Prometeus Network:
A secure and decentralized infrastructure for personal data markets, enhanced with artificial intelligence

White paper (rus) ver: 0.12 Previous version
October 16,2018

Abstract

Personal data laws force social networking websites to enable users to manage their own data. Facebook [4], Google [5], Instagram [6], and Twitter [7] are now offering services with which data can be directly downloaded. Moreover, Facebook, Google, Microsoft, and Twitter have recently launched the Data Transfer Project [4] (a white paper [5]) this makes it easier for users to connect their different social network accounts.

Prometeus Network is developing an infrastructure system wherein data science algorithms are employed to process private data, making it possible for users- without risking data integrity or violate privacy- to safely sell their personal data to interested parties.

Motivation

The use of Prometeus Network is not limited to a specific kind of industry use case , but for purposes of this document we knowingly and deliberately choose not to go beyond the discussion of disrupting data exchange needs related to “Influencer Marketing” . For the purpose of simplicity and ease of launch and due to similarities between different use cases, we will -in the remaining of this white paper- discuss Prometeus Network only in the context of Influence Marketing .

1.1. API limitation by social networking websites

Social networking websites seek to make the use of their data as difficult as possible by curbing APIs all the way down to make them complete uselessness [1][2][3]. General Data Protection Regulation and the Facebook–Cambridge Analytica data scandal further accelerate and justify these limitations.

1.2. Influencers focused on a dozen of platforms

There are now numerous social networking websites, video hosting services, and personal blogging platforms. The key platforms for Influence Marketing are: Instagram, YouTube, Twitter, Twitch, Weibo, WeChat. An effective Influence Marketing campaign necessitates analyzing a blogger's audience on every platform by interest group, roughly

estimating engagements per post, assessing bot percentages and audience "ranking boost," and so on, and so forth.

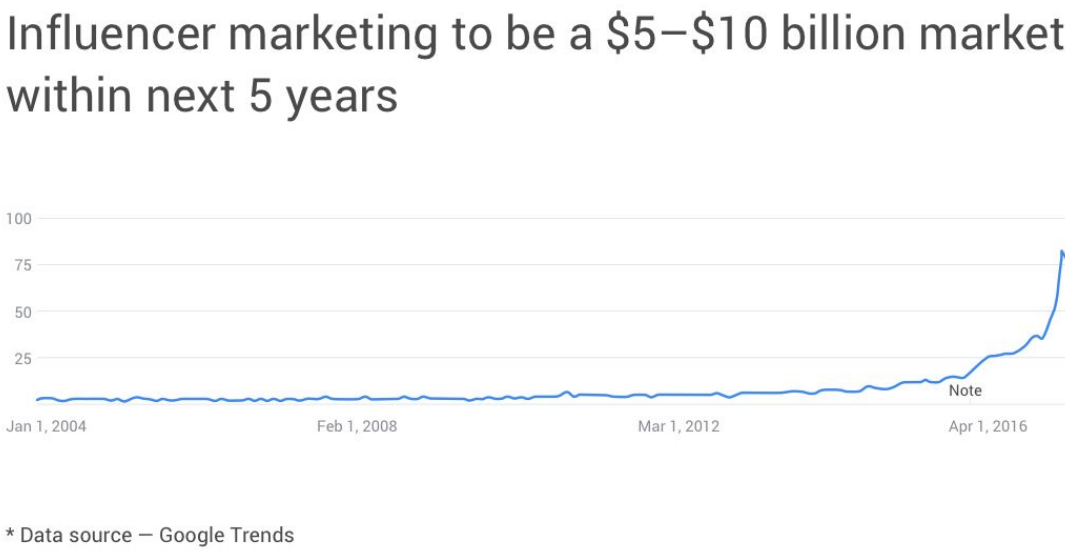
This being said, as described earlier, platforms do not allow third-party services to receive data about influencers, not to mention their audience.

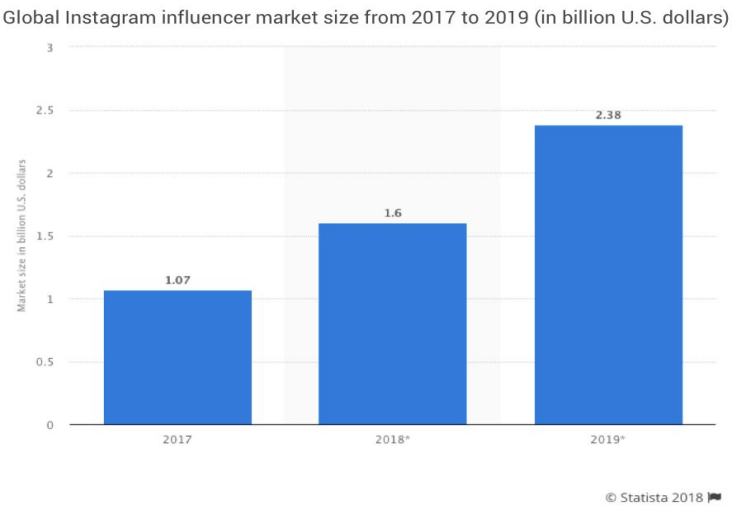
1.3. Any attempt to develop a transparent centralized tool is destined for failure

There are currently centralized Influencer Marketing campaigning services. However, amendments to the Terms of Use of social networking websites, closing off access to the APIs of social networking websites, legislative changes (GDPR), and technical issues associated with creating a blogger-friendly method for collecting and processing data from numerous social networking websites cast doubt on the future of these services. The fact is that any transparent analysis, matchmaking, and communication between advertisers and influencers will sooner or later put legal and financial pressure on corporate platforms [1][2][3][4][5][6][7].

2. Industry Overview

Influencer Marketing is today one of the fastest-growing advertising markets. Analytical reports show an increase in almost all metrics related to advertising through the blogosphere. Assuming rapid growth, this market will become a $5–$10 billion industry in the next five years, according to recent estimates by Mediakix.





The average monthly U.S. Google searches for "influencer marketing" stand at about 10,000, three times higher than in 2016. This shows the interest in influencer marketing is getting stronger and growing rapidly.

Further, according to Statista, the market value of Instagram influencers alone will increase to $2.38 billion in 2019.

Given the existing metrics and research on Word of Mouth marketing and confidence in Influencers, Influence Marketing on social networks is becoming the number one advertising channel. Despite the large number of Influence Marketing campaigning differences , more than half of the reported companies talk about plans to increase Influence Marketing budgets for the coming year.

According to research, 86% of marketers employed Influence Marketing in 2016, of whom 94% found this advertising format effective. 67% of marketers are forecasted to increase their Influence Marketing budgets compared to only 4% of the marketers planning to decrease their budgets by late 2017. The most effective social media in terms of Influence Marketing use include both Instagram and YouTube. Influencer blogs were found to be an influential channel, too, with 87% of marketers leveraging them as a tool for improving 'discovery' and 'brand image'. Today, the average budget for an advertising campaign varies somewhere between the $50,000 and $100,000, analytics agencies forecast an 18% growth in the segment of advertising campaign budgets which are between the $100,000 and$500,000,.

Judging from an eMarketer report, Influence Marketing is more effective in areas such as: buyer goods, tourism & leisure, personal care, alcoholic products and beverages [1]. The brands using Influence Marketing claim to generate an average ROI of $12.21 in on each dollar spent. Furthermore, research shows that 92% of the buyers are prone to trust the advertisements posted by opinion leaders. Accordingly, from the standpoint of ROI, Influence Marketing has proven to be more effective than traditional online advertising, while allowing brands to have a stronger impact on their audience. That is why, Influence Marketing is now the most effective and long-lasting marketing strategy utilized.

Instagram alone has more than 50 million influencers (accounts with at least 1,000 followers). Accounts on particular topics are nearly impossible to find without the use of special search tools.

Moreover, when posting information through an influencer, the advertiser primarily wants to reach its target audience. However, reaching the target audience is not guaranteed; often less than 10% of an audience represents an advertiser's target audience, thus causing up to 90% of the budget to be wasted.

The leading development platforms in the Influencer Marketing sector receive investment in order to create products that would allow finding and conducting campaigns involving influencers with the aim of reaching an advertisers target audience.

The list below gives examples of the investments received by platforms:

- **Maker Studios** – Acquired by Disney for $950M, $65M of total funding (https://www.crunchbase.com/organization/maker-studios)
- **Lithium Technologies** – Acquired by Vista Equite Partners for a total of $1BN, $201.7M of total funding (https://www.crunchbase.com/organization/lithiumtechnologies)
- **Klout** – Acquired by Lithium Technologies for $200M, a total of $40M in funding (https://www.crunchbase.com/organization/klout)
- **TapFluence** – Acquired by IZEA, the total funding is $22.7M (https://www.crunchbase.com/organization/tapinfluence)

- **FameBit** – Acquired by Google for an undisclosed amount (https://www.crunchbase.com/organization/famebit)
- **Niche.co** – Acquired by Twitter for an undisclosed amount, the total funding is $3.1M (https://www.crunchbase.com/organization/the-niche-project-inc)
- **FullContact** – a total of $55.1M in funding (https://www.crunchbase.com/organization/fullcontact)
- **Tubular Labs** – $31.8M of total funding (https://www.crunchbase.com/organization/tubular-labs)
- **BrandWatch** – $64.7M of total funding (https://www.crunchbase.com/organization/brandwatch)
- **Influential** – the total funding is $36.5M (https://www.crunchbase.com/organization/influential)
- **IZEA** – a total of $34.9M in funding (https://www.crunchbase.com/organization/izea)
- **Pipl** – the total funding is $19M (https://www.crunchbase.com/organization/pipl)
- **Mavrck** – $12.9M of total funding (https://www.crunchbase.com/organization/mavrck)
- **Traackr** – a total of $12.7M in funding (https://www.crunchbase.com/organization/traackr)
- **Takumi** – the total funding is $11M (https://www.crunchbase.com/organization/takumi)
- **HYPR** – $8.0M of total funding (https://www.crunchbase.com/organization/dollarsocial)
- **NeoReach** – a total of $4.3M in funding (https://www.crunchbase.com/organization/neoreach)
- **Open Influence** – the total funding is $4M (https://www.crunchbase.com/organization/instabrand)
- **Klear** – $2.9M of total funding (https://www.crunchbase.com/organization/twtrland)

3. Proposed solution

Our solution involves a decentralized and censorship-resistant distributed database (blockchain) with a well-thought-out and weighted system of incentivizing participants for providing/updating data, inviting new participants and a market-determined free access to the data for anyone wishing to have so. With encryption mechanisms enabling secure personal data storage and sale.

Influencers will be offered:

1) The opportunity to create their own secure blockchain identity and upload here data from all their social media accounts. Data can be sourced from any social networking website by:
    a) uploading under the GDPR – users' own data upload cannot be banned;
    b) downloading the screenshots of audience data from a social networking website – internal audience statistics are available to any blogger in social media applications. Social networking websites are unlikely to close off their users' own account audience analysis as it is an essential part of user analytics. The problem of trust is solved by our encryption mechanism, data cannot be stored in the system other than in an encrypted form.
2) The opportunity to sell their data;
3) The opportunity of using the system to promote their own social media identity and to reach fellow bloggers for mutual advertising proposals.

Advertisers, agencies, and platforms will be offered:

1) The opportunity to search for suitable bloggers through the use of advanced filters that are not available on social networking websites or which can be obtained from aggregators;
2) The opportunity to obtain in depth information on a blogger with insights like: Audience Credibility, Paid Post Performance, Lookalikes, Interests etc;
3) The opportunity of exchanging/selling data concerning their contracted bloggers while securing an extra channel of customers.

Data and AI Scientists will be offered:

1) The opportunity to earn money while enriching a blogger's data;
2) Access to the social networking websites enormous data sets with the aim of improving algorithms, possible uses are:
    a) The findings made by other Data and AI Scientists can be used to test and improve own algorithms;
    b) Purchasing a huge amount of unnecessary data at low prices in order to teach own algorithms;
    c) In the future in the adjacent markets, e.g., health care when a hospital or other Data and AI Scientists will sell big amounts of data to third parties at low prices for educational purposes.

4. Why Blockchain?

Centralized systems that deal with Influence Marketing data have a number of disadvantages:

1) They are liable for any legal issues which may arise in the event of amendments to the Term of Use of social networking websites, legislative changes (GDPR), or data leaks. It means that the system is broadly vulnerable and susceptible to censorship, pressure, and prosecution by corporate and government entities, possibly causing system outages or complete shutdown.

2) They manage pricing throughout the entire system (fees, discounts, service price, and VIP access) and compete only with similar centralized solutions. This can cause underpricing for bloggers (all the way down to zero since advertising is the main source of income for a blogger) and overpricing for buyers.

3) All data collection rules and data handling models represent the so called "black box" s of such systems. Where this kind of approach is used, data owner or enricher do not know who purchased their data, when it was purchased and how many times or whether or not the data is perceived to be true and accurate.

4) They are aimed at increasing profits for the system itself, resulting in a lack of incentivizing mechanisms allowing other participants (bloggers, enrichers, and buyers) an honest remuneration.

In decentralized systems, all rules and provisions are described through the use of smart contracts in a transparent and readily understandable format. These systems are designated to strictly comply with the rules described in smart contracts and all of the system activities are readily trackable by the system participants.

History and transparency of all transactions

All transactions in Prometeus will be traceable and trackable and their history will be viewable.

In centralized systems, bloggers do not know who purchases their data and the buyer does not know where the blogger's data in the system comes from. This stems from the reluctance of system owners to disclose how they use data and, besides, necessitates solving legal issues related to data transmission.

All transactions as well as their history in Prometeus are fully transparent due to the peculiarities of blockchain technology. All Validators, Enrichers, and DataMarts will have a public ID (e.g., Validator – Best Influencer Data). Consequently, which enricher processed data, how many times, when it was updated, which DataMarts bought data, which validator provided it – all of these histories constitute transparency and is available to all system user.

Public ranking

All validators and enrichers will be prone to an automatic transparent ranking system.

The Data Owner's reputation is very often unknown in centralized systems and can be boosted or falsified even if the identity of the Data Owner is public. Reputation is difficult or almost impossible to verify due to its private status, making it hard for Data Buyers to choose the right Data Owners from where to purchase the data.. The Buyer has to purchase data from multiple Data Owners and compare the data quality between them but, even then, the Buyer has no guarantee of the subsequent data updates having the desired quality.

Reputation is computed in Prometeus by a smart contract in automatic mode and is based on the number of sales, the number of resales, data update frequency and many other factors, thus rendering reputation boosts uneconomic (see a detailed description below). Furthermore, reputation is for reference only and does not affect any internal system processes. Reputation can help system users assess each other, making it easier to make validator and enricher selection.

Data loss prevention

Centralized systems are prone to the risk of data being copied but not purchased. The reason is that data is a special type of commodity whose value pays off once accessed. Data is easy to copy, alter, or transmit. These peculiarities make data protection an important problem for centralized systems. But good protection cannot be ascertained by the user, even if available, since it is a closed system.

All data will be encrypted in Prometeus using the Data Owner's key. In order to protect data from copying, the Validator creates open metadata (sample and description) with the aim of allowing the buyer to assess the commodity before making a purchase without the need to view data itself. Where a transaction is executed, a smart contract warrants that data cannot be decrypted until the purchase is made.

5. Product description

Our objective is to develop a fully decentralized open system for a win-win data exchange between influencers and advertisers that is beyond the control of corporate giants like Facebook and would be resistant to possible administrative or legal pressure on the part of corporations. This makes Blockchain technology the perfect fit.

5.1. High-level system description

Let us discuss system participants and components in the same order as the data flows through the system from data owner to the buyer:

5.1.1. Data Owner/Data Provider

Data can be provided/delivered by the data owner/user directly as well as by various aggregators or data exchanges. Typically, the Data Owner does not upload data directly to the system, but rather operates through the Validator (as described below).

The Data Owner, alone, sets the price for their own data, i.e., the fee the Data Owner wishes to receive per sale. The Data Owner's identity is recorded in Prometeus alongside the data, and the fee is transferred at the time of sale either directly to the Data Owner or first to the Validator who later arranges a settlement with the Data Owner.

The main method of capturing data from Influencers is by Influencers uploading their own personal data from online account to Instagram/Facebook. The upload represents a zip archive containing data in a machine-readable JSON format. This archive is uploaded by the Influencer to the Validator's interface via mobile application or website. System architecture, however, does not clearly limit data type or size – all what matters is demand for any given data format. Possible scenarios include the Data Owner downloading the screenshots from Youtube analytics or uploading data from Facebook or even data from devices and not people – what matters most is their being demand for the data . If so, Enrichers and DataMarts will follow.

5.1.2. Validator

Validator is the blockchain data storage ingress gateway, and it fulfills three roles:

1) Arranging a convenient gateway to receive/make payments in order to work with Data Owners. This kind of gateway offsets a speculative component of token economics and makes life easier for all system users, enabling them to make settlements in fiat currencies. That said, the token remains the basic intra-system settlement unit;
2) Verifying data ownership.
3) Incoming data quality control in order to prevent the system from being penetrated by spam, incorrect or outdated data;
4) Arranging a user-friendly interface for data collection from Data Owners.

All of the Validator's preliminary work is off-chain work. For example, if a popular blogger wishes to sell its data, he or she enters into an agreement which or

undergoes a formal authentication/KYC procedure. Note, data does not enter the system until all verifications and validations are completed. When uploading data to Prometeus, the Validator sets its data price, with the difference between the Validator's price and the price set by the Data Owner constituting the Validator's service fee.

Validator is not the only possible entry point; theoretically, Data Owner can act as Validator and record data in the blockchain, but we rely on reality and the reality is influencers are unwilling to become knowledgeable about blockchain – all they want is a "one-button" interface and payout in a convenient currency. Validators act as these interfaces.

The Validator should be incentivized to deliver only quality data that would be in demand among Buyers. Validators are encouraged in two different ways only to deliver quality data :

- the Validator pays a small fixed fee for every downloaded piece of data. Accordingly, if the data is not purchased its price is eventually lower than the paid fee (as a result of temporary discounting), the Validator will lose its money;
- being a Validator requires N tokens to be deposited. The size of the deposit depends on the total number of tokens in circulation. Random users without a deposit are thus left locked out.

Agencies can act as Validators, too. "Agencies" refer to companies offering Influencer Marketing campaigning services and who work directly with bloggers. These companies have a vast network of Influencers and can encourage them to download and update data in Prometeus. Bloggers uploading their data directly can also assume the role of Validators – with this in mind, we will develop a number of free mobile applications (iOS App, Android App, and website), with the lowest possible system fee being charged.

5.1.3. Enricher

Advertisers have relatively limited interest in any raw data received from the Influencer as it mainly contains publicly available information. The value of data grows considerably after *enrichment*, which provides new insights emerging from data analysis conducted by AI and other additional external sources. Examples of new insights include the interests, paying capacity of the Influencer's audience, bot / ranking boost percentages, etc.

The role of Enricher is introduced because of the following reasons:

1) Most of the potential Validators and Data Owners in the Influence Marketing segment do not have any resources or competencies in place when it comes to conducting in-depth data analysis through the use of AI – we reached this

conclusion after four years working in this market with over 44,000 customers (each being a potential Validator);

2) To be able to enrich data, the Enricher needs the data required for neural network training and other machine learning algorithms. Not a single Validator has this amount of data in place, necessitating the data of Validators being clubbed together for algorithm learning purposes. This is prevented by a lack of mechanisms guaranteeing no data will be stolen by the Validator which is responsible for creating these algorithms. In our case, this kind of mechanism will be represented by the Enricher. Interactions with Validators (see a detailed description below), data transmission for AI training/learning purposes, the Enrichers' fee, data integrity will all be guaranteed as part of Prometeus ;
3) Validators are not interested in helping their peers. Consequently, the model in which data enrichment is handled by no one but Validators themselves can cause algorithms used by one Validator to be unavailable to others, affecting the overall condition of the system. Enrichers are specifically designated to avoid this kind of situation – they are separated from Validators and can work with an unlimited number of Validators. Enrichers need data (to improve their algorithms) and the sale of enrichments, while Validators need enrichments to attract customers for their data – the honesty and transparency of this mutually beneficial cooperation is guaranteed by Prometeus.

The role of Enricher can be carried out by AI and Machine Learning companies and data scientists that previously provided these services only to platforms.

5.1.4. DataMart

The objectives of DataMart are opposite to those of the Validator, i.e., DataMart is responsible for selling and uploading system data to the Buyer. The roles of DataMart are as follows:

1. Providing Buyers with a user-friendly interface in order to search for and select Influencers fitting their needs and to pay for the data received.
2. Possibly also providing Buyers with an interface via API
3. Packing Influencer's data in any form which the Buyers desire (HTML page, PDF report, etc.)
4. Settle payment(s) with Buyers using fiat money.

It is DataMart that sets the price for Influencers' data.he difference between the price set by the Validator (and Enrichers) and the price at which data is sold to the Buyer constituting DataMart's income. If the Buyer uses fiat money, DataMart starts

from purchasing data, paying full price, and then has its expenses offset by funds from the Buyer.

The role of DataMart can be carried out by Influence Marketing platforms. “Platforms” means here companies that allow customers to search for a blogger independently while enabling bloggers to sign up and search for advertisers. The platforms also offer blogger selection services to major brands. Their main difference from agencies is that platforms work with bloggers in automatic or semi-automatic mode. These companies have resources for integration with third-party services in order to receive additional information about bloggers and a pool of customers which need this particular set of data.

5.1.5. Data Buyer

This role is typically carried out by advertising agencies or individual brands conducting their own advertising campaigns, other options include for example Influencers looking for fellow bloggers and conduct a shared advertising and audience exchange. In principle, the Buyer can purchase data directly and not through a DataMart, but this would require substantial technical efforts since selecting Influencers in the blockchain data sets is not an easy task.

5.2. Internal competition

Any number of Validators, Enrichers, and DataMarts can be connected to the system, all of which operate independently and compete for Data Owners and Buyers. We believe that competition thus improves the overall quality of the system’s operations and increases resistance to the negative impacts of corporations (Validators and DataMarts work with fiat money and should therefore have all relevant legal requirements in place).

Likewise, any node can combine the roles of Validator, Enricher, and DataMart at its sole discretion.

5.3. Protection from bad actors

The main threat is the emergence of unscrupulous Validators and Enrichers that can upload large amounts of fake or poor-quality data to the system hoping to profit from random sales.

Bad actors do not pose a major threat to the regular Buyers purchasing data in industrial-scale amounts as such buyers have already gained some experience and purchase data only from “tried and tested” Data Owners. Bad actors can, however, present a serious problem for new Buyers as well as those purchasing data occasionally,

resulting in financial and time loss associated with purchases from “wrong” actors; likewise, they can leave new Buyers with a negative impression of the system.

Quality control using administrative methods (ban on bad Validators and Enrichers) is impossible in a decentralized open system. In addition, there are no objective data quality criteria – different Buyers need different data covered by different requirements. It therefore seems more reasonable to allow Buyers to rank quality by voting with their feet. . In other words, it is the market that ranks quality – the higher the sales, the better the data quality. We believe this sort of ranking being absolutely impartial, independent, and immune to manipulation since it reflects the dealings of most market participants.

Our objective is to formalize tokenomics and public quality ranking rules summarizing all of the above. This ranking will help inexperienced Buyers select data owners.

5.3.1. Ranking

Ranking reflects the standing of Validator, Enricher and DataMart in Prometeus and it is based on the feedback from Data Buyers. Ranking is determined based on following rules:

- Ranking is proportional (not necessarily linearly) to the volume of data sold. Volume is expressed in tokens, i.e. data price;
- Returning customer principle; roles with a high number of returning customers will receive a higher ranking compared to those roles benefiting mainly from one-time purchases even though the *volumes sold* are similar. We assume here that customers who keep buying data on a regular basis from a single source value the data as ‘quality data’;
- High volume Data Buyers will have a significant higher impact on rankings compared to those who buy lower volumes, because we assume Data Buyers who buy high volumes of data will have the experience and resources to value data correctly and therefore should have a higher weight;
- The distributions of aggregate sales and purchases among market participants typically follow a power law; if factored in directly, the volumes of major participants can therefore have a disproportionately large impact on a ranking (see also Nassim Taleb’s Extremistan). Volume must therefore be factored in nonlinearly.
- The effect of older data purchases on the ranking of stakeholders must be gradually factored out, otherwise the ranking of stakeholders which joined the

platform a long time ago will end up becoming a constant; their ranking will be more based on historic activities and less on current ones.

Including the above characteristics, we envision the 'Ranking' formula as following:

$$Rating = \sum_{i=1}^{N_{buyers}} \left(\log(BuyerTotalVolume_i + 1) \sum_{t=1}^{N_{weeks}} \sqrt{WeekSellVolume_t} \right)$$

$$BuyerTotalVolume_i = \sum_j \gamma^{age_j} BuyerVolume_j$$

$$WeekSellVolume_t = \sum_j \gamma^{age_j} SellerVolume_j$$

- $BuyerTotalVolume_i$ is the historical part and calculates for every customer the aggregate volume of all deals with a specific stakeholders throughout history . The weight (or factored-in volume) of every deal decreases pro-rata over time ("age" means the amount of days ago a transaction took place) multiplied by γ^{age}, where $0 < \gamma < 1$ is a declining factor governing the speed at which a deals weight reduce over time. To take an example; assuming that $\gamma = 0.998$, one year later a deal's weight will almost halve relative to the initial value $0.998^{365} \approx 0.48$, three years later the same deal weight is down by 90%, so on so forth. The number 1 is added to this part of the formula to obtain an aggregate volume to ensure the log outcome is always positive. Applying the log function should offset the impact of major buyers;
- Calculated in respect of stakeholder's direct deals with specific customer over the past week is the aggregated volume of deals for every week throughout their common history and is noted here as $WeekSellVolume_t$ with N_{weeks} being the common history length in weeks. The volume of deals, similarly to the deals with the customers in the first part of the formula, are multiplied by a correction factor γ^{age} . Of this the square root is obtained to ensure repeat deals have a higher impact on the outcome than one-time deals. To illustrate this part further an example: the volume of data sold by a vendor is worth 16 tokens. If this vendor sells all the data in one time during one week, its weight will be: $\sqrt{16} = 4$. In case the vendor sells 8 tokens worth of data during one week and 8 tokens during the next week to the same customer, the aggregated weight is $\sqrt{8} + \sqrt{8} \approx 5.657$, which is 1.4 time higher than selling all the data at once. If, however, the same vendor sells to the same customer one token worth of data per week during a 16

week period, the total weight is $\sqrt{1} \times 16 = 16$, which is 4 times higher than selling all the data at the same time.

- The aggregated weekly volume of deals with the customer is multiplied by the customer's ranking in order to calculate its rankings in respect of the stakeholders's deals with every buyer (with N_{buyers} being the total number of customers). These rankings are added up and form the stakeholders's final ranking.

Ranking serves only as an informational tool, it has no direct effect on stakeholders and its main purpose is to ease the burden on Data Buyers allowing them to navigate the market place in a more efficient way.

Ranking can be calculated dynamically, although daily recalculation would be a the preferred option since reading the whole blockchain-based deal history over a longer period of time makes it a relatively costly operation.

Weekly calculation groupings make it more difficult to boost rankings by regularly closing smaller deals with a fictitious customers as effective boosting requires a one-week pause after every deal.

6. High-level architecture

6.1. Storing data

Being able to store and facilitate a secure transmission of data, are a crucial goal for Prometeus. Given large amounts of data to be stored in excess (on multiple nodes) and be readily available at any time, the Data Owner, the Validator, the Enricher, or the DataMart cannot be entrusted with storing data.

A decentralized cloud data storage network enables data integrity, weakens the effects of possible equipment failure on data availability and reduces the security risks. Additionally, network size and openness leave further room for cost reduction in the systems. Most important is the fact that data stored in this kind of network will be resistant to censorship, forgery, theft, or availability failures. All of the above is described in detail in projects like FileCoin, Sia, or Storj.

We at Prometeus are considering the following input data storage options: ipfs30, BigchainDB, OrbitDB, Fluence, Ethereum Swarm, Picolo or Postchain (ChromaWay). The final choice will be made at the project implementation stage. The Validator will download data to the node that will save data to storage and record only

hash on the blockchain. The format of downloaded data will be made up of two parts: Metadata and Encrypted Data. “Metadata” means open public information as to what is downloaded by the Validator. Metadata is required for data search and description for DataMart. “Encrypted Data” refers to any data to be purchased through the DataMart. It is noteworthy that data format is not limited by the system in any manner whatsoever since data format varies by social networking website – all the way down to screenshots. Metadata was specifically invented to give DataMarts an idea of what they are selling.

As any personal data can become outdated and lose its value, it will be deleted after a while depending on demand and data age. The intention is to reduce the nodes load.

Apart from data, the storage will contain the data stored on the blockchain itself:

- public keys and re-encryption keys;
- hashes of all newly saved and re-encrypted data;
- a list of all Validators and Enrichers specifying their ranking;
- a list of each Validator's announcements with the hash of all metadata describing data in every announcement.

6.2. Encryption

Sensitive data will enter Prometeus Network in an encrypted form (the initial encryption will be carried out by the Data Owner or the Validator). Encryption will be performed using the Data Owner's keys and all data handlings will be controlled by smart contracts.

Secure data transportation to the buyer will be implemented through the use of proxy re-encryption and the NuCypher project developments. The peculiarities of this approach guarantee that the Data Owner's data can only be received by the Buyer's private key owner.

6.3. Dead Man Switch

The Data Owner is unable to stay online all the time and generate re-encryption keys in real time. To that end, we employ Dead Man Switch – a mechanism that can be delegated the role of re-encryption key generation by the Data Owner (at their sole discretion). This mechanism gets triggered provided that the Data Owner has been inactive for a certain amount of time.

In Prometeus, the Dead Man Switch is made out of two parts:

1) The first part is a smart contract trigger. All a smart contract does is record a request for accelerated transaction execution on the blockchain. The need for this smart contract stems from the fact that there are no timer or

delayed contract launch concepts in Ethereum – one can certainly use services like Chronos, Oraclize, EAC that add the delayed start function to smart contracts, even though for a fee. This smart contract will therefore be launched by the Buyer whenever deemed appropriate. In fact, some Buyers are not particularly troubled with receiving data with a short delay.

2) The second part is a secure storage that will generate a re-encryption key for the Buyer. Of course, key generation will never occur until all checks are complete: an order exists on the blockchain, the order has been prepaid, the Buyer has submitted a key generation request. We at Prometeus Network are considering the following options for implementing this mechanism: enigma, keep.network, teex.io (tee-ds), Covalent.ai, Arpachain.io. The second part can also be delivered through the use of Shamir's Secret Sharing, allowing elimination of intermediaries but imposing certain requirements on nodes and network size.

The mechanism will work a following:

- the Buyer who places an order decides to have the transaction accelerated. Possible reasons include but are not limited to: the Data Owner is currently offline – a delay of several hours; the Data Owner is completely gone – a delay of several days;
- the Buyer calls a smart contract and submits a request for accelerated transaction execution;
- secure storage verifies/validates the request and the order generates a re-encryption key and saves it in the order;
- the Buyer downloads the key and decrypts the data.

6.4. Data access control

A Data Owner must have control over its own data and therefore be able to stop data sales or completely remove data from the system.

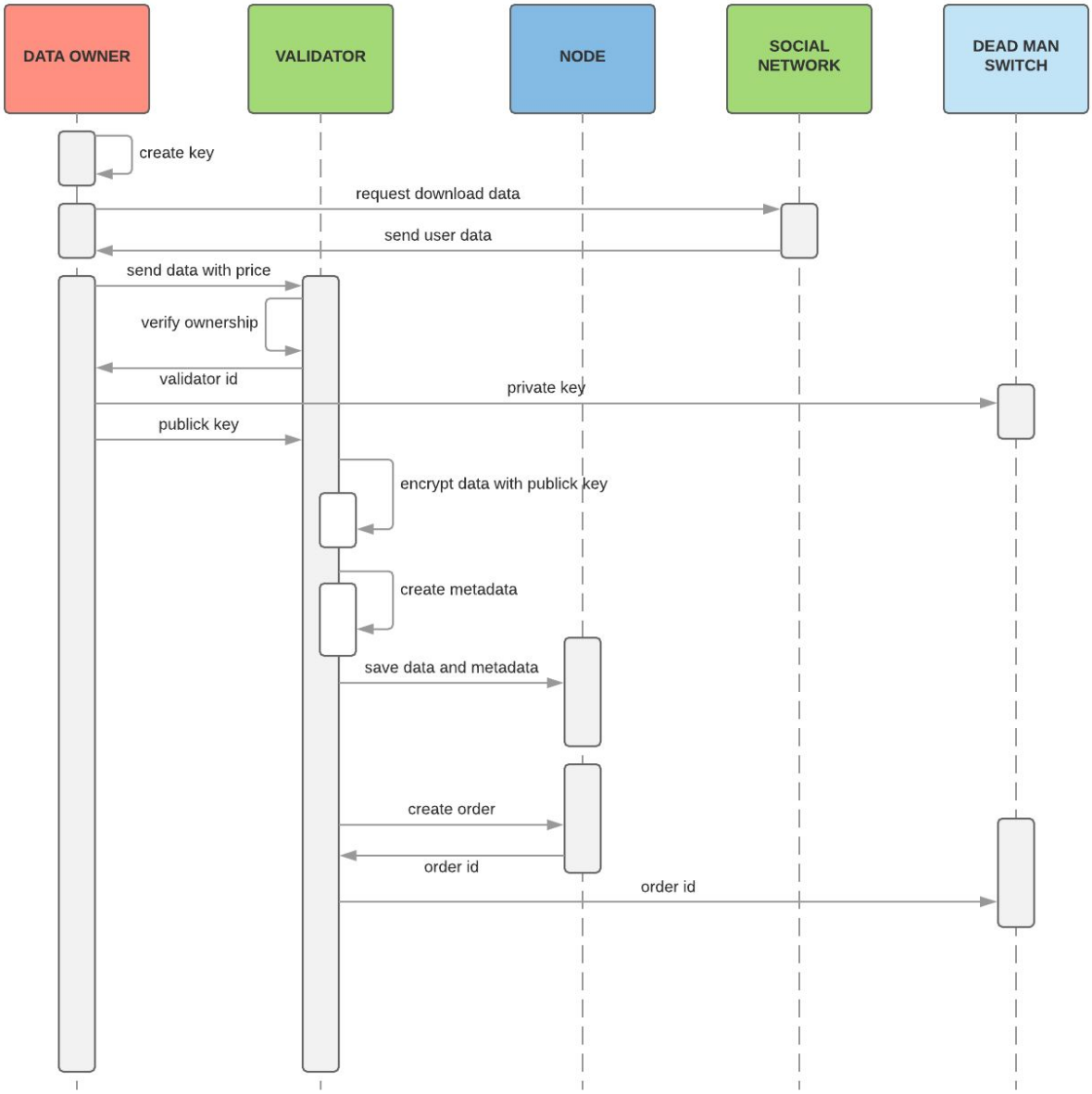
In Prometeus,data control is delivered through the Data Owner's private key. The fact that the Owner's data is encrypted using its own unique private key, enables changing announcements through the use of smart contracts and giving the node a signal to remove data from storage. For further information, see the detailed description of the algorithm below.

6.5. High-level operating algorithms for Prometeus

Roles:

- Data Owner: downloading data from social networks , creating the Prometeus encryption key (also known as address);
- Validator: downloading data to the node, initial encryption using the Data Owner's public key, data verification, and the engagement of / off-chain support for Data Owners;
- Enricher: data enrichment;
- DataMart: data sale;
- Dead Man Switch: off-chain storage of the Data Owners' keys, e.g., Enigma;
- Node: storing data, supporting network operation;
- Buyer

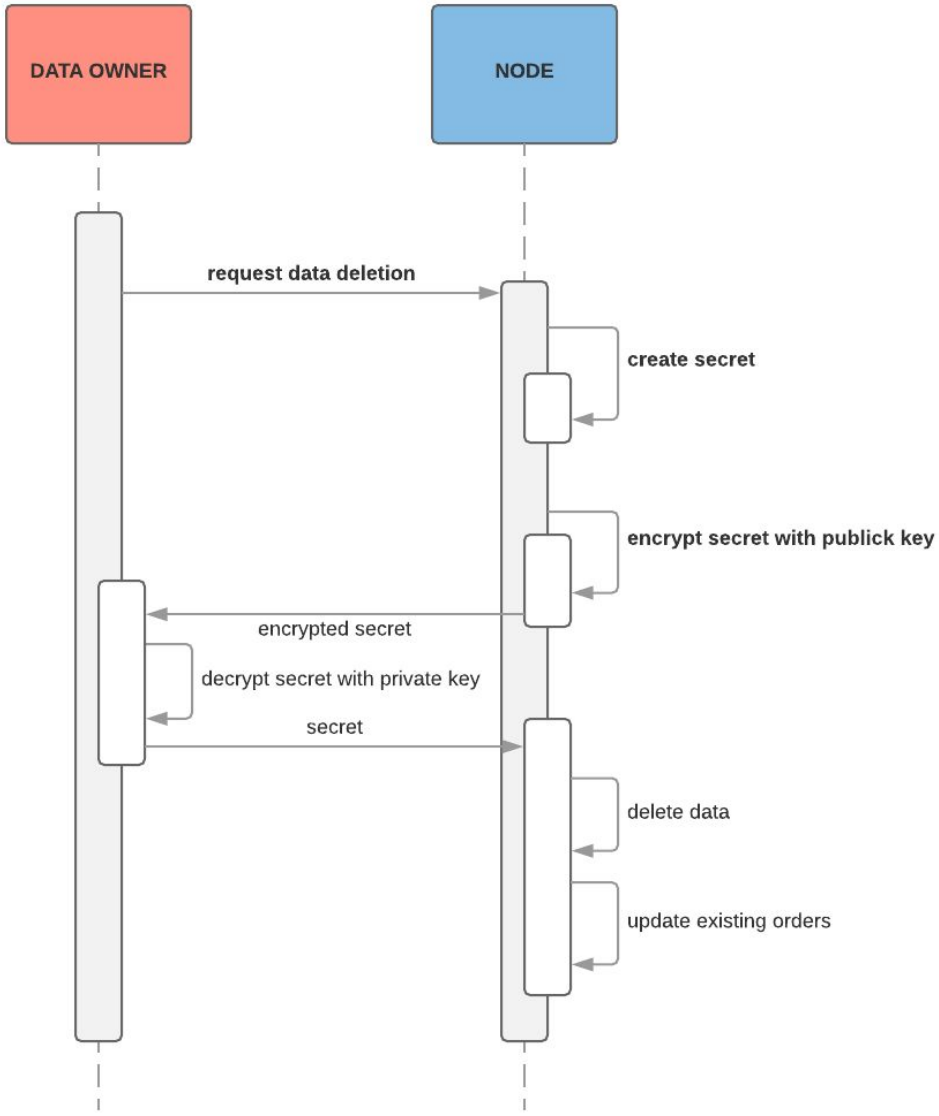
Algorithm to create a sale announcement (simplified option without the Enricher):



1) the Data Owner creates a private key and downloads data from social networking websites;
2) the Data Owner forwards its data to the Validator alongside the public key and a price;
3) the Validator verifies and validates the Data Owner (KYC, agreement, meeting, etc.), analyzes data; once complete, the Validator:
    a) creates metadata describing the available data for buyers. Metadata will be publicly available, free of charge, and stored in an open, easy to access place;
    b) encrypts input data using the Data Owner's public key.

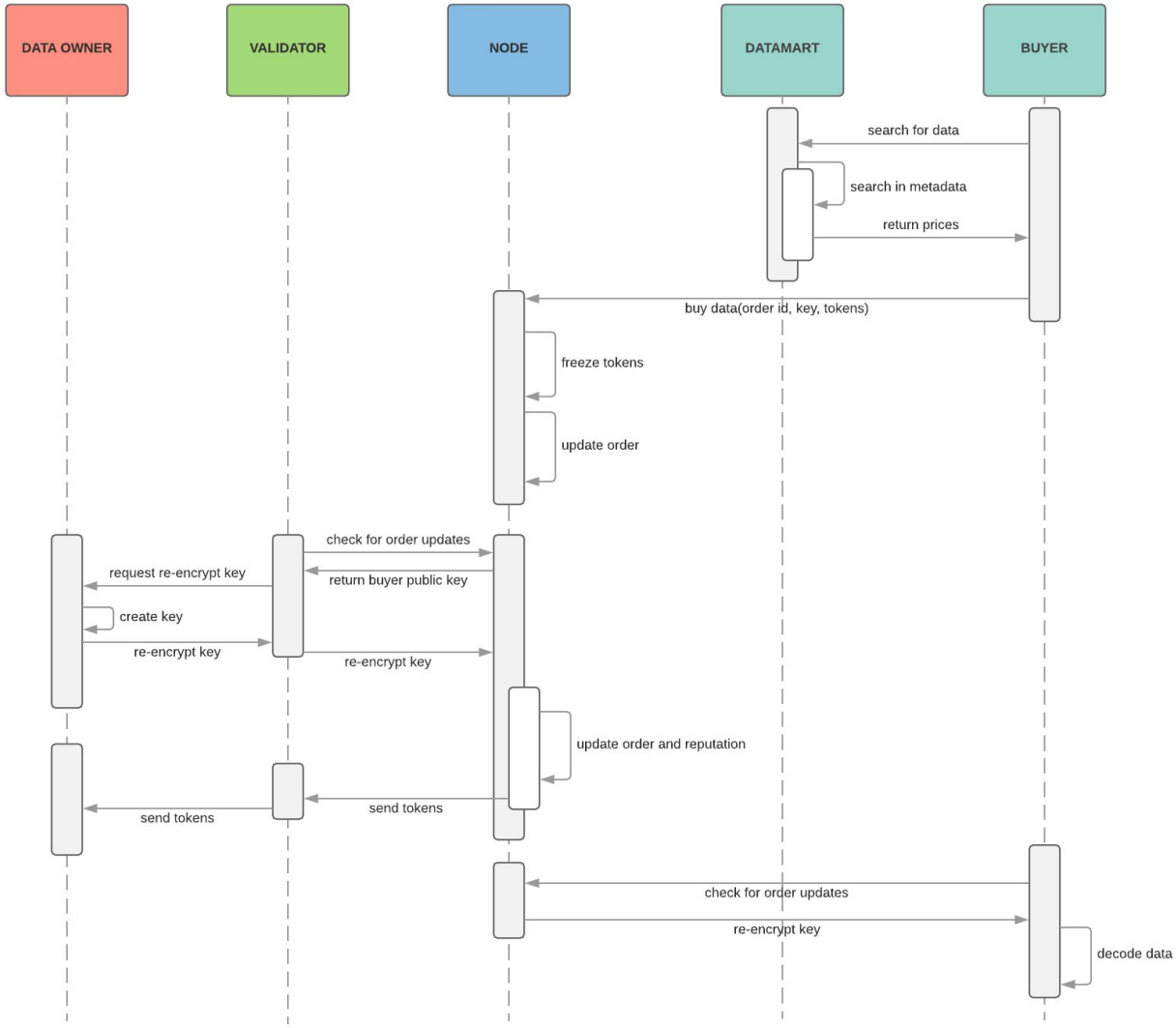
4) the Data Owner saves its private key into the Dead Man Switch (at their sole discretion) and allows the Validator to add the announcement ID to the same Dead Man Switch;
5) the Dead Man Switch saves the key in a non-readable format. The only thing that needs to be done after verifying the announcement is to create a key for data re-encryption and charging the desired service fee;
6) the Validator saves data to storage and receives two hashes (in respect of metadata and encrypted data);
7) the Validator creates a sales announcement through the smart contract specifying two hashes and the price;
8) the Validator saves the announcement ID to the Dead Man Switch, which is required to execute a transaction when the Data Owner is offline.

Algorithm for controlling the Data Owner's data:



1) the Data Owner submits a (partial) data removal request(full account or only the part provided by the Enricher) ;
2) the Node creates a secret (random string) and encrypts it using the Data Owner's public key, Afterwards it saves the request on the Node;
3) the Data Owner decrypts the secret using its unique private key and saves the requested result and verifies if the request is ‘true’;
4) the Node changes all orders and announcements related to this Data Owner (by removing all unwanted hashes of files from orders and announcements);
5) the Node deletes all unwanted files (the hashes of these files retrieved from the blockchain history stop giving data);

Data purchase algorithm (simplified option without the Enricher):



1) the Buyer -by using the Market- finds data and a source from which it wishes to buy;
2) the Buyer creates a data purchase order, by:
    a) specifying its public key for re-encryption;
    b) entering Tokens, which remain locked until the deal is complete;
3) if the Data Owner is online:
    a) the Validator is continuously checking their announcements with the aim of discovering a purchase order;
    b) the Validator detects a new order;
    c) the Validator contacts the Data Owner and requests that the Data Owner creates a re-encryption key for the Buyer (the private key is available only to the Data Owner);
    d) the Validator updates the order and specifies the re-encryption key;

e) the Node forwards tokens to the Data Owner and the Validator;
f) the Buyer is continuously checking the order with the aim of discovering a re-encryption key;
g) the Buyer downloads the file and re-encryption key and decrypts the file (NuCypher).

4) If the Data Owner is offline and saved their key to the Dead Man Switch:
   a) the Buyer decides to accelerate the deal after a while;
   b) the Buyer submits a request for the Dead Man Switch;
   c) the Dead Man Switch checks the order (the announcement number was saved by the Validator when creating the announcement and the Dead Man Switch checks if the order belongs to the announcement) and creates the key using a private contract (see the Enigma White Paper);
   d) the Dead Man Switch updates the order specifying the re-encryption key;
   e) the Node forwards tokens to the Data Owner, the Validator, and the Dead Man Switch for their delivered services;
   f) the Data Buyer is continuously checking the order with the aim of discovering a re-encryption key;
   g) the Data Buyer downloads the file and re-encryption key and decrypts the file (NuCypher)

Operating algorithm for Enricher 1 (free data):

1) the Enricher publishes information regarding the enrichment types offered, the data required for each type, and the enrichment price payable by the Buyer;
2) the Validator downloads information about the existing Enrichers and their reputation from the blockchain;
3) the Validator selects the Enricher based on criteria such as; reputation, the enrichment types offered by the Enricher, and the data format the Validator requires. Once completed, the Validator creates a request for data enrichment from the desired Enricher;
4) the Enricher finds a request for data enrichment on the blockchain and updates the request in case the Enricher accepts it;
5) the Validator creates the data which needs to be enriched, encrypts it using the Enricher's public key, and saves it to storage;
6) the Validator updates the request for the Enricher by specifying the hash of the file which needs to be enriched;
7) the Validator updates the sale announcement by specifying the Enricher’s ID, thus allowing the latter to add its data to the announcement;
8) the Enricher enriches the data and creates metadata describing the enrichment;
9) the Enricher encrypts the enriched data by:

a) using its key, thus shielding the data from the Data Owner – in this case, however, a higher selling fee will be charged due to the necessity of using one more re-encryption key (generating a re-encryption key and saving it to the order);
b) using the Data Owner's public key. Note that in this case, a lower data selling fee will be paid out to the Data Owner but the Data Owner will receive the enriched data free of charge;

10) the Enricher saves all data to storage;
11) the Enricher updates the Validator's announcement and adds hashes with enrichments and metadata;
12) … When executing a deal, the Buyer can select enrichments for the data being purchased, in which case the smart contract will automatically allocate the profit in tokens among the Data Owner, the Validator, the Enricher, and the Dead Man Switch (if the Data Owner is offline).

Operating algorithm for Enricher 2 (paid data):

1) the Enricher downloads information about the existing Enrichers and their reputation from the blockchain;
2) the Enricher selects a Validator and the announcements that the Enricher wishes to enrich;
3) the Enricher purchases data for the selected announcements;
4) the Enricher enriches and encrypts data and then saves it to storage;
5) the Enricher submits a request for adding enrichments to the announcement. The smart contract checks whether or not the data contained in this announcement was purchased by the Enricher and allows the Enricher to add their enrichments to the announcement;
6) … the rest follows the standard pattern.

6.6. Prometeus stages

The implementation of Prometeus necessitates the availability of smart contracts, cheap or even free transactions, cheap data storage and the possibility of basic operations involving data (possibly through a side chain). The most important requirement however is a formal decentralization. This implies that the blocking of nodes or closing of validators and/or the elimination of *any* other network role- will not affect the functionality of Prometeus.

We considered a wide range of project implementation approaches for Prometeus before we finally settled on the following step-by-step project implementation solution.

Step 1: maximizing the use of the existing projects in the first release and therefore abandoning in this stage -for now- the Dead Man Switch mechanism and enrichers-related mechanisms . The Ethereum + Ethereum Swarm + NuCypher binding will enable a quick system launch with the minimum functionality possible. Business logic and Ethereum-based smart contracts, storing data in Ethereum Swarm, delegating data access through the use of NuCypher. This choice stems from the production-ready projects described above, formal decentralization, and ease of use. These have all been launched and the other applications designed on their basis are ready for launch.

This kind of architecture has its drawbacks, too, including expensive transactions within the Ethereum network.

Despite the existing drawbacks we will start engaging customers, since we believe high transaction prices will be offset by wholesale purchases, and agreements with major Validators will counterbalance the drawbacks related to not having a Dead Man Switch available from the start.

Step 2: Network scaling through the engagement of Validators. Transactions will be made cheaper by launching an in-house Ethereum-based network with Proof-of-Stake consensus. Adding Enricher support and data control mechanism.

Step 3: Implementing the Dead Man Switch mechanism based on working solutions available in the current market (Enigma, teex.io, arpachain.io). Engaging a large number of data owners and increasing network size will enable adding data storage function to in-house nodes, thus further reducing the cost of using Prometeus.

Step 4: Scaling to adjacent data markets: health care and genomics, data for AI algorithm learning (autonomous vehicles, face recognition) and human resources, statistics, photos, and other types of multimedia.

Let us discuss the highlights of Step One in greater detail:

1) Quarter one of 2019. Developing client version one that uses NuCypher and Ethereum Swarm for data handling and encryption purposes. This will allow both saving data to storage and exchanging data via a secure method.

2) Quarter two of 2019. Creating the main Ethereum smart contracts:

- announcement handling – all aspects of creating a data sale announcement, Validator interface;
- order handling – all aspects of creating a data purchase order and executing a deal, DataMart and Buyer interface.

This will make it possible to execute simple deals with non-enriched data, thereby meeting -to a certain extent- demand in respect of major wholesale buyers.

3) Quarter three of 2019. Creating open source client(s) for Data Owners, DataMarts, and validators;

System participants will thus be allowed (without investing too much in development) to work in Prometeus – Data Owners will be able to upload data easily; DataMarts, to sell data.

Enricher interface development, Dead Man Switch, and data control are all part of Step Two.

7. Tokenomics

The Prometeus Network **native token** performs two main functions: allowing customers to access the product (*utility token*) and rendering financial support for stable and secure network operation (*work token*). One token combines the two functions in an effort to strike a balance between the interests in intra-network token price rises and falls. Tokenomics is primarily aimed at:
a) contributing to the product price competitiveness by reducing the share of overhead,
b) ensuring the economic stability of a decentralized network of storage nodes and consensus nodes.

7.1. Token functions and circulation

The functions of *utility token* are:

1. the token is **the only** payment method in settlements between DataMarts and Validators/Enrichers, this means the internal price of data is denominated in tokens;
2. the token can be a possible payment method for purchases by the Data Buyer from DataMart. If the Data Buyer chooses to pay in fiat currency or cryptocurrency, DataMart **purchases** the necessary amount of tokens in the secondary market for internal settlement purposes;
3. the token can be a possible payment method in settlements with Data Owners. If the Data Owner selects different payment options, the Validator or the Enricher **sells** part of the received amount of tokens in the market.

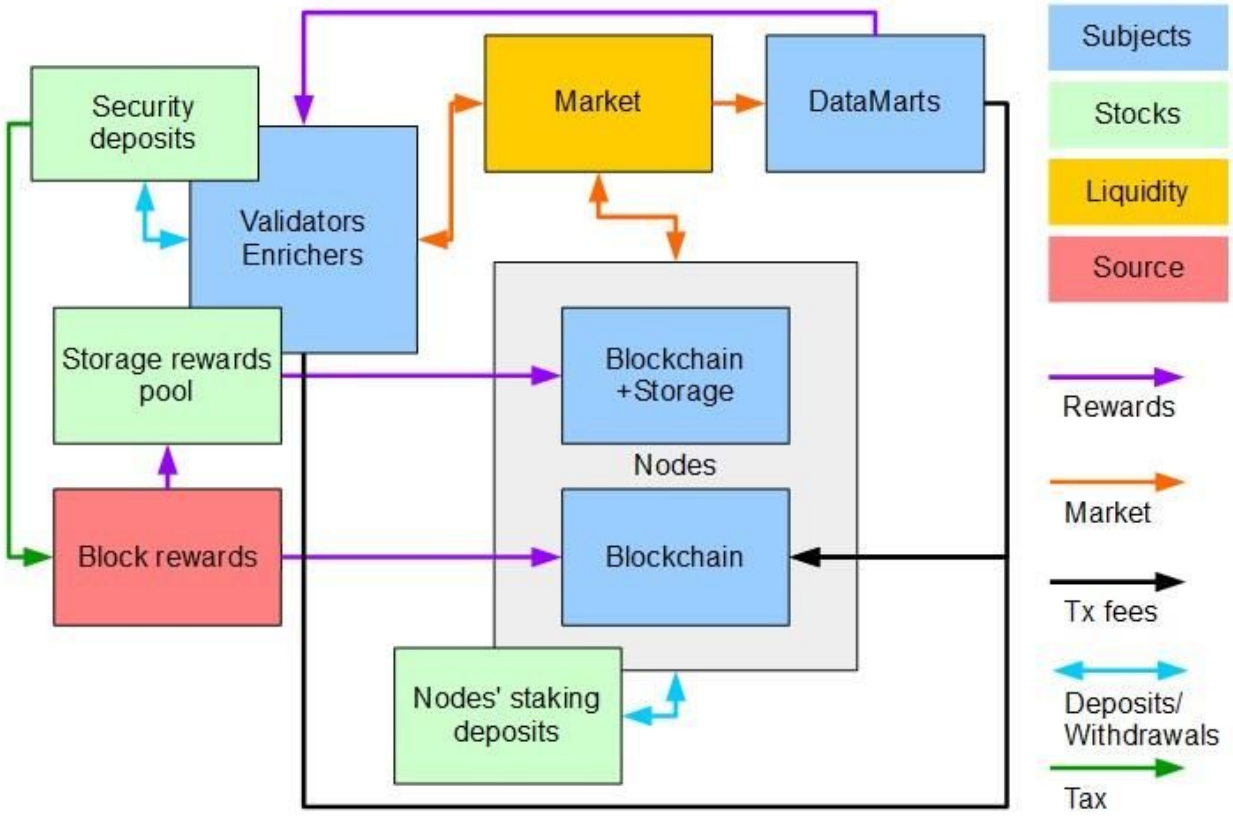
The functions of *work token* are:

1. serving as the currency of Validators' and Enrichers' insurance deposits, which is a prerequisite for adding data to storage and selling it to Buyers; the insurance deposit tax finances the blockchain node fees upon completion of token minting;

2. serving as the blockchain node deposit currency in the Proof-of-Stake consensus algorithm, as well as the consensus service fee;
3. serving to pay the transaction fees on the blockchain;
4. constituting data storage and processing fees;

Paragraphs 2–4 above imply a **stable supply** of *work tokens* in the secondary market, which is to be offset by the *utility token* demand (from Buyers and DataMart), as well as demand from new users and nodes for the formation of insurance deposits and PoS deposits. The transaction fees are payable by system users (Validators, Enrichers, DataMarts) and receivable by blockchain nodes. The data storage and processing fee is received by the owners of storage/re-encryption nodes, which simultaneously act as blockchain node owners; this fee is paid out of a special token pool that is topped up at the protocol level out of the share of PoS minting.

The figure below shows a schematic layout of tokenomics:



Normal system operation depends on token circulation and price in the secondary market, where most network roles can act on both the demand and the supply side, but the exchange rate will primarily be affected by demand from DataMarts and supply from nodes.

7.2. Network tokenomics

Token issuance/minting is performed by the blockchain nodes that serve Proof-of-Stake consensus. The likelihood of receiving a block depends, among other things, on the amount of blockchain node deposit; a more precise PoS algorithm will be laid down later, yet a relatively low deposit must retain the ability to receive a block, thus allowing decentralization to be maintained and the network to be expanded in the context of high token prices. The **block reward for blockchain nodes** is set at a fixed amount; the higher the likelihood of receiving a block with growing deposit, the higher the nodes' deposit yield . Part of the block reward is transmitted at the protocol level to the **contract pool** out of which the data storage and processing fee is paid.

Token circulation in the network consists of **two long-term phases**:

1. Tokens continue to be issued until the maximum token supply is reached. Tokens will be distributed between preminting and minting in such a way that during this phase BFT consensus requirements are met and one third of tokens is prevented from being concentrated in the hands of potential fraudsters;
2. After token supply reaches its maximum value, the block rewards are paid to PoS nodes in the same amount and pace out of the **insurance deposits tax**. The tax rate is recalculated from time to time as the ratio of the rewards payable to blockchain nodes in the period to the total amount of insurance deposits in the system (see paragraph 7.3).

The rewardsfor storage/re-encryption nodes become due and payable provided that the challenges from Validators are successfully met, with the premium being won by the node which is the first to provide on-chain proof of data availability (e.g., adds the relevant transaction to a block). The algorithm of obtaining proof is coded in a smart contract, with the cost of operations ("gas") therefore included into the Validators' expenses. The premium is essentially paid from the smart contract pool; accordingly, the network takes on the task of incentivizing independent nodes. Validators can also be reimbursed for their gas costs from the pool.

This Validator-controlled lottery model is easy to implement and has the following features:

1. the chances of being awarded a premium can be increased for nodes with local data access;
2. the Validator can encourage the storage of specific (more valuable) data;
3. the nodes with a larger PoS deposit are encouraged to store bigger data amounts; overall, the chance of winning depends on a specific Proof-of-Stake algorithm;

4. the premium must be fairly high as the award to the node depends on block finalization.

Where an **external data storage service** (e.g., Fluence) is used, the service costs are incurred by Validators.

The cost of transactions will, to a greater extent, depend on efficiencyof smart contracts. Minting, with node operation costs factored in, must cover this cost for users (Validators, Enrichers, DataMarts); in this case, the network will be completely **self-supporting**. If the profitability of a blockchain node drops below this point, the internal token-denominated data price or the Buyers' price will include the transaction fees. The amount of block reward and the consensus algorithm clarification will be determined according to profitability requirements.

7.3. Insurance deposits

The deposits of Validators and Enrichers are designated to signal buyers that participants have a stake in the system, to protect the system from bad actors, and, broadly, strike an economic balance.

Specific requirements for the deposit amount depend on the participant's historical ranking in the system. The calculation of a historical ranking based on sale history is described herein (see 5.3.1 above). It is an additive ranking, i.e., the overall sale history in the network can be reflected in the overall ranking, which is the sum of each participant's rankings. The amount of required deposit in tokens is determined using the formula $A_i = \varphi^{R_i/R+c} D_0$,
where

- *Ai* is the adjusted reposit requirement for the participant *i*
- *Ri* is the participant's historical ranking
- *R* is the overall historical ranking
- *c* is a technical constant
- *Do* is the constant determining the deposit requirement assuming zero ranking
- *phi* is a curve factor that is also equal to the factor for *Do* assuming that the maximum ranking is Ri=R, 0>phi>1

The required deposit in tokens must be sent to a special smart contract from the participant's address. **Deposit sufficiency is checked** in two cases : when data is added to storage and during weekly/quarterly historical ranking recalculations. Should a deposit be found to be **insufficient**, a participant loses the right to add, update, or sell data. The amount of deposit in excess of the calculated requirement can be withdrawn to the participant's initial address. The remaining part of a deposit may be withdrawn only in the event of **complete withdrawal** from the system, in which case the participant's historical ranking is reset to zero.

At a later system development stage (after token supply reaches its maximum value), the daily/weekly **tax in favor of nodes** is deducted from all insurance deposits at the rate calculated as the ratio of all payments made to blockchain nodes in the period to the total amount of deposits in the system (see paragraph 2 above). Deposit sufficiencyis not checked after the tax is deducted.

An optimal amount of deposits in the system *D*, depending on ranking spread, will oscillate between the maximum $D = nD_0$ (assuming zero network ranking) and minimum values $D = \varphi^{1/n} nD_0$ (assuming a uniformly distributed ranking). Participants are not incentivized to deposit any sums in excess of the mandatory amount, taking an additional reserve for possible drop in ranking and tax deduction into account.

There is a technical deposit for tax service purposes, which is topped up out of the block rewards as needed (i.e., if there are no active users). Since the minimum required deposit in the system is $D = \varphi D_0$ in the case of one participating user and the tax rate is calculated as b=E/D, where *E* is the rewardpayable to blockchain nodes for the period in question, the amount of technical deduction is bE = $\frac{E^2}{\varphi D_0}$.

7.4. Token stabilization

Lower token prices put the network at risk of collapsing due to insufficient node owner motivation and a low cost of insurance deposits. A decrease in the number of blockchain nodes will bump up their profitability, thereby striking a new balance in the system. The token price inflation might, in turn, put upward pressure on the *utility token* demand. All of the above are market stabilization instruments. Data availability must be boosted by Validators through the use of challenges (see 7.2 above), which does not lead to an excessive circulation growth assuming a low number of nodes. The protection of investors' interests at an earlier stage must be ensuredby the initial structure of token supply (i.e., metrics).

Upward pricing trends pose a threat to price competitiveness due an increase in transaction cost. When this is the case, a natural increase in the number of blockchain nodes can be supplemented by the growth of storage nodes due to the more frequent issuance of lottery challenges by Validators, with circulation and supply set to rise more notably. The internal price deflation will dial down the demand of *utility token* from DataMarts. This represents an additional channel of Validators' influence and could be leveraged at an earlier stage to expand the network and get business expenditures under control.

Fin