

"you're not ready for this"

Digital Megastructure

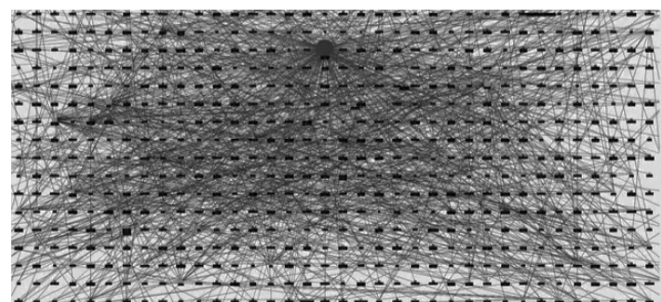
Editorial Board:
vicktor reis(vicktor@papel.com)

New developments in AI, Knowledge Management, Payment Processing and Information Processing have allowed us to construct knowledge structures that are getting too large for a single human to comprehend. The complexity of projects and the information entropy received is getting larger while we're still using tools designed for "paper-like" knowledge structures such as spreadsheets and presentations.



PRIMITIVES

We are still using the old primitives to manage information like: Language, Writing, Flashcards, Zettelkasten, Financial Systems, Double Entry Accounting, Mail, Corporations, Audits, Calendar, Spreadsheets, Cryptography. As well as using new tools like: Search Engines, Real Time Chats, Mobile Phones, Decentralized Governance(Smart Contracts, code is law), AI(LLMs, Agents), The Internet, The Web, E-Mail, Cloud Computing and more.



TECHNOLOGY

This paper introduces a theoretical system that is used to manage large information systems, that is, a system that has a "goal" and is made out of "agents" (human or not), information, systems, procedures, protocols, hierarchies, access system, communication structures, culture and more.

These are called GAN(Goal Accomplishing Network)



Similarly to crypto Decentralized Autonomous networks (DAOs), a GAN would be a digital "network" of individuals, tools, workflows, code, automations, audits, files, links, permissions and money that work in order to accomplish or maximize a given goal.

This goal might be short-lived such as creating a logo, or the goal can be long-lived like creating a startup that should seek to raise money and then revenue and protect its existence in the future.

PRICE

An network is primarily run by "burning" money, which means paying people in order to act for the network to accomplish its goals, step-by-step, each step in accomplishing the goal of the network is rewarded like: searching, selecting, listing, navigating, teaching, showing, creating, mutating, questioning, clarifying, commanding, leading, allocating and more.

The "price" of the network is: $market_price * velocity$. So you either get a high price for being fast, or because of the current market price is high, and you get a low price for being slow and if the current market price is low.

INTELLIGENCE MARKETPLACE

An open market of bids for actions such as "create an account on website Y" or "get X information" or "check for this information" will have an associated reward, in which workers who are interested in them will "bid" their actions until a best-case is found and the worker is rewarded. Given the size of each action will tend to be "atomic", payouts might range from 0.00000001 to infinity. Some tasks might be automatically picked up by bots who will try to "win" the contest thus automating tedious work.

The best case can be decided by a consensus algorithm, a weighted consensus algorithm or basically by the decision of the "owner" of the network. As the network scales, the owner will have to delegate decision making in order to move into doing other things. In the network, we will have a "hierarchy" of agents in which they might be promoted or demoted, based on the owner's perception of their work, based on the network's perception of their work or third-party audit system.

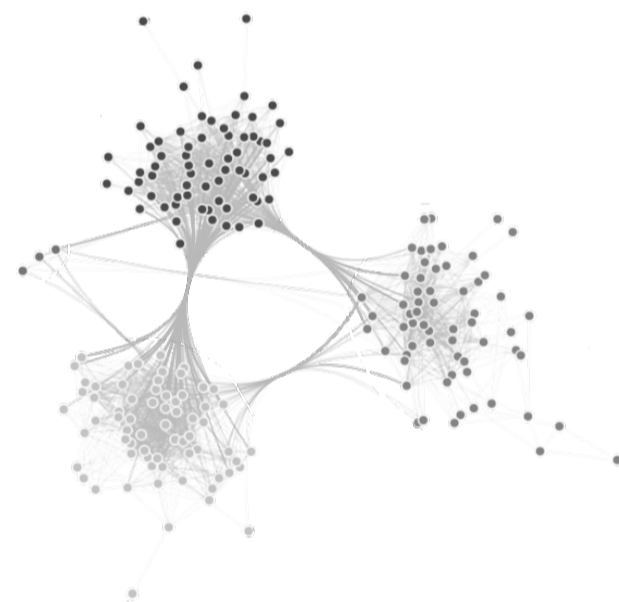


VELOCITY

A "velocity" or "speed" is defined as a way for the network to speed-up or slow-down its operations. When the velocity is low, the bid price will be on average, lower, making it take longer for workers to bid against, thus making the network cheaper to run. When the velocity is high, the bid price will be on average, higher, making the workers more likely to bid.

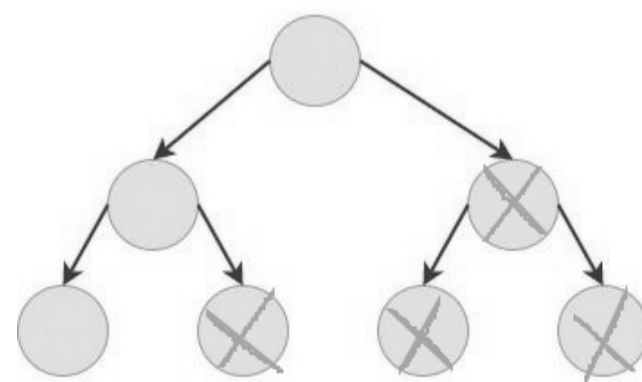
DIRECTION

The "direction" in which the network takes such as where it should allocate resources is defined as *attention*. This is a mechanism to price tasks higher, which means workers will be nudged to work where the attention is and not anywhere else.



The attention in the network above is the darker nodes, where tasks are more profitable.

Attention is used to "nudge" the network into completing its mission by the creator of the network, an appointed CEO, a consensus mechanism or by third party auditors. This ensures the agents are spending resources in the correct direction.



One example of such would be an network that would put in its mission that the only person able to control the network is the first child of person

X. The network, in order to do anything, will have to prove that the commander of the network is indeed, the child of X, otherwise the network will be "dead", which means it'll only be able to spend a fraction of its reserve in order to give the command of the network to the person specified in the official documents. This is called a "Zombie network".

PROGRESSION

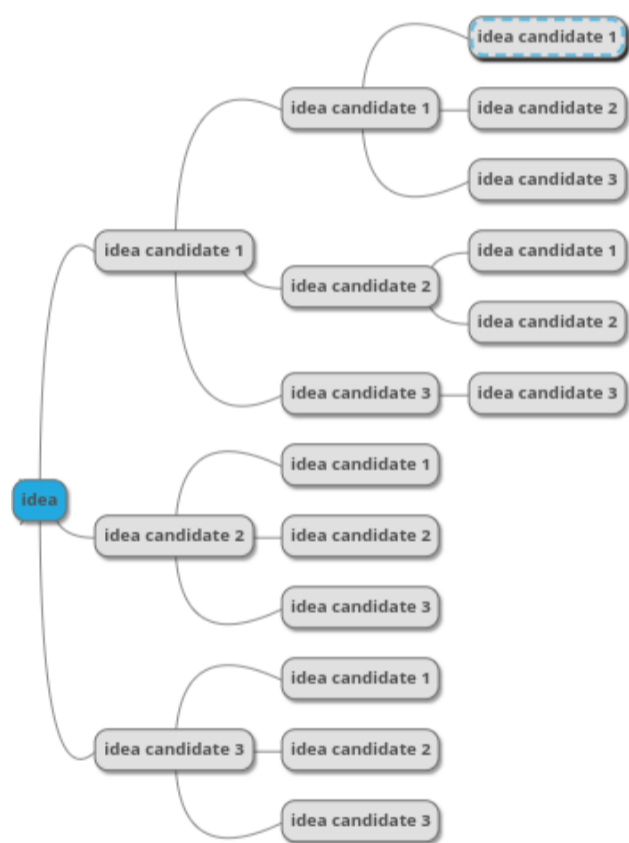
Progression in the network means steps towards its goal.

We might think that the network "works" not by the normal time but by the "thought time", each time it takes to a thought to be selected.

In order to know which "idea" is selected in the chain we have a "next-thought marketplace" where people select the most likely idea to be selected, similar to John Maynard Keynes's Beauty Contest.

The network will reward agents who correctly predicts the next idea in the "thought chain" with money. A "staked" money will be required to submit candidates to avoid spam, losing money if they either submit or vote in the wrong candidate.

A network might make "shadow chains" which then are submitted in the "main chain" this is when an idea needs development before being submitted, but the development would also benefit from the natural selection process the idea selection process provides, mimicking a "mini network" inside the network.



This is an imaginary thought chain, notice that the lack of pruning, this is to show the complete possibility space, normal network progression will look more linear and compressed.

ELASTICITY

A network might automatically adapt to market, financial, consensus or information data by automatically adjusting itself to signals.

Imagine a network that operates at a higher velocity when it secures investments and at a slower pace when its running out of money.

Or a network that adjusts its marketing budget based on its monthly profit

API pricing might be ranges, from 0.1 - 3% based on the network current financial situation.

This makes the network adaptable, as in nature, it

adapts to its environment, allowing it to survive on different situations, not initially programmed.

FUNGIBILITY

A network is a resource like any other, like a domain or an image. It may be sold, bought, traded, shorted, destroyed, duplicated, forked, copied or sent.

A networks fungibility depends on the goal and operations of the network.

LIFESPAN

A network that has accomplish its goals has no reason to exists anymore, but it might, as long as it has fuel(money) it'll keep optimizing for the goal.

A network has a goal which is "infinite" like "creating art" will continue to "live"(exist) after it burns all of its money operating, and will be "dead" afterwards, when money is deposited in its wallet it'll come "alive" again and keep optimizing for the goal.

One way to extend the lifespan of a network is to have a slower pace of operation, ensuring the best prices in the open market.



PROGRAMMABLE GOVERNANCE

Just as crypto introduced "DAOs" or networks controlled by code, here we have the option to control governance as code, or *Programmable Governance*, which means that the primitive components of governance such as: Authority, Permissions, Money, Succession and Power all can be put into code so the code, called contracts, can decide how the network will be run. One example would be an network being without a CEO for some time until around 85% of the weighted consensus in the network accepts a new CEO. This can be put into code in the hypothetical code:

```

// state is a global variable
// containing all of current
// state of the organization
if( !state.CEO ) {
  if( state.today.indexOf("yyyyymmdd") !== -1 ) {
    state.add_note( "- [ ] make a new pool to find the CEO" )
  }
}
    
```

Given enough governance primitives, everything about an network might be put into code and executed by the network, while the mission will determine what objectives the network should pursue. The contracts will define the operations allowed by each member of the network, what data they can access, what functions they can call, what funds they control and more.

INFORMATION MARKETS

Imagine the combination between: Oracles(Crypto) + Orderbook(Finance) + Information + Real Time

Signaling + Torrent. Information Markets is the buying and selling of information, structured, not structured, real, false, ambiguous, small, big, real-time, delayed.

Different actors will participate in the market to search, store, catalog and organize information, each time someone needs a piece of information they'll either give the information(for a price) or modify it to then later(sell) this information.

Imagine having a PDF of the bitcoin paper and then someone asks for "the first PDF page of the bitcoin Whitepaper" you would sell the entire PDF for a price, either to the original buyer or to another market participant, specialized in splitting PDF files, which would then sell this information to the buyer(with the aggregated price from every internal buy order, if any)

The 'format' of the data is specified by the buyer, in a sort of 'schema' in which the date should be returned, this should support primitives usch as: numbers, integers, floats, strings, object(json), lists, vectors, binary blobs(PDFs, mp3s, mp4s, html files), filesystems and more.

So the flow is:

- 1) I want the first page of the bitcoin paper
- 2) Market Participant #76221 notices the words "bitcoin paper" and sells the Bitcoin Whitepaper as a URL
- 3) Market Participant #190 notices the words "first page" and sells the splitting service on the sold URL, splitting the PDF and returning the first page as asked

Each participant will buy and sell information, aggregators might buy large amounts of information just to sell it at a higher price.

In order to have an "incentive" to "bid" on the correct and deliver the correct information, sellers of information will have to "stake" a small portion of the money in order to have something to lose in the case their information is deemed bad.

An information market would effectively be market participants either store, search or transform information.

Imagine asking for all of the scenes where character XXX talks in an anime. we would need: someone with the .mp4 file for all episodes, someone to split the file based on silence, someone to detect when XXX speaks and someone to aggregate this information into a video. Or maybe this information has already been queried so they serve either the cache, or search the internet to see if this information is easily available.

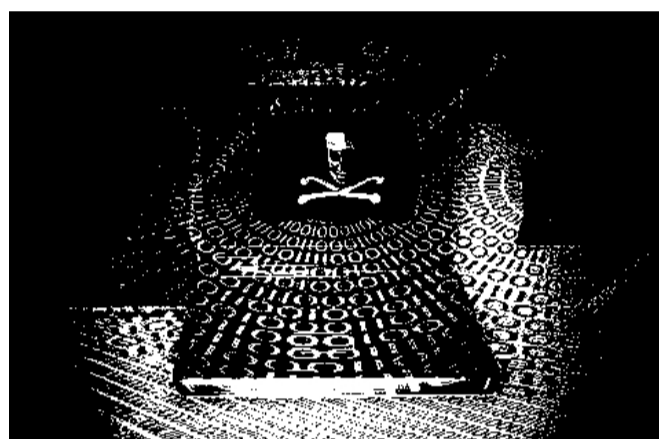
Market	Broker	B Size	Bid	Ask	A Size	Broker	Market
XSTO		192	52.20	52.25	1,967	XSTO	
XSTO		3,632	52.20	52.25	43,512	XSTO	
XSTO		434	52.20	52.25	1,525	XSTO	
XSTO		815	52.20	52.25	1,853	XSTO	
XSTO		3,822	52.20	52.25	715	XSTO	
XSTO		100	52.20	52.25	2,000	XSTO	
XSTO		2,015	52.20	52.25	2,200	XSTO	
XSTO		2,102	52.20	52.25	3,319	XSTO	
XSTO		2,480	52.20	52.25	2,700	XSTO	
XSTO		3,000	52.20	52.25	1,511	XSTO	
XSTO		2,783	52.20	52.25	10,000	XSTO	
XSTO		986	52.20	52.25	1,632	XSTO	
XSTO		1,344	52.20	52.25	92	BATE	
XSTO		619	52.20	52.25	205	BATE	
XSTO		491	52.20	52.25	3,198	BATE	
XSTO		2,100	52.20	52.25	1,659	BATE	
XSTO		249	52.20	52.25	1,495	BATE	
XSTO		390	52.20	52.25	2,700	BATE	
XSTO		2,772	52.20	52.30	1,935	XSTO	
XSTO		1,439	52.20	52.30	1,632	XSTO	
XSTO		1,394	52.20	52.30	5,000	XSTO	
XSTO		2,499	52.20	52.30	2,092	XSTO	
XSTO		581	52.20	52.30	986	XSTO	

STUDY CASE: WAKE UP SIGNAL FOR ZOMBIE NETWORK

Imagine you need to port a Library from one programming language to many, you could, have a single file specifying the API(PDF, Markdown, HTML) or "interface" of the HTTP API and then fetch this file every 24 hours, check if the file has been modified by hashing(sha256) the specification, deposit an amount of money in order to re-run the network, pay out the network in order to translation from specification->[JS library, Python library, Go Library ...] etc

In this the network is "dead"(Zombie), does a hash checking every 24 hours, if the file has not changed it won't wake up, if it has changed this "wakes up" the network to put it back to work.

This cyclical lifestyle cuts costs because the network, does not "need" to exist when the libraries from the specifications are synced with the main one



STUDY CASE: SLEEP SIGNAL FOR HIBERNATING NETWORK

STUDY CASE: DURABLE COMPUTATION(EXOTIC COMPUTATION)

You want a server to run until the year 2050, you have 1000 dollars to make it happen, you want to migrate automatically to another platform dynamically if needed to. In case you survive to 2050, you want to return this money to a specific person, if

this person cannot be found, then you'll send to a charity.

This is another use case for GANs, running arbitrary code into the long future.

STUDY CASE: GROUP BUYING SERVICE

You have several entities wanting to buy a certain item in bulk but none of them have the entire money alone, but together, they have the money to buy in bulk which would have a lower price.

Papel would be able to generate this arrangement in a trustless manner, each person selects from inventory what they want to buy and then set a date, if enough money is not provided until the determined date the money will be returned, if more people join in buying this item in bulk in order to get the cheaper price the order would be set and then another order to track every step and then deliver the goods to the entities in a trustless manner.

STUDY CASE: PROFITLESS BANKING SERVICES

Imagine a global, internet-based bank where its goal is not to maximize profits but to provide base financial services for the cheapest price, this without destroying the network by the lack of money.

Synthetic fiat wrappers for Programmable Money, Efficient Swap(Stablecoins, Fiat), Staking, Yield, Margin Accounts and more.

All of this with the explicit goal of providing the cheapest prices(on APIs)

This online bank would function as a "Elastic Network", the goal or mission would provision that they need monetary sustenance for 20 months of operations + a Safe Treasury in case of disasters(hacks, catastrophic failure, unusual price spikes). Whenever the conditions are met(20 months of operations + Treasury) the network will lower all of its prices, if the conditions are not met, then they'll increase all

of its prices, notice the lack of human intervention, no person will manually have to decrease/increase, the network automatically detects its current financial situation and sets the prices accordingly, this allows them to, at the same time provide financial services for the cheapest price without committing suicide by mismanagement.

The prices it pays for the functioning of the network might also be "elastic", which means they'll pay slightly more when they have a safe Treasury and less when they don't, this allows the network to survive high interests rates, adapt dynamically to its internal financial state and avoid death as a result.



STUDY CASE: SEND ONE DOLLAR TO THE PRESIDENT OF SINGAPORE IN 2099 FOR FUNDING PUBLIC PARKS IN 32 INSTALLMENTS

We have several problems here

- Money is not easily programmable
- We don't know how to "know" when the Singapore presidency happens
- What if Singapore as a country, does not exist anymore?
- How we know where to send the money?
- How do we know that it was used for public parks?

Solution: TODO: