Neuledge

Bring Neural Context to the World

Moshe Simantov
moshe@neuledge.com
www.neuledge.com

Abstract. This paper introduces a peer-to-peer knowledge-base platform that can represent everything in the world in a large association graph. Inspired by neural networks, a decentralized graph-database can denote all the knowledge of the human world in such way that is very close to the way we already think and store information in our brain. Unlike native languages, the data in the graph is represented by simple association-arcs, eradicating any duplexity of information. This methodology does not enforce any restrictions on the details each piece of information can have. Furthermore, it is language agnostic and eliminates semantics ambiguity. In one of the most prominent use-cases, we can use this graph for artificial intelligence algorithms which need to interact and understand the human world blindly. The peer-to-peer network enables everyone to access the database, add new records and search for them. All the operations within the database are protected by the blockchain technology using a unique cryptocurrency, which is dedicated for this purpose.

1. Introduction

Language gave humans the ability to speak and to cooperate on a large scale. While there are over 7,000 different languages in the world,[1] most of them share the same principles such as words, sentences, nouns, and verbs. Those principles give us a convenient way to communicate and share our thoughts one with each other, but they also narrow down our communication with a limited amount of words and have strict rules. There must be a better way to collect and share information.

1.1. Problem Statement

A single word or even a sentence can represent different things to different beings. Given that distinction between our brain and our current languages, we can never be sure that our minds share the same ideas. Our data is unreachable due to lack of organization, language diversity and misleading prioritization of humans over machine learning algorithms. Our data, by definition, should be less ambiguous and more accessible. Machines should have an easy way to search and understand our data and its storage space should be more dense and immune from duplications. Instead of creating massive data pools and then index them, we can store the data itself in such way that is already searchable and indexable.

1.2. New Way of Communication

Any thought that we have can be mapped as a single node (vertex) in a graph. If that thought reminds us of something, an arc is drawn between the current node and the associated node. Sharing an idea is equivalent to sharing a single node in the graph. When several minds relate to a concept, all of the nodes will be entirely separate one from each other, but it is still possible that some nodes, representing our literal words, are well-defined in both minds.

Using our native language as a bridge in the graph gives us a way to build a complete world of thoughts and ideas that we can share and learn. As we evolve, we may use well-defined nodes instead of our native words so that the communication can be faster and more efficient. We can use some machines to translate the things we are talking about and convert them to nodes in the graph. The more data transferred and switched to the graph representation, the easier it is to understand and learn how to bridge between our native language to the graph.

2. Universal Neural Graph

We define a Universal Neural Graph (UNG) as Directed Acyclic Graph (also known as DAG) where each node is constant and referenced using a well-known hash function. Each node will have a type and a body. The type differs between different types of nodes on the graph while the body contains all the information required to attach it to a graph, including its value or its arches. This design ensures that with a single node-id, we could validate and rebuild the whole graph structure.

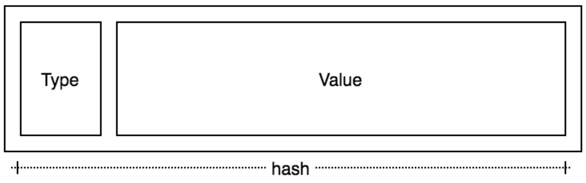


Figure 1: Node structure

2.1. Data Representation

There are two main types of nodes, data-nodes, and anonymous-nodes. Data-nodes are always on the edge of the graph and represent a real data from our day-to-day life. This data can be anything from numbers, words, images, time, places, and so on. Those nodes are the only nodes that can hold a real data on the graph, and we can "label" them by their content. In contrast, the anonymous-nodes are only capable of representing a connection or a relationship between two other nodes. They cannot represent any new data by themselves and therefore label-less and called "anonymous".

By definition, the graph is free from data-duplications. Since two equals nodes with the same content will generate the same id. Changing node value is also impossible and require the creation of a new node. To prevent changes and ambiguity, we define each node-type as broad as possible. For instance, all numbers will be represented only as 128-bits floating number, usually known as `__float128`, which include most day-to-day numbers. This method will prevent duplication when referring the same number in different formats. In the same way, keywords will be used only in lowercase without punctuations.

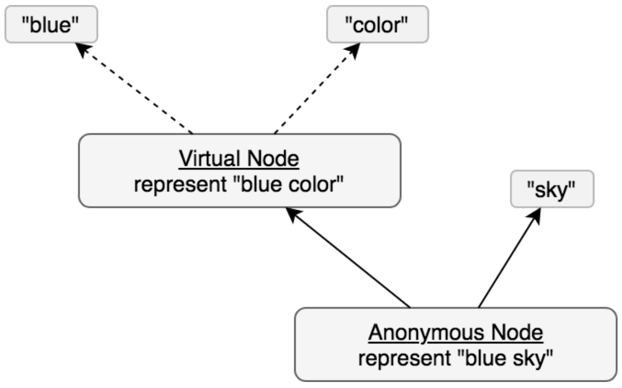


Figure 2: Neuledge representation of "blue sky".

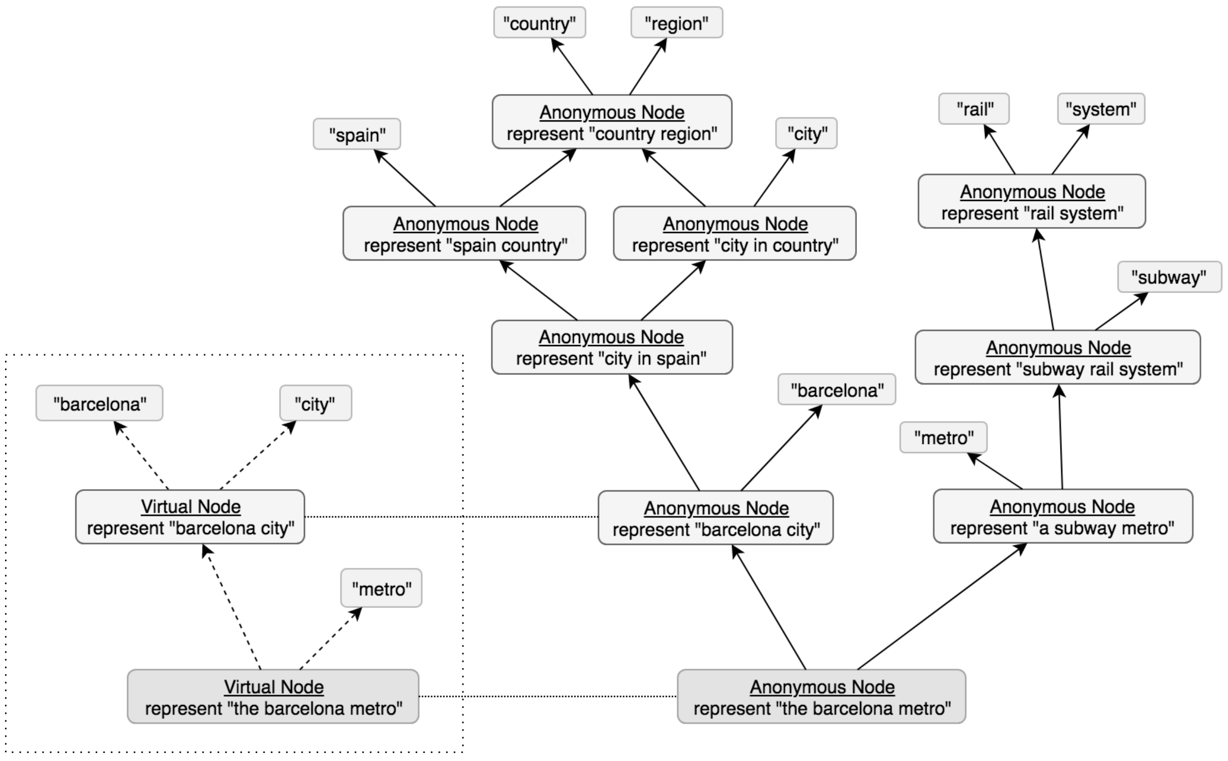
2.2. Searching

Most of the nodes in the graph will probably be anonymous-nodes that represent many relationships and hierarchy in the world. We defined anonymous-nodes as nodes, but it is more accurate to refer to them as arcs or links in the graph. In Figure 2, the main node in the graph represents a “blue sky”. A more specific description can be a “sky” with the “color-blue” association. If we want to search for all the nodes related to “color” and “blue”, we can search for anonymous-nodes connected with the specific node “color-blue”. Although anonymous-nodes are restricted to have only two connections, this method gives us the ability to evaluate all the possible connections for a single word, idea or data in the graph. Once we find the node we are looking for, it is easy to find all the links to-and-from this node.

2.3. Virtual Nodes

While browsing the graph can be easy, creating a new connection and choosing the right association between thousands of anonymous-nodes can be very frustrating. Moreover, data can be changed, deprecated or even be wrong. We can not afford to update all the nodes in the graph for each change. Virtual nodes can solve that problem by defining a more flexible association between the nodes.

A virtual node is a node that can indirectly refer to multiple anonymous-nodes. Virtual-nodes and anonymous-nodes are very similar; both have two parent-nodes. The only difference is that virtual-nodes simultaneously refer to multiple anonymous-nodes. They potentially connect to any appropriate node at any level in the graph. Thus, although we choose two parents, we never point to a specific node that could be obsolete later on. This behavior increases the accuracy of our information and adds more chances that other people will use our nodes in the future.



"country"
"region"
Anonymous Node represent "country region"
"spain"
"city"
"rail"
"system"
Anonymous Node represent "rail system"
Anonymous Node represent "spain country"
Anonymous Node represent "city in country"
"subway"
Anonymous Node represent "city in spain"
Anonymous Node represent "subway rail system"
"barcelona"
"city"
"barcelona"
"metro"
Virtual Node represent "barcelona city"
Anonymous Node represent "barcelona city"
Anonymous Node represent "a subway metro"
"metro"
Virtual Node represent "the barcelona metro"
Anonymous Node represent "the barcelona metro"

Figure 3: Demonstrate how virtual nodes represent real information.

2.4. Multilingualism

One of the exceptional features of UNG is the ability to index the same nodes or the same information in different languages at once. By using the virtual-nodes, every node can refer to the linguistic meaning of the word instead of the literal word. While this pattern gets the same meaning in different languages, it also keeps a reference to the original words from the source. The virtual-nodes ensure that the real meaning of the words will always improve as time goes on.

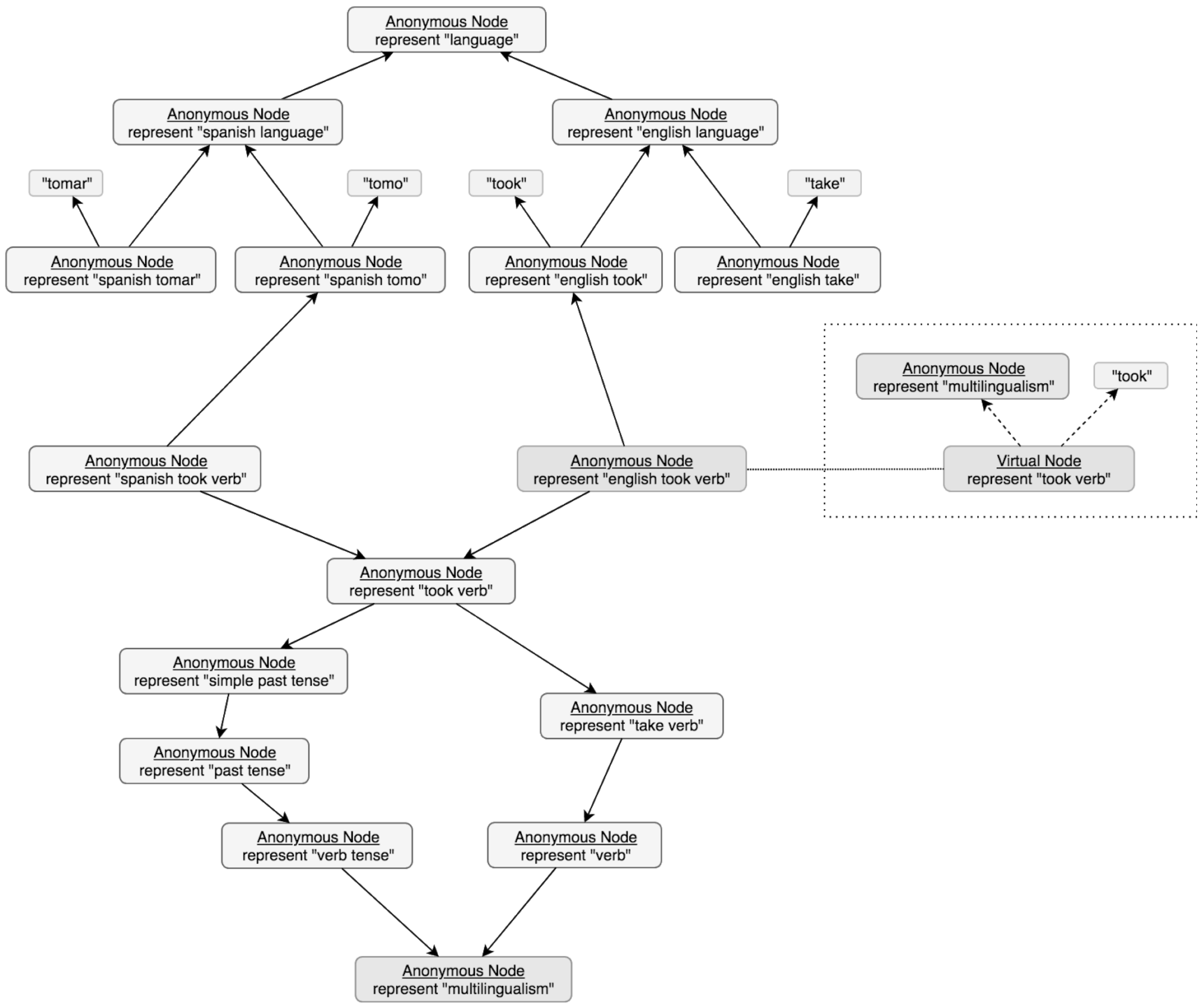

Anonymous Node represent "language"
Anonymous Node represent "spanish language"
Anonymous Node represent "english language"
"tomar"
"tomo"
"took"
"take"
Anonymous Node represent "spanish tomar"
Anonymous Node represent "spanish tomo"
Anonymous Node represent "english took"
Anonymous Node represent "english take"
Anonymous Node represent "multilingualism"
"took"
Anonymous Node represent "spanish took verb"
Anonymous Node represent "english took verb"
Virtual Node represent "took verb"
Anonymous Node represent "took verb"
Anonymous Node represent "simple past tense"
Anonymous Node represent "take verb"
Anonymous Node represent "past tense"
Anonymous Node represent "verb tense"
Anonymous Node represent "verb"
Anonymous Node represent "multilingualism"

Figure 4: Demonstrates how to create a graph that supports multilingualism
(Some nodes and connections have removed for simplicity).

2.5. Automation

Most of the written knowledge come in forms of sentences. Each source can be represented as a node in the graph and can connect to a URL, a book or even a person as a reference to this source. It is possible to generate a graph representation for each sentence and then link each sentence to source-node. Users will be able to search information on the graph, based on specific sources or keywords in the same way we used today on the search engines. SyntaxNet, for example, is an open-source program that can output a hierarchy tree from any given sentence. The script already outputs a graph that is very similar to UNG representation and can be used to create nodes based on the script output. [2]

```
Input: I found a website to post AI tutorials .
Parse:
found VBD ROOT
 +-- I PRP nsubj
 +-- website NN dobj
 |   +-- a DT det
 |   +-- post VB infmod
 |       +-- to TO aux
 |       +-- tutorials NNS dobj
 |           +-- AI NNP nn
 +-- . . punct
```

Figure 5: SyntaxNet example output for given sentence.

3. UNG Blockchain

As Neuledge will grow and become more prominent, it will become impossible to give the same importance to all the connections in the graph. Some associations will appreciate more than others and will be appraised according to the real value they represent. Using the blockchain and the economic forces, we can ensure that it each node value is going to be fair and accurate.

3.1. Nodes as Assets

The UNG Blockchain will allow users to assign a value to each new node they create. Nodes that will be worth more will be preferred over others, and users will be able to invest or even gamble on the more promising nodes. Each investment will buy a share in the node but will also require a partial deposit in the node's parents. Those deposits can increase the parent's node values and therefore create a profit for the parent's nodes shareholders. The proportions between the investment and the deposit will be 50:50, giving 25% deposit for each parent-node (Figure 6). This division will make a dynamic value for each node that can be changed merely by the belief that this node will be used more than others in the graph. Please note that this credit division can only occur once, on the invested node, and there is no need to propagate it recursively through all the parents in the graph. When a user requests to withdraw his investment, the funds received consist of his exact share in the node as well as the deposited amount from the parent nodes.

Example. Let "Z" be a node connected to the parent-nodes "X" and "Y". Bob decided to invest 200 coins on node Z, and therefore only 100 coins (50%) will be credited for node Z. The other 100 coins will split equally between nodes X and Y, without giving any shares to Bob. If before the investment, node Z was equal to 400 coins, node Z now equals 500 coins, which gives Bob 20% shares in a node. Let's say Bob investment succeed. The value of node Z raised and its current value is now 1000 coins, while the deposits on the parent-nodes remain the same. When Bob decides to withdraw his funds, he will get 300 coins in total: 200 coins for his 20% share in node Z, and 100 coins from nodes X and Y combined.

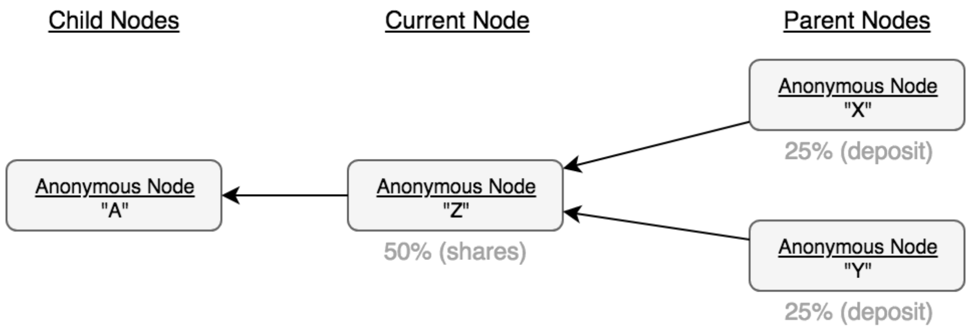


Figure 6: Nodes hierarchy with the divided credits.

A profit can be made only from the investor's deposit part from each node and not from the shares part. This limitation was made to protect the investors from price manipulations on the nodes values. Getting profit from any parent-nodes is only possible if there are enough deposits on the nodes so that any parent-node investor can get at least the initial 50% investment back. That is to say, investment on any rooted node should always be lossless and will consider a solid investment.

3.2. Transactions

Investing in each node require an independent currency which is scalable and fee-less. It also has to be decentralized and resistant to spam attacks. All of those requirements can be accomplished via the following “mergeable” blockchain.

As any blockchain, the mergeable-blockchain will start with the genesis block, which is a hard-coded block embedded on all peers devices. From there, any peer can create a new transaction and attach it to the latest block via signing the transaction with the latest block hash-value. After a short amount of time, all the known valid transactions for the latest block get collected via their hash-values and therefore, creating the next block on the chain. Each new transaction will now use the new block hash-value as its referenced block and by that validating the correctness of this block as well.

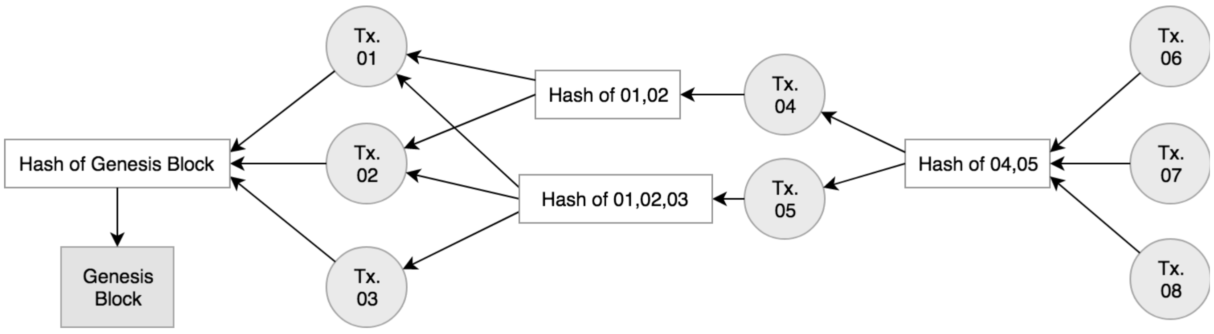
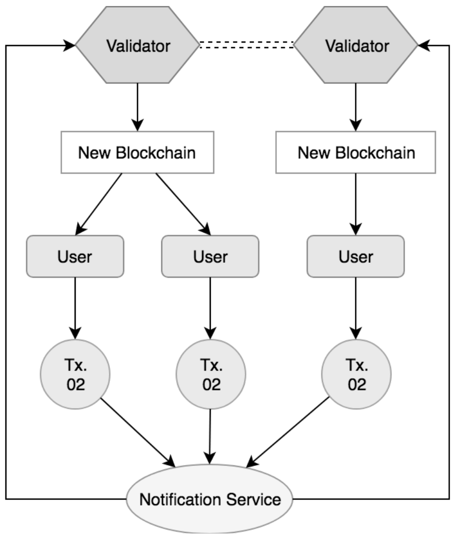


Figure 7: Blockchain made from transactions.

A little Proof-of-Work (PoW) will be added for every transaction to prevent spam and abuse. In case of uncertainty regarding the next block on the chain, we choose the block with the most valid transactions. Transactions that in one way or another are not included in the latest blockchain can be joined to the next blockchain as long there are no conflicts. While most of the transactions will join the blockchain, eventually only the malicious transactions will stay outside, making any reasonable peer forgetting them.

3.3. Validators

A validator node is a node which can process a significant amount of transactions and generate every few seconds a new hash of the next blockchain. The validator will order the transaction in the order they came in and will add only the valid transaction in a first-in-first-out order (FIFO). Other validators should connect to each other and make sure they are in sync, pushing any missing transaction to the other validators as well. The purpose of the validators is only to add order to the system while leaving the real power to the masses. That is to say, for each transaction, each peer can choose which of the validators has the next valid block in the chain.



3.4. Maintaining Consensus

At any given point, an attacker can publish a double-spend attack, with at least two transactions that contradict each other. This attack will force the network to choose each time which transaction to add to the blockchain. If two different validators select different transactions, their chains will never be able to join back together, as one of the transaction has to be invalid. The network will automatically choose the chain with more transactions in it. If both chains have an equal amount of transactions, it will wait for the next chain until one of the chains have more transactions. This process should not take too long as each validator will choose their favorite chain, causing all the related transactions to follow.

Although the double-spend attacks will fail, it can affect the level of trust in the network. Collaboration between the validators can be a win-win situation for everyone. Assuming the validators release a new blockchain every 10 seconds. Each validator can decide to share the next chain with other validators a few seconds before it releases it to the public. Then, all the validations will have the chance to coordinate their positions right before the chain is published, lowering the probability of a conflict.

3.5. Storage Costs

Adding new nodes to the graph requires a transaction with at least four fractions of a coin so each parent will get one fraction as a deposit. While those fractions can be worthless, each transaction will also require a little Proof-of-Work which prevent spam and abuse in the system. As we see next, this PoW is the minimum cost for each new node in the graph.

Consider the following scenario, when an attacker creates and stores a large graph offline. This new graph has many nodes and child-nodes that are yet unknown to the network. Each node has a size limitation, but with a large number of nodes, a very large data-store could be made. The attacker then decides to add a new transaction to the network with only the edge-node of his large graph. This transaction will deposit 50% to the node's parent-nodes and therefore will force each peer to download and store this edge-node locally. If the transaction amount is worth less than the cost of the PoW, no one will bother to invest on the node’s parents to gain a profit from the attacker, leaving this edge-node as the only new node that the network has to store. That means that any other node that is not in the blockchain can be forgotten. If other users invest in those nodes, they will also require resharing the content of those nodes so the transaction can be added to the chain.

5. Neuledge Ecosystem

As a peer-to-peer network, Neuledge requires several services and protocols to work in collaboration as a whole ecosystem. Some of these services were inspired by existing projects such as BitTorrent, Ethereum, and others, in order to save time and effort involved in developing the complete services from scratch.

5.1. Universal Hash Dictionary (UHD)

Same as known Distributed Hash Table algorithms (DHT), this service serves a peer-to-peer dictionary that can store data and answer queries for any known hash-value. It will be used to store all the graph-nodes data, as well as the transactions and the blocks in the blockchain. To lower the response time of the service, all the values in the dictionary will have a maximum size. Larger values could be split into small fragments and be joined together with a mapping index-table (Figure 9). The fragmentation process can repeat multiple times, so there is no real limit for the value size.

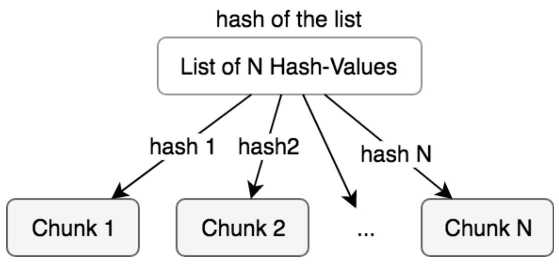


Figure 9: Large value split into small fragments.

5.2. Universal Messaging System (UMS)

The messaging service allows any client to register to a channel and receive real-time messages for this channel. A channel-id could be any output from the hash function and can be used to broadcast new nodes or transactions. Using the UHD service, each peer will search for peers listening for given channel and build a Mesh-Network by opening a session with several other peers. Each message will require having a small Proof-of-Work, to prevent spam and abuse. Only new and valid messages will transfer to the next peers.

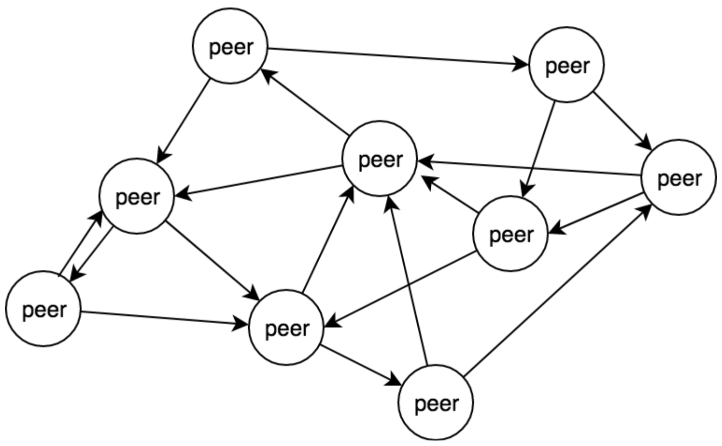


Figure 10: A peer-to-peer Mesh Network.

5.3. Blockchain Validators

The validators peers will handle all the hassle of gathering transactions and digesting them into a new block. Each peer will continuously query them with the latest block in the chain, trust or validate it and use it as a reference when creating a new transaction (Chapter 3.3). The validators service could be free, but in the future, some faster validators can charge a monthly subscription fee, for example, to cover the processing expenses. We leave this decision for every individual validator to decide.

5.4. Neuledge End-Client

The Neuledge End-Client (The Wallet) is a peer-to-peer client that continuously listens to several validators and keeps a record for each node's value. It will validate each new block from the validator and share its current block with other peers. The storage consumption for this client can be relatively low, as there is no requirement to hold each block in the chain. It is sufficient to keep track of the latest block as well as all the accounts and nodes balances. In case of conflict, as long as one peer holds the data, the UHD service can fill and bridge the gap.

5.5. Neuledge Search Service

One of the main features of the graph is to search for all the children for a given node, sorted by their node's values. Those queries can be very demanding as it requires from each peer to hold and index the entire graph database for every single request. Using sharding and cluster methodologies, we can distribute the database to multiple peers, letting an only small fraction of the network process each query. Each peer will randomly select some nodes in the graph and continuously index every single node that's related to them. With a large number of peers, every graph-node will be well indexed. Sending a query can be done via the UMS by selecting one of the root node's parents as a channel.

6. Conclusion

We have proposed a new way to store knowledge using a graph database. The graph holds anonymous nodes, and each node will represent a single concept in our world. We show that data can be represented without duplications and can be used simultaneously by multiple peers. Besides, the data can be stored and understood in various languages, without losing references to the source's original words. To validate and prioritize the data, we used a decentralized blockchain system which gives the market the ability to determine the value of each node. Using a scalable and feeless cryptocurrency, users will invest and make profit from new data entered into the systems. Publishing data to the graph will require a minimal amount of Proof-of-Work. Many services and tools can build on this platform including search engines, storage services, and databases.

References

[1] Simons, Gary F. and Charles D. Fennig (eds.). 2018.
Ethnologue: Languages of the World, Twenty-first edition. Dallas, Texas: SIL International.
http://www.ethnologue.com.

[2] Aniruddha Tapas. 2017. How to use SyntaxNet.
https://github.com/Aniruddha-Tapas/How-to-use-SyntaxNet