

cyber•Chain: Motivated Search and Evaluation Engine for Permanent Web

Dima Starodubcev

cyber•Fund

Working Draft

Abstract

Existing search engines are restrictive centralized databases everybody forced to trust. Emergence of content-addressable storage and distributed ledger technology create opportunity to shift existing web's ubiquitously used client-server architecture based on DNS, HTTP and IP protocols to a truly peer-to-peer interactions based on stateless IPFS and variety of stateful consensus computers such as Ethereum. Moreover blockchain architecture itself allow to organize search engine for public information in a way inaccessible for previous architectures. This creates a challenge for a search engine based on emerging technologies and specifically designed for them. In this paper we discuss opportunities behind this transformation, challenges of crawling, indexing and evaluation for the next generation web and propose a blockchain based experimental set of smart contracts to address discussed issues.

Introduction

Lets us start a discussion from disadvantages of conventional general purpose search engines:

- No Transparency. Nobody outside of Google understand how the ranking really works. That creates a market for black and white SEO. The truth is that if e.g. Google disclose complete details of ranking algorithm it would be easy for adversaries to game organic search results that kill quality of results and ad revenue streams. Pagerank [PR] has no inherent trust mechanism resistant to sybil attacks. This problem can be addressed adding transparent and accountable blockchain based ledger with properly designed economic incentives built into the system.
- No Access. Currently all search engines are centralized. Nobody is able to add to index as well as participate in improving quality of search results. Google itself internally use a workforce of *search evaluators*. It is our belief that user generated search engine could have higher quality of results as in a story with almost every web site in existence.
- Broken Incentives. Vast majority of contribution to a search quality is made by users. Than any user search something she extend semantic core. Than any user click on search results she train a model. This creates an opportunity to continuously improve ranking model at the expense of users. Than a search engines sell users to advertisers at the expense of harming user experience and acquire revenue streams which are not returned back to users at all. This simple loop created Alphabet's \$550 billion capitalization (~\$80 per Earth capita) in 18 years. We want to change that.
- Central Control. Google become too powerful. It is scary to imagine a future there *everything* about *everybody* is known and controlled by **closed** AI corporation. Imagine the world there (1) the only country exist, (2) nobody can control it's government and (3) everybody should obey decision of government without any explanation. There should be open, transparent and accessible *alternative* with *decentralized control* built on principles of modern distributed interplanetary content-addressable cyberspace [IPFS] and DAO like governance [RALF].
- Annoying Ads. Separation for organic and ad search results is unnecessary. In fact all ranking decisions are being

made by search authority. But for paid search Google use free market solution to determine a fair ad price for every word in its gigantic semantic core. Historically free market solutions are more efficient in virtually any area of decision making. Why do not use the same principle for the ranking itself desintermediating annoying ads? Let us imagine that every link can be (1) curated or audited by everybody, (2) based on this trusted metric cyber•rank (page rank based on economically incentivized curation and auditing) is calculated and then (3) everybody can promote this link further by burning some money automatically bringing value for everybody in existence. For every action everybody earns a share proportionally to contributions. This non-zero-sum game is significantly more sybil resistant and that is where we are heading.

- One Way Trust. Everybody used to trust Google, Baidu and Yandex. But Google, Baidu and Yandex don't trust users. E.g. you cannot report some kind of proof that given link is a lie and should not be indexed so high. It can count your attention during ranking but can reject to count it. You cannot know what happens inside because Google, Baidu and Yandex don't trust us. We want to establish a system where trust is bidirectional between search engine and users because search engine ownership is distributed across all its users based on which all ranking decisions are made.
- Zero Privacy. All search engines will answer you only if they explicitly know how to map your device with your real identity or pseudo identity which is tracked by RTB [RTB]. Otherwise you should prove that you are not a robot every time you search. That harms our privacy. Moreover, robot abuse is another hot topic that is about to happen. Nonetheless, nothing should harm our privacy.
- Censorship. Though it's well known that Google working hard to prevent censorship we all know about China case and [Transparency Report](#). Good search should be resistant to censorship without exceptions and build for interplanetary scale in mind.
- Online only. Worth to note that you cannot search offline even if necessary information is stored next door. If we are cut from the wire or backbone we are powerless. Global offline search is not a feature which can be easily deployed even by multibillion corporation. This goal is nearly impossible to achieve based on centralized architecture. Only accessible distributed systems can solve this fundamental problem for the next generation Internet. This future is not about gateway keepers in form of ISPs but about mesh networking and peer-to-peer communications.
- Weak Security. What happens if tomorrow my Google account will be blocked? Do we have something to prevent this? Do I have necessary level of assurance that *guaranty* us our security based on math. All solutions are here but to solve this important issue we need to do a lot of work as security is a foundation for life, liberty and property.

Pretty huge amount of problems to fix. It would be naive to bootstrap a search engine from a scope where Google, Baidu and Yandex exist. We need to find a special area where general purpose search engines suck. Variety of distributed ledgers such as blockchains and tangles can be primary content-addressable data suppliers and this is a scope where current search engines are not the best at work. Moreover blockchain technology evolves very rapidly and has a lot of promises so it is a sure bet.

The idea is to initially deploy a blockchain based search engine for the purpose of searching across other blockchains so it can be useful from the first day. In parallel we design the cyber•Fund *application* [CFUND] based on cyber•Chain to solve a problem of trustless realtime blockchain asset valuation. But we need to design cyber•Chain in a way to be scalable for a more broad definition of a general purpose *search and valuation engine*, so more applications can emerge. Currently about 10 trusted (and thousands of non trusted) and globally available distributed ledgers exist with about 1 billion transactions accumulated. Just in last year amount of accumulated blockchain transactions increase tenfold. Not all these transactions are financial in some sense. E.g. Steem [STM] blockchain and its Russian sister Golos [GLS] are primary store user generated text such as posts and votes. As transactions are the only way of changing states in databases currently used by any web application we foresee that distributed ledgers become primary source of public information in 21st century due to tremendous benefits of the technology [ENIGMA, ...].

Thus we are to declare the *principles* of a general purpose decentralized and distributed search engine for the upcoming age:

- Privacy and Security. Just it.
- Ubiquitous Ownership and Access. Everybody should have a right to possess a piece of it.
- Mesh networks future proof. It should work in every connected surroundings.
- Interplanetary scale. It should work on Earth and Mars.
- Tolerant. In the era machine learning it should work for any kind of thinking beasts.
- Blockchain Agnostic. Foundations behind its design should not rely on any protocol or stack rather be explicitly derived from the nature of the information itself.
- Beautiful. Business model should not harm every minute experience.
- Transparency and Trustfulness. Every piece of its reasoning and behavior can be audible by everybody.
- No Single Point of Failure. Nobody should have a single key to modify or change it.
- Sybil Attacks Resistance. This resistance should be derived from the properties of a free market but not from some single authority.
- Intelligent. It should answer *natural* questions with *easy to read* and *provable* answers no matter text, media or natural numbers should be involved in the answer.

Among this principles there are *requirements* such as performance, usability and scalability. Keep discover. We discuss everything step by step.

Design Rationale

The idea of a permanent web behind IPFS is beautiful in its simplicity. Every *piece of data* has unique address:

Using this address this piece of data can be found in a global data structure called MerkleDAG [DAG] (logical representation of data storage) across peer-to-peer IPFS network using bulletproof DHT. Nodes are weakly incentivized for fair data exchange using BitSwap protocol. This link can point to to *any* piece of data such as GIT object, BitTorrent link, Bitcoin block, JSON document, picture, video, plain binary data or even a small piece of text:

Let me explain the power of this solution for a search engine:

Data is unique and self-authenticated. If you know this hash is a piece of data you trust, you should not care where it is came from. This property free search engine from rescanning work significantly reducing resources necessary for keeping index fresh. We call this approach *setup once index forever*. Workflow of conventional search index is straightforward: web crawler fetch a page, than follow up links, fetch the following pages and so on. After some loops web crawler is able to get virtually all links in existence. Information contained in webpages is mutable. Thus search engine should decide than and how to fetch indexed links for new versions of documents. As webpages are loosely structured than it should somehow filter all the noise. In our case we just need to check do we indexed this unique piece of data before or not. If we didn't we can make a decision to put it in index.

Direct Content Distribution. Weak incentivization of the BitSwap protocol has very interesting side effect: the more popular the file => the more people store pieces of it => the faster you can get it. This is in contrast with conventional client-server architecture where the more people want the file => the more resources a server needs (and more expensive distribution become) => the slower you can get it. This property significantly reduce resource usage to *deliver* results right to a user through distributed network.

Flexible Data Discovery. Big pieces of data is permanently linkable, thanks to MerkleDAG, so you can trivially reach a

small chunk of data. That enable a lot of powerful applications impossible for a peer-to-peer network before. E.g. SQL like data queries to a database distributed across the network, or REST like queries.

Thus documents which are located at `/ipfs` are *immutable*. But what if we want *mutability*? IPFS offer self-signed naming system which don't rely on conventional centralized and slow DNS (urls we use to use). Everybody can publish a link in a namespace `/ipns` with a mutable pointer to any `/ipfs` piece of data and sign it with its node's private key:

After an owner is able to mutate this link pointer to another `/ipfs` piece of data. This gives us another important properties for advanced search engine:

Fast Discovery. Publishing is visible in DHT and can be nearly instantly visible for all participants across distributed network without necessity to continuously send, receive and process HTTP requests. Average request is about 800 bytes long, so do every response [HTTP]. This overhead goes through the wire every time search engine want to know freshness of the web page. Interestingly, though HTTP/2 bring new awesome features it is still rely on the plain old location based addressing and ancient DNS, thus this cannot be faster by design. In our case we need sniff changes in DHT to know all news from a network. Currently this is about 100kb/minute though IPFS network is quite small. There is estimation [QGTC] that about 10% of Internet traffic consumption has been made by Google crawlers. It easy to imagine how much overhead can be eliminated across entire Planet.

Flexibility. Search results can be not ugly static snippets from 20 century but small dynamic programs *owned* by creators. It is hard to overestimate this idea. We will cover this topic further.

No Middleman. Search engine should not rely on a DNS middleman and can communicate directly with resource creators.

Ok. Now we understand that we have a usable way to store, reach and mutate the data across the globe. But what about ability to *comprehend* what data is behind this meaningless hashes? Interplanetary File System use a novel multihash, multicodec, multiformat, multibase [<https://github.com/ipdl/cid#cidv1>] formats for Interplanetary Linked Data (its not a joke either). IPLD is a project aims to define CID that being used across IPFS network. Content Identifier or CID is a self-describing content-addressed identifier. That crates enormous opportunity for optimization. Imagine a 32 byte link can contain everything to *independently* understand how to reach a piece of data and how to interpret it. Finally CID is the thing which make simplistic design of our proposed search engine possible in a consensus computer.

A nice property is that CID are based on well known cryptography so we don't need to rely on IPFS if something better came. Either the same addresses can be used to exchange piece of data in different peer-to-peer network. Though this don't solve a problem of extensibility. Every CID that has been write to an index should contain a first tag CID-1. Thus if something better came we can extend it.

But now IPFS is a perfect design choice with huge momentum across academia, open source engineering community and (the most important) blockchain engineering teams. It is our belief that it should lay as foundation for the next generation search engine. Everything is perfectly linkable (MerkleDAG), fast (DHT), accessible (BitSwap) and comprehensible (IPLD) and no single point of failure. IPFS is a protocol and it can be more useful with a good search engine specifically designed to it. Its possible to compute a pagerank for the whole MerkleDAG. But there are two problems with it:

- Amount of links in MerkleDAG grow $O(n^2)$. That is not either conventional web pages where 20-100 links per page. For one file 1 Mb in size can be thousands of links. Once the network starts to take of complexity inevitably increase.

- Even if we address some algorithm to extracts relevances from this links we should address even more algorithms to extract a meaning.

What we need is to find a way to incentivize extraction from this data fog a meaning that is *relevant to users queries*.

Search Workflow

Our proposed design of a search engine is based on advanced blockchain technology and enable everybody to participate and be rewarded. Everybody with account can search. To execute queries user should sign a **search** transaction with CIDv1 as payload and broadcast it.

```
cyberd search hipster true
```

A document should be a valid CIDv1. Post headers are purposefully unique. Rationale is the following:

- strongly encourage to bring valuable links first and get reward
- simplify search execution and ranking calculation
- make possible high-performant flat data structure

Than cyber•chain verify correctness of signature and either of two things happens:

- if CIDv1 is not in search index => Write to search index
- else broadcast a vote for existing CIDv1 and return a sorted links of relevant CIDv1

Based on this data client-side application can deliver documents by 3 ways:

- using full javascript implementation of IPFS [JSIPFS] (fast, but require initialization)
- using REST API provided by IPFS HTTP gateway (depends, can be fast or not)
- using local IPFS node (the fastest) This approach is simple and powerful. Developer have enough choices to balance between usability and performance.

CIDv1 can mean any piece of data. This are several use cases what can be returned for a user depending on the request:

1. Plain text for autocomplete, e.g. ['apple', 'asos', 'amazon']
2. A piece of media content which a user can play without necessity to go somewhere
3. Static formatted snippet with a link to a conventional web.
4. Static formatted snippet with a text answer.
5. A piece of javascript that can return dynamic snippet.

It depends on developers (who primary submit answers) and (users who primary rank answers) with what kind of things they want to answer questions. Possibilities are limited with imagination. Thus we propose a free market for answers on search queries everybody are encouraged to participate. How does it work?

Information about indexed CIDv1 as well as about its rankings is available for everybody. Thus those who are interested in rewards can monitor the blockchain for semantic core updates and submit links nearly instantly. Everybody can sign **answer** transaction with a link from to as payload and broadcast it:

```
answer hipster true
```

A document should be a valid CIDv1 and unique outbound link from answer. So for any given question the only unique answer is possible. Then cyber•chain verify correctness of signature and either of two things happens:

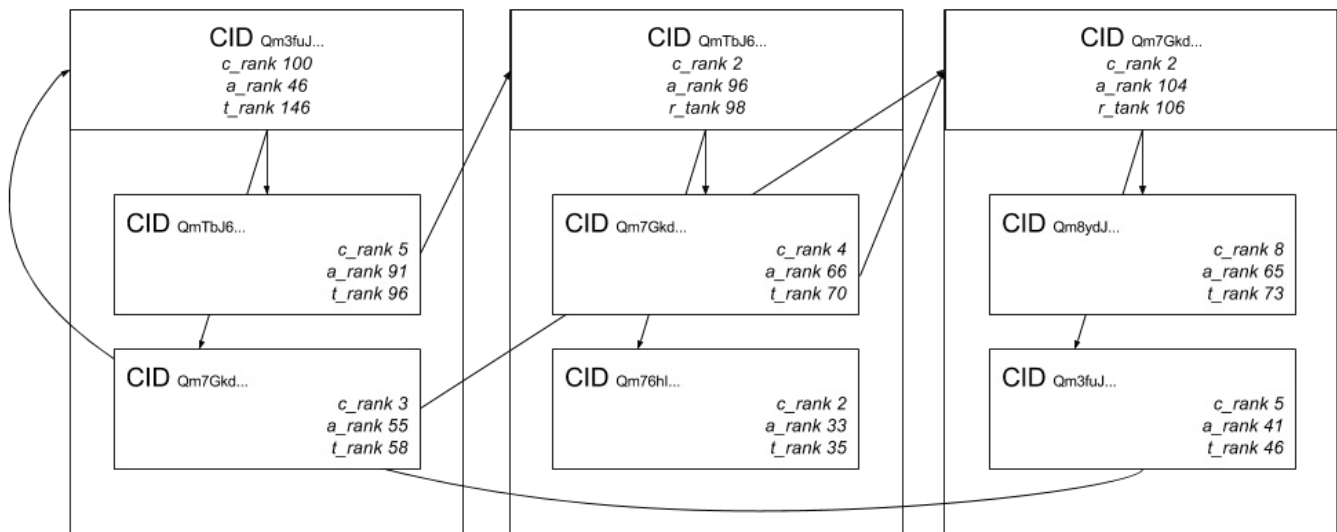
- if CIDv1 is not in search index => Write to search index
- if answer CIDv1 has no link to question CIDv1 => Write to answer index
- else broadcast a vote for existing answer CIDv1

We follow blackbox rule. In order to answer a question right you don't need a full comprehension neither the question nor the answer. You just need to match query with a relevant links.

That is a core API for the entire blockchain. Other methods accomplish support role for the thing. Such compact design open huge opportunity for performance optimizations. Also clean and comprehensible experience is very important for those who want to be involved. That is. The entire graph of semantic core with weights are open for everybody and available for data mining or any kind of weird AI stuff. But to make it work fast and relevant we need to find a way to calculate it.

cyber•rank

The idea is to combine two simple yet powerful algorithms: Google's pagerank and Steem's reward mechanism:



Where $t_rank = n_rank + s_rank$

n_rank , or natural rank is a rank based on Steem reward system. s_rank , or synthetic is a plain old pagerank used by Google.

Natural rank is acquired in a process of auditing and curation. Based on this rank payouts to those who involved (search and answer indexing, auditing and curation) are made. Each piece of data above consensus defined threshold get paid in 7 week. Details of implementation can be found in our Github. Worth to note that Steem reward mechanism is sybil resistant as votes are quadratic based on principle 1 token in system = 1 vote. In order to get a vote one should vest in shares for at least for 3 months. That solve a problem entirely because those who vote are strongly

incentivized in a growth of his wealth.

Synthetic rank is take as input natural rank. Conventional search engine work with nearly zero trust data. More than 200 factors used to calculate initial pagerank for a new document in the graph. Our novel approach allow to assign initial value based on sybil resistant voting. It is our belief that proposed approach can significantly simplify ranking and be more precise for information those nature is subjective.

Calculation of a synthetic rank for a huge graph is computationally intensive.

Recursion Limiting. Our model is recursive and require enormous amount of calculations which are limited within blockchain design. Model recalculation does not happens on a periodic basis rather it continuous. To limit precision of our model we can define Z as threshold for recomputing any given cyber•rank. It is hard to estimate what threshold should be set as correct. But it can be defined by market forces depending on model size and available computing resources within network. Thus Z should be defined as consensus variable by delegates.

Blockchains provide realtime high quality structured data which don't requires crawling. One node of any given blockchain is enough to provide verified data about transactions within on ledger without necessity to continuously revisit resources significantly reducing costs.

Both algorithms have strong proof in form of Google's \$550 bln capitalization in 18 years and Steem \$40 mln capitalization in 9 months. Combining both we can empower the world with a new kind of search quality.

Self Indexing Dilemma

Proposed approach has very unexpected limitation. What if we want to index cyber•chain using cyber•chain itself? Let us say that we have an awesome transaction that happens inside the chain and everybody are talking about it. It is popular thus we should display it in our search results. Adding it to an index spawn another transaction which (surprise) also should be indexed. This entanglement creates an infinite loop that bloat cyber•chain. This can not be a problem either. Consensus computer capacity and power is limited by the market forces. So we have two possible decisions:

1. Let it be. The market is a king. A bit of bloat can be a good piece of a knowledge about itself.
2. Strictly forbid indexing of the blockchain itself. Fortunately it is not so hard to implement on a consensus level. All we need is to check that CID has not been included in a cyber•chain before. That mean that cyber•chain transaction itself remain unindexed because in order to achieve this we (1) either should mutate data for hashed and timestamped transactions, (2) nor create possibility for a self-bloat. None of the options is not valid. Again, fortunately direct search without ranking among internal cyber•chain transactions can be available inside search results with the help of CID magic.

To be honest I don't know the right answer. To try or not try? Now we have solid foundation for indexing lets discuss what and how

Challenges of Indexing Distributed Ledger

Probabilistic Settlement. ...

Meaning Extraction. ...

Protocol Diversity. ...

Lazy Oracles

One specific ability is crucial for the next generation search engine. Application developers should have a motivation to provide structured arbitrary data feeds. Thus search engine can answer natural questions aggregating data from *highly structured and defined* feeds. This make possible a user get high quality *calculated* answers in realtime about state of reality expressed not only in links (which intelligent agent don't know how to parse) but in actionable numbers based on which its possible to make independent economic decisions.

But it is hard to find a tool to agree on publicly available and continuously evolving facts. We propose an approach to solve this.

Today different blockchain have functionality necessary to implement this. For eg. Ethereum enable construction of smart contracts that can validate and incentivize data feeds. But Ethereum has strong limitation: a price. Due to a network design every operation should be validated across the network of 5k nodes. For every put operation developer of such contract should pay in hope that somebody in the future will use this feed in the future returning costs. Current cost of a permanent storage inside Ethereum contracts is around \$200/megabyte. Worth to note that Ethereum has consensus variable gas limit. Currently a network load is around 10% of established limit. Once demand for computation reach a limit we will have a situation very similar for Bitcoin block size debate and price for storage can reach \$2000k/megabyte easily without validation costs. Pretty expensive for unlimited possibilities. There is alternative - Factom [FACT]. Its consensus design rely on a small amount of payed servers. Thus the cost is around \$1/megabyte. Such low price comes with high limitations. You can only put data to Factom and read it. There is no validation and incentivization built-in. There are permissioned blockchain designs such as BigChainDB [BCDB] and Hyperledger [HYPL] which solves validation problem perfectly but require strong efforts for developers to program and establish a network and then somehow monetize it. Lazy Oracles are going to fill this gap providing robust, cheap and reliable way *for monetizing* structured public data.

The process consist of 5 steps:

Step 1: Everybody can declare a soft protocol for data feed by posting a CID with tag *oracle* pointing to the following structure document:

```
// Basic Validation
doc_type: market_update // should be fixed
  exchange: string // domain name of exchange
  base: string // link to a definition of a reference namespace e.g. ISO or
chaingear
  quote: string // link to a definition of a reference namespace e.g. ISO or
chaingear
  price: uint16
  volume: unit32
// Audit Rules
Document format and data types should match a protocol
For every unique exchange and pair `market_update` report can be submitted no
more frequently than once per minute.
If either a price or volume for unique exchange and pair has not been changed
more than 0.1% a report should not be submitted.
// Tags
<CIDv1> <oracle> <market-update> <exchange> <base> <quote>
```

There are some soft rules for protocol declaration:

- Protocol declaration should be agnostic from language implementation and unambiguous
- Protocol declaration should have a git link to open source implementation of data feed crawler

- Protocol declaration should have a git link to open source implementation of plug-in that programmatically define strict audit rules.

Step 2: Now every reporter can submit data according to a soft protocol. If data begin to be submitted without protocol definition a value of this data can be significantly lower. Thus auditors are encouraged to flag such documents. If false or malicious data come auditors have strong incentive to flag such data. All lazy oracles should be feed with self-descriptive tags so other participants will be able to faster process auditing.

Step 3: Auditors validate any given document by scripts.

As result reporters, auditors (where objective script based validation is possible) and curators (where subjective human made evaluation is more appropriate) has strong incentive to (1) bring important data, (2) curate and audit data conscientiously. Proposed approach doesn't has strong guaranty of truthfulness of data such as Augur does. Rather its correct to say that we can have strong assurance that data is correct. But it is cheaper, faster and significantly more flexible. The process is robust in its unstructured simplicity.

Step 4: Payouts for auditors are made. Payouts information is input information for cyber•rank. Using cyber•rank its trivial to filter feeds using plugins to give significantly more precise data with higher level of assurance.

Step 5: High quality data feeds (or lazy oracles) are available for a consensus engine of search engine (thus cyber•rank can be continuously improved) and everybody on the planet.

We call this type of oracles lazy because they don't require strict rules of validation by the expense of reducing the level of accuracy (but still enough to reason with high level of assurance). Also they are lazy because they don't require to think about monetization for participants rather consensus engine print rewards based on valuation of subjective contributions. This approach is superior to Ledgeys [] than reporters should sell data pieces using costly Ethereum storage.

The most obvious use cases for Lazy Oracles

- Block indexing
- Transaction indexing
- Balance calculation
- Identity crawling
- Token valuation
- Token rating
- Token description
- Crowdsale tracking
- Weather measuring
- Traffic measuring
- Business performance reporting

Worth to note that use cases are not limited with above mentioned. Data structure and validation rules are arbitrary. So we can think of it as general purpose tool for *popular structured public data* auditing and curating. Now we can bootstrap and index with amounts of useful information. But what about meaningful search results?

Dynamic Snippets

This can be thought as server less micro javascript applications that can dynamically take as input data from the following sources:

- a search query itself
- asynchronously from another search query.
- from a browser APIs
- from device sensory information
- from information about a user stored on cyber•chain
- from other blockchains
- from IPFS and IPNS
- from conventional HTTPS or WebSocket APIs.

Every application is CID written to search index and answer index. Before developing an application developer should target it for a specific semantic core before defining it. Semantic core and its statistic is publicly available in a blockchain. Developer shall need to develop application and submit a links to application for a targeted semantic core. Thus its up to developer to define what search queries a better fit to a particular application. Keep in mind that dynamic snippets are naturally compete for a higher position in search results. Dynamic snippets can be sorted by cyber•rank, and as result become trustful. Worth to note that developers can significantly reduce spendings on app infrastructure as dynamic snippets can be delivered through content-addressable distributed network. Proposed architecture make possible to design an application that don't require having all user's query in the index. Thus developers should not develop a snippet for every unique search request and has ability to program target semantic core. Implication of this approach is hard to overestimate. E.g. dynamic snippets combined with blockchain wallet make possible to shop right from search results.

Spam Protection - not ready yet

.. Voting power decay .. Vested Interest .. Broadband limiting

Possible Applications - not ready yet

Its hard to imagine what kind of applications can be built on top of proposed foundation. I'd like to mentions some outstanding opportunities which can be build using cyber•Chain and IPFS.

- Relevance everywhere
- Blockchain browser
- Offline search
- Smart command tools
- Self-aware robots
- Open search API
- Language convergence

Relevance Everywhere. Proposed approach enable social, geo, money or anything aware search. It is trivial to implement a search relevant to a particular identity using proposed algorithm. The more a user train a model the more behavioral data can be associated with she. This personalized information can be stored local for (1) faster retrieval and (2) offline access. During search request global and local request can be done in parallel. This queries can be easily merged adjusted on user-defined importance of personalized search for her. The same approach is able to fix vocabulary mismatch problem.

Blockchain Browser.

Actions in search.

Offline Search.

Smart Command Tools.

Self-aware robots.

Open search API.

Language convergence. Programmer should not care about what language do the user use. We don't need to have knowledge of what language user is searching in. Entire UTF-8 spectrum is at work. Semantic core is open so competition for answering can become distributed across different domain specific areas, including semantic cores of different languages. Unified approach creates opportunity for cyber•Bahasa. Since the Internet we observe a process of rapid language convergence. We use more truly global words across the entire planet independently of our nationality, language and race, Name the Internet. The dream of truly global language is hard to deploy because it is hard to agree on what mean what. But we have tools to make that dream come true. Its not hard to predict that the shorter a word the more it's cyber•rank will be. Global publicly available list of symbols, words and phrases sorted by cyber•rank with corresponding links provided by cyber•chain can be the foundation for emergence of truly global language everybody can accept. Recent scientific advances in machine translation [GNMT] are breathtaking but meaningless for those how wish to apply them without Google scale trained model. Proposed cyber•rank semantic core offer exactly this.

This are sure not exhaustive list of possible applications but very exciting though.

Incentive Structure and Distribution Mechanism - not ready yet

To make cyber•rank economically resistant to sybil attack and to incentivize all participant for rational behavior a system uses 3 types of tokens: CYBER (or cybers), CP (or cyber•power) and CD (cyber•dollar)

CYBER is a transferable equity token which is analog of STEEM. Intrinsic value of CYBER came from ability to convert it to CP.

CP is a non-transferrable equity token which is analog of SP in Steem. Intrinsic value of CP came from the right to (1) write to an index according to a bandwidth limit, (2) rank objects, (3) promote objects (4) make consensus decisions. CP can be converted to CYBER in one year.

CD is a debt token with relatively stable value which came from ability to convert it into CYBER within 3 days by the price submitted by witnesses and calculated according to cyber•rating methodology [] (don't confuse). 1 CD tracks $1/10^{12}$ of *provable* blockchain economy.

Reward Pool is defined as 100% of emission and split among the following groups:

Infrastructure Reward Pool

- Witnesses - 10%
- Investors - 10%

Indexing Reward Pool - 30%

- Reporters - 10%
- Auditors - 20%

Linking Reward Pool - 60%

- Responders ~ 20%
- Trainers ~ 40%

Our implementation also offers an incentive for CD holders. They receive APR on holding according to rate defined by witnesses.

As our primary focus is developers of decentralized and distributed applications we offer convenient way of revenue generation in addition to possible revenue on dynamic snippets development. Every search and answer transaction can define a smart contract with up to 6 beneficiaries with distribution of potential reward from this particular action. This creates a solid foundation for bootstrapping application ecosystem across conventional web, mobile app stores and upcoming VR platforms.

Conventional advertiser are also has a tool to participate. Any link can be promoted thus increasing cyber•rank. Revenue from promotion is burning thus decreasing supply of CYBER and bringing value to all shareholders of a system. We do believe that this approach is fair and don't harm user experience but open opportunity to compete with conventional search ad a bit.

Virtual loop of the business models for our decentralized autonomous organization is pretty simple:

More indexing => More people search => More developers build => More people earn, rank and promote => Better infrastructure => More indexing

A network starts from 100 000 000 tokens

- Crowdsale - 80%
- cyber•Fund reward - 5%
- Founders reward - 5%
- Sharedrop 5% -
- Development Reserve - 5%

Than a network prints every block xx tokens

There is one problem with proposed incentive structure. We call it *language incentivization bias*. In the core of cyber•chain is quadratic voting. System needs it to effectively incentivize participants for quality ranking. But that natively leads to weak incentives across different language groups. E.g blockchain Golos was deployed as Russian alternative to Steem because Russian posts acquired only 0.2% of rewards though providing 10% of content. The idea of deploying cyber•Chain is great if it can be truly global from the start. The only way to overcome this bias is a global deployment from day 0, because otherwise we need significantly increase complexity of the reward system. We offer good incentives for translation of this white paper to 50 languages worldwide as well as call for action to all blockchain communities across the globe.

Extensibility - not ready yet

Currently our implementation has the following functionality available for application developers.

- Custom Operations ...
- Plugins ...
- Escrow ...
- Private Messaging ...
- Dynamic Account Permissions ...

A system Our design allow to implement intensive on-chain calculations in case of 17 of 21 delegates accept a hardfork. This is for a purpose of keeping a protocol from bloating.

The following possibilities can be available in a future:

- Sidechains ...
- State Channels ...
- Smart Contracts ...

Performance and Scalability - not ready yet

3 seconds block ...

Our current node implementation use memory for faster performance. As of now all blockchains are about 1 bln immutable documents which size is about 200 GB with avarage tx 200 kb. We need to store all hashes which are on average 64 bytes long. We estimated that storing in the index all interlinked CIDs and votes is roughly the same as storing all raw blockchain data.

Currently it is not hard to assemble a hardware with 2 TB of memory which enable to put into blockchain index all blockchains tenfold. Such appliance can cost about \$25k so we can think of it as affordable for those who are seriously want to be involved in a project. Thus overall costs of basic network infrastructure can be about \$500k.

Scalability improvements include:

- Hardware part. This year Intel 3dx Memory arrive so it
- Implementation optimizations. It is not so hard little harm to performance to use fast M.2 PCI-E X4 slots for SSD storage which are fast. This enable to scale a node for as much as 100 TB that is enough for years of developments given as input current blockchain size growth.
- Consensus optimisations. The future is parallel ...

...

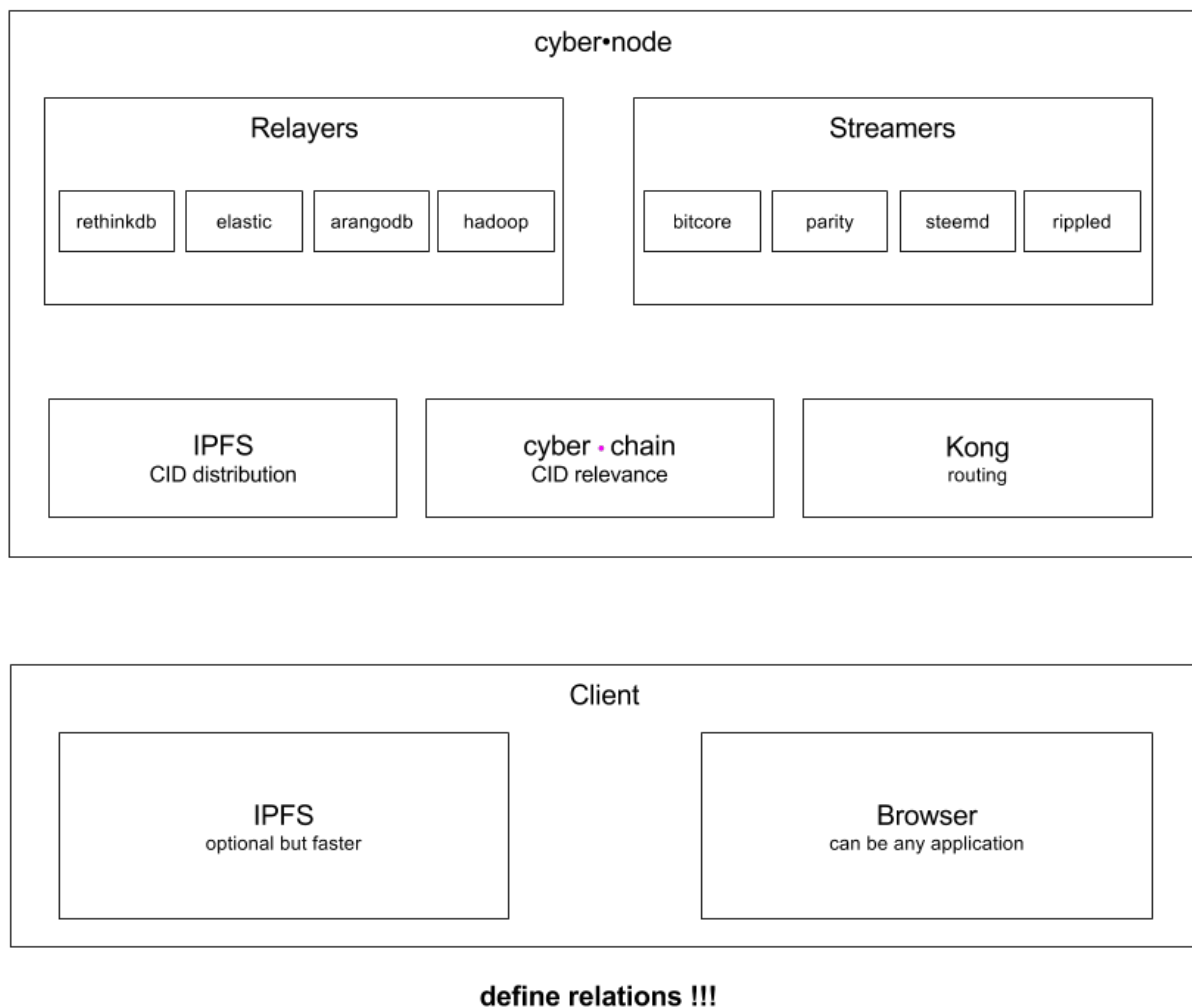
Self Governance - not ready yet

...

Search Appliance - not ready yet

Fo bootstraping we are going to offer well tested open source specs for hardware configuration of commodity computer which cost around \$10k and is able to participate and earn by itself.

Software is open source. Cybernode



Hardware is open source. Enterprise.

...

The power of cyber•Chain - not ready yet

Key purpose of our proposed design is not just replicate abilities of existing search engines which return only links but enable answering new class of question:

- How much value of X do I possess now?
- What probability of event Y?
- What packages do I need to install in order to improve ROI on available resources?

Our proposed design has all necessary components to bootstrap a markets for a new generation of answer applications.

Proposed economics model disintermediate conventional ad model there users are sold to advertiser and enable any business or people or robot benefit from pure peer-to-peer interactions which bring value for every involved participant.


Worth to note that minimalistic but continuously self improving approach provides necessary tools for emergence of economically rational robots. Blockchain technology enable creation of devices which are able to earn, store, spend and invest digital assets by themselves. What is needed is verified information about state of reality evaluated in

transact-able assets. ...

No Noise Filtering. Blockchains are highly optimized databases. Every transaction cost money so they are neutrally protected from spam and contain only data that is really matter. That reduce noise filtering to nearly zero level. Also blockchains provides self-authenticatad data. Structured raw data about blocks and transaction is available for everybody. This fact also reduce consumption of resources and make extraction of data

How Google can see it

How it can be indexed directly from a blockchain

	<table><tr><td>author</td><td>dana-edwards</td></tr><tr><td>title</td><td>Real computers vs Turing machines</td></tr><tr><td>children_rshares2</td><td>3,642,226,283,779,252,593,623,040,000</td></tr><tr><td>permalink</td><td>real-computers-vs-turing-machines</td></tr><tr><td>json_metadata</td><td>{ "format": "markdown", "links": ["https://steemit.com/crypto-news/#dana-edwards/automata-theor", "https://steemit.com/automata/#dana-edwards/prof-wolfgang-thc", "http://hackingdistributed.com/2016/06/18/analysis-of-the-da",], "tags": ["computation", "ethereum", "thedao", "cybersecurity",], "image": ["https://img1.steemit.com/0x0/https://upload.wikimedia.org/",], "app": "steemit/0.1" }</td></tr><tr><td>scanned</td><td>2017-01-10 03:35</td></tr><tr><td>allow_replies</td><td>1</td></tr><tr><td>url</td><td>/computation@dana-edwards/real-computers-vs-turing-machines</td></tr><tr><td>net_rshares</td><td>764,457,683,485,000</td></tr></table> <p>What is a real computer? ===== A real computer can be defined as a computer which can exist practically rather than merely theoretically. The physics as we know it requires all computers have a limited amount of memory and a finite numbers of states. We can conclude from this that real computers may be modeled as finite automata. In my previous post I described how finite automata work, and how it is a finite state machine (this amount of states is not within a limit).</p>	author	dana-edwards	title	Real computers vs Turing machines	children_rshares2	3,642,226,283,779,252,593,623,040,000	permalink	real-computers-vs-turing-machines	json_metadata	{ "format": "markdown", "links": ["https://steemit.com/crypto-news/#dana-edwards/automata-theor", "https://steemit.com/automata/#dana-edwards/prof-wolfgang-thc", "http://hackingdistributed.com/2016/06/18/analysis-of-the-da",], "tags": ["computation", "ethereum", "thedao", "cybersecurity",], "image": ["https://img1.steemit.com/0x0/https://upload.wikimedia.org/",], "app": "steemit/0.1" }	scanned	2017-01-10 03:35	allow_replies	1	url	/computation@dana-edwards/real-computers-vs-turing-machines	net_rshares	764,457,683,485,000
author	dana-edwards																		
title	Real computers vs Turing machines																		
children_rshares2	3,642,226,283,779,252,593,623,040,000																		
permalink	real-computers-vs-turing-machines																		
json_metadata	{ "format": "markdown", "links": ["https://steemit.com/crypto-news/#dana-edwards/automata-theor", "https://steemit.com/automata/#dana-edwards/prof-wolfgang-thc", "http://hackingdistributed.com/2016/06/18/analysis-of-the-da",], "tags": ["computation", "ethereum", "thedao", "cybersecurity",], "image": ["https://img1.steemit.com/0x0/https://upload.wikimedia.org/",], "app": "steemit/0.1" }																		
scanned	2017-01-10 03:35																		
allow_replies	1																		
url	/computation@dana-edwards/real-computers-vs-turing-machines																		
net_rshares	764,457,683,485,000																		

_Free Market of Indexing.

Free Market of Linking. After all we have recent advances in machine learning enable to reason about a piece of data quite well. All this algorithms require enormous highly distributed computation which as nearly impossible to achieve in a trestles consensus computer. With current state of blockchain technology implementing this algorithms inside decentralized computational network seems unfeasible. What we need is to find a way to *outsource* this computation for the entire community.

_Free Market of Curation and Auditing

Open Semantic Core. Click-through information is stored on chain. Every time a user follow a link positive voting transaction is broadcasted e.g. with grade 1. Voting on a protocol level is a number in a range from -100 to 100. Thus application developers has a tool to implement different grades for different kind of interactions. Such design is crucial to train the model and acquire a data about search popularity of semantic core and its volume. Currently search engines are very careful in revealing this information because this information is the most important part of ranking. We want to change that. Every time a user click on a snippet developer earn fair portion of emission and on chain model is trained. Application acquire the more rank the more rank acquired by its links. The more cyber*rank acquired - the more revenue streams for application developer.

Self-Hosted Search API. Available for everybody \$5 - 1000 запросов у Гугл. 30 requests a month for avarage users. The on should not store we entire index. It is possible to process (let us say without) some chain starting from a trusted part and cache only the most relevant and new documents thus limiting volume of stored information. This approach enable effective using of offline search

Conclusion

We describe and implement a motivated blockchain based search engine for permanent web. A search engine is based on content-addressable peer-to-peer paradigm and use IPFS as a foundation. IPFS provide significant benefits in terms of resources consumption. CIDs as primary object is robust in its simplicity. For every CID cyber•rank is computed by a consensus computer with no single point of failure. cyber•rank is a combination of Google's pagerank and Steem's rewards system. cyber•ranks is resistant to sybil attacks and is computed based on interactions with a graph of CIDs and it's internal relations. Embedded smart contracts offer fair compensations for those who participate in indexing, linking, auditing and curation process. Primary goal is indexing of peer-to-peer systems with self-authenticated data either stateless, such as IFSS, DAT, GIT, BitTorrent, or stateful such as Bitcoin, Ethereum and other blockchains and tangles. Proposed market of linking offers necessary incentives for outsourcing computing part responsible to find a meaningful relations between objects. Proposed market of curating and auditing creates essential incentives for ranking high quality links and objects. Dynamic snippets in search results make possible functionality necessary for the next generation search. Lazy oracles enable indexing of structured publicly verifiable data feeds in a highly competitive environment. A source code of a search engine is open source. Every bit of data accumulated by a blockchain is available for everybody for free. Performance of proposed software-hardware implementation is sufficient for seamless user interactions. Scalability of proposed implementation is enough to index all self-authenticated data that exist today. The blockchain is managed by a decentralized autonomous organization which function under DPOS consensus algorithm. Thought a system provide necessary utility to offer an alternative for conventional search engines it is not limited with this use case either. The system is extendable for numerous applications and make possible to design economically rational self-owned robots to spawn a market of AI outsourcing.

References - not ready yet

[QGTC] <https://www.quora.com/How-many-pages-is-Google-crawling-every-day> PR [] RALF
<http://merkle.com/papers/DAOdemocracyDraft.pdf>

ENIGMA <https://www2.deloitte.com/content/dam/Deloitte/uk/Documents/Innovation/deloitte-uk-blockchain-full-report.pdf>

RTB https://en.wikipedia.org/wiki/Real-time_bidding