized underwriting. We solve liquidity issues by placing insurance related funds into an off-chain ring fenced account which is regulated. Zillerium then would settle transactions by issuing zillcoins and any liabilities transferred to the general fund to be settled at the end an accounting period.

References

[1] Akerlof, George A. ”The market for” lemons”: Quality uncertainty and the market mechanism.” The quarterly journal of economics (1970): 488-500.

[2] Lofgren, Karl–Gustaf, Torsten Persson, and Jorgen W. Weibull. ”Markets with asymmetric information: the contributions of George Akerlof, Michael Spence and Joseph Stiglitz.” The Scandinavian Journal of Economics 104.2 (2002): 195-211.

[3] Benet, Juan. ”Ipfs-content addressed, versioned, p2p file system.” arXiv preprint arXiv:1407.3561 (2014).

[4] http://growthcrossings.economist.com/wp-content/uploads/sites/47/2017/05/eiu-growth-crossings-treasury-pillar-briefing-paper-final-may-2017.pdf. Rebooting Supply Chains.

[5] http://www.fao.org/save-food/resources/keyfindings/en/. FAO of the UN.

[6] Nobel Prize in Economics 2016. https://www.nobelprize.org/nobel_prizes/economic-sciences/laureates/2016/popular-economicsciences2016.pdf

[7] Rochet, Jean-Charles, and Jean Tirole. Defining two-sided markets. mimeo, IDEI, Toulouse, France, January, 2004.

[8] Lee, Hau L., Venkata Padmanabhan, and Seungjin Whang. ”Information distortion in a supply chain: The bullwhip effect.” Management science 43.4 (1997): 546-558.

[9] Supply Chain 4.0 in consumer goods By Knut Alicke, Daniel Rexhausen, and Andreas Seyfert. April 2017. McKinsey & Company.

[10] Phlips, Louis. The economics of price discrimination. Cambridge University Press, 1983.

[11] https://blockchainhub.net/blockchain-oracles/. Oracle.

[12] Doyle, John, and Gunter Stein. ”Multivariable feedback design: Concepts for a classical/modern synthesis.” IEEE transactions on Automatic Control 26.1 (1981): 4-16.

[13] Buterin, Vitalik. ”Ethereum white paper.” (2013).

[14] Nakamoto, Satoshi. ”Bitcoin: A peer-to-peer electronic cash system.” (2008): 28.

[15] Zeleny, Milan. ”Autopoiesis and self-sustainability in economic systems.” Human Systems Management 16.4 (1997): 251-262.

[16] Barratt, Monica J., and Judith Aldridge. ”Everything you always wanted to know about drug.” Science International 251 (2016): 87-94.

[17] https://medium.com/@mustwin/building-an-oracle-for-an-ethereum-contract-6096d3e39551.

[18] Bowman, Robert. ”Will Google Shopping Express Help Retailers Fend Off Challenge from Amazon?.” Forbes, June 7 (2014).

[19] LeCun, Yann, Yoshua Bengio, and Geoffrey Hinton. ”Deep learning.” Nature 521.7553 (2015): 436-444.

[20] Jia, Bokang, et al. ”Opus-Decentralized music distribution using InterPlanetary File Systems (IPFS) on the Ethereum blockchain V0. 8.3.” (2016).

[21] Katsh, Ethan Ethan, M. Ethan Katsh, and Janet Rifkin. Online dispute resolution: Resolving conflicts in cyberspace. John Wiley Sons, Inc., 2001.

[22] Friedman, George H. ”Alternative dispute resolution and emerging online technologies: Challenges and opportunities.” Hastings Comm. Ent. LJ 19 (1996): 695.

[23] Briggs, Adrian. Agreements on jurisdiction and choice of law. Oxford University Press, 2008.

[24] http://www.investopedia.com/terms/e/ecommerce.asp Electronic Commerce - ecommerce

[25] https://www.ons.gov.uk/businessindustryandtrade/itandinternetindustry/bulletins/ecommerceandictactivity ONS.

[26] https://info.shapeshift.io/

20. Appendix

20.1. FDA Data


```
{
  "meta": {
```

```
    "disclaimer": "Do not rely on openFDA to make decisions
    regarding medical care. While we make every effort to
    ensure that data is accurate, you should assume all
    results are unvalidated. We may limit or otherwise
    restrict your access to the API in line with our
    Terms of Service.",
    "terms": "https://open.fda.gov/terms/",
    "license": "https://open.fda.gov/license/",
    "last_updated": "2017-09-08",
    "results": {
      "skip": 0,
      "limit": 1,
      "total": 15049
    }
  },
  "results": [
    {
      "effective_time": "20110818",
      "inactive_ingredient": [
        "INACTIVE INGREDIENTS Sucrose/Lactose"
      ],
      "keep_out_of_reach_of_children": [
        "KEEP OUT OF REACH OF CHILDREN Keep this and
        all medicines out of reach of children."
      ],
      "purpose": [
        "USES To relieve the symptoms of itching."
      ],
      "warnings": [
        "STOP USE AND ASK DOCTOR If symptoms persist/worsen
        or if pregnant/nursing, stop use and consult
        your practitioner."
      ],
      "spl_product_data_elements": [
        "Mezereum DAPHNE MEZEREUM BARK DAPHNE MEZEREUM
        BARK DAPHNE MEZEREUM BARK SUCROSE LACTOSE white"
      ],
      "openfda": {
        "product_ndc": [
          "68428-511"
        ],
        "is_original_packager": [
          true
        ],
        "package_ndc": [
          "68428-511-12",
```

"68428-511-03",
"68428-511-06",
"68428-511-05",
"68428-511-11"
],
"generic_name": [
"DAPHNE MEZEREUM BARK"
],
"spl_set_id": [
"00006ebc-ec2b-406c-96b7-a3cc422e933f"
],
"upc": [
"0740640391006"
],
"brand_name": [
"Mezereum"
],
"manufacturer_name": [
"Washington Homeopathic Products"
],
"unii": [
"X2N6E405GV"
],
"spl_id": [
"59a7a4be-8303-4e81-8f0a-5e5751f5c47f"
],
"substance_name": [
"DAPHNE MEZEREUM BARK"
],
"product_type": [
"HUMAN OTC DRUG"
],
"route": [
"ORAL"
]
},
"version": "1",
"dosage_and_administration": [
"DIRECTIONS Adults: Dissolve 3 to 5 under the tongue three times a day or as directed by Lic. Practitioner. Take at greater intervals as condition subsides. Children: Dissolve 3 to 5 under the tongue three times a day or as directed by Lic. Practitioner. Take at greater intervals as condition subsides."
],
"package_label_principal_display_panel": [

```
        "PRINCIPAL DISPLAY PANEL The OTC potency range of
        MEZEREUM is 2x{30x, 1c{30c, 200c, 1m, 10m, 50m, and CM.
        Availability is subject to change. All WHP single remedies
        are made to order; thus, the labels are printed on the
        same label stock as the orders are filled. 'Bottle Size'
        and 'Potency' vary on the label depending on customer
        choice. Standard bottle sizes for pellet-form remedies
        are 2 dram, 4 dram, 1 ounce, 2 ounce, and 4 ounce.
        Mezereum label example"
      ],
      "indications_and_usage": [
        "INDICATIONS Indications: MEZEREUM Itching"
      ],
      "set_id": "00006ebc-ec2b-406c-96b7-a3cc422e933f",
      "id": "59a7a4be-8303-4e81-8f0a-5e5751f5c47f",
      "active_ingredient": [
        "ACTIVE INGREDIENTS MEZEREUM"
      ]
    }
  ]
}
```


20.2. Grohe Example

Data Load Example

As an example of a product, the following is given - Grohe 36292000. Grohe is a large global brand in bathrooms and the wireless product is one of their more complex products. This example shows how parts are listed for the main product, and the present data structure involves 2d and 3d images, lists of features, installation requirements, required products, a general description, and compliance details.

This example shows the complexity of data for a single product from a single brand in a single market sector.

The description is - F-digital Veris F-Digital Digital controller for bath or shower - part number 36292000 and an EAN 4005176894640,

Components		
1	MOTOR	42337000
2	retaining ring	47765000
3	Thermostatic compact cartridge 3/4	47881000
4	Solenoid valve	42340000
5	Non-return valve	47753000
6	Digital Controller & Remote Controller	36309000
7	Connection wire	47727000
8	Transceiver	36356000
9	Plug power supply 230 V	65790000
10	Plug power supply 110-240 V	36078000
11	Uninterruptible power supply	36394000
12	Connection wire	65815000
13	Socket Spanner	19332000
14	Holder plate for Digital Controller	40710000