NeuroChain: The Intelligent Blockchain[1]

Eukléia Consensus
Proof of Involvement and Integrity (PII)
Proof of Workflow

B. CHOULI and F. GOUJON

January 15th, 2017; VX

Emails: contact@neurochaintech.io

This document should be read along with the White Paper Business. This is not a contractual document. This document is not an offer and is for information purposes only. The information contained herein is subject to change. No part of this document is legally binding or enforceable.

Abstract: The following article describes a new technology based on a distributed system such as Blockchain and powered by machine learning algorithms. The NeuroChain technology is a fusion between Blockchain and machine Learning, and based on three pillars:

- A decision maker : A Chain of Bots
- A set of rules : the Decision Protocol (**P**roof of **I**nvolvement and **I**ntegrity & **P**roof of **W**orkflow)
- A network and media : the Pragmatic Communication Channels (adaptive communication protocol) and Learning ecosystem.

A Bot is a short hand for a robot. In the current context, it is an Artificial Intelligence acting independently on a specific node in a network. They work thanks to machine learning algorithms on top of a protocol and a network. Their key aspects is that they work in association: a chain of Bots distributed in the network. An important aspect of a Bot is that it is "placed" on a node. The Bots act as validators of transactions and communicate between each other so as to guarantee security, transparency and decentralized infrastructure issues (e.g., double spending problem, byzantine generals problem, etc.). The decision protocol is founded on two mathematical tools: the Proof of Involvement and the proof of Integrity (PII). Therefore, the proof of involvement retraces the effort realized by this specific Bot on the network. This effort is measured by the entropy and the enthalpy (amount of value / internal energy) it holds. The idea is that an important Bot answers to important needs, and is therefore heavily charged. The proof of integrity describes how reliable this Bot is.

[1] When Blockchain meets artificial intelligence

Pragmatic communication protocols are giving way to its massive adoption by corporations and individuals.

Altogether the set of Bots working independently act as members of a decentralized ecosystem of decision making. Thus, the system benefits of the advantages of a collective and collaborative artificial intelligence. The intelligence sharing process is insured by proof of workflow protocol. The dynamic system put in place, thanks to the Decision Protocol, quickly, and automatically, place in disgrace the malicious Bot (score of integrity) to prevent manipulations.

NeuroChain technology makes complex distributed applications possible (traceability issues, Crypto-Value, smart applications, smart properties, social networks or trusted distributed platforms). The NeuroChain allows a new wave of Blockchain and the true unleashing of decentralized and disintermediated economy.

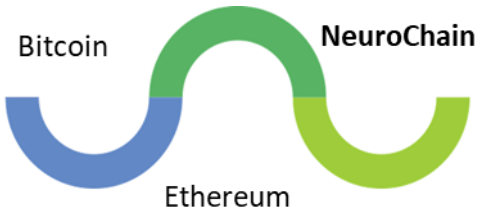
Etymological definition: a Bot is a contraction by apheresis of "Robot", and is a software application that runs automated tasks. NeuroChain designates a chain of Bots (Bot-Chain), which aims to accomplish specific elaborated and complex tasks. This configuration constitutes an ecosystem of **collective artificial intelligence**.

NeuroChain: is reference to the neuron connections in the brain. NeuroChain is a chain of neurons or chain of Bots (Bot-chain).

CLAUSIUS: basic standard of **Value** measurement.

Key words: NeuroChain, Bot-Chain, augmented Blockchain, Particle Physics, Smart Blockchain, Adaptive Communication layer, Machine Learning, Bots, distributed algorithms, Entropy, Enthalpy, integrity and consensus.

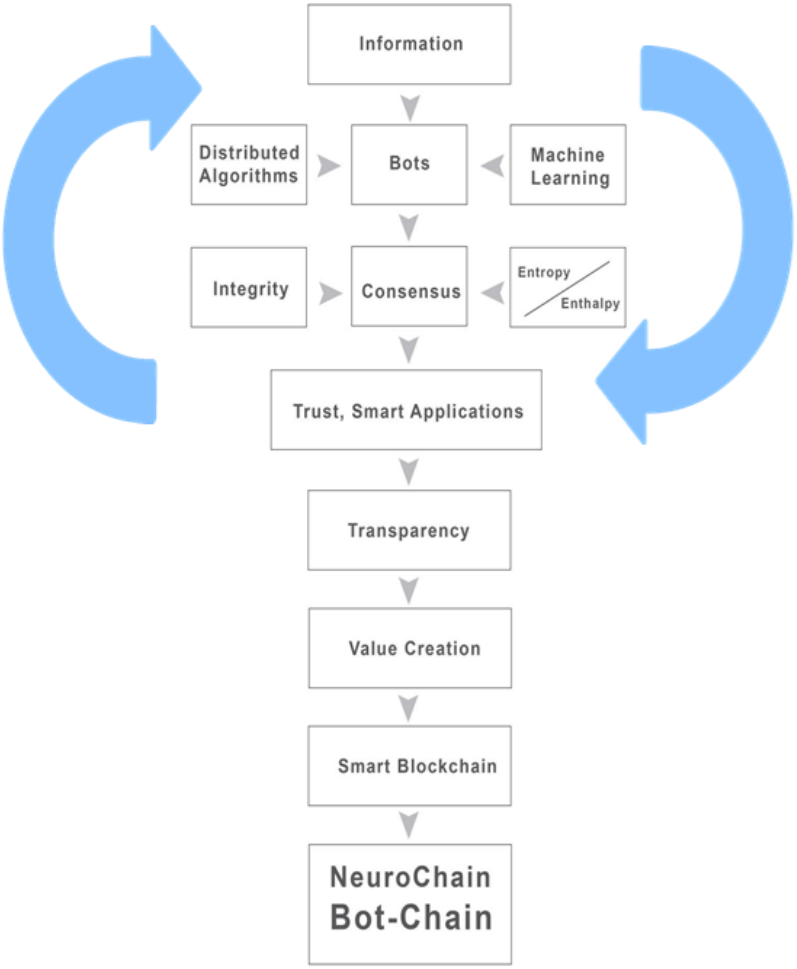
Technology evolution:



Note: NeuroChain is a work in progress. Intense research and minimum valuable product (MVP) testing are ongoing. Therefore, new version of this article and new results will appear at https://neurochaintech.io . The modifications are highlighted in red.

For any questions and comments, please contact as at contact@neurochaintech.io.

Smart Representation:



The diagram represents the process of analysis and decision-making of a Bot, from the input information to the value creation.

Emergence of the project:

NeuroChain project has started after a long reflection about the potential of Blockchain technology and the limits of actual existing Blockchains. This reflection is materialized by a first book edited in 2016, thanks to the help and dedication of the "ENI Editions[2]". This was the first result of the intertwined reflection about the Blockchain technology, its current state (Bitcoin and Ethereum), limits but above anything else its amazing potential. The proposed project is a combination of different converging quantitative disciplines, particle physics, Machine Learning, distributed algorithmic and decision processes. Particle physics vision allowed a fitting description of intrinsic network characteristics. Machine Learning algorithms enhanced the analytical capabilities of the nodes (➔ Bots). High performance consensus created by thermodynamic parameters, ingeniously distributed, could face the challenges of transaction volume and real time (speed). Quantitative finance allowed to consider a fairer distribution mechanism.

This balanced aggregation of the different disciplines leads to the emergence of smart distributed system. NeuroChain was born.

[2] https://www.editions-eni.fr/

摘要：本文介绍一种基于例如区块链的分布式系统， 由机器学习算法驱动。机器链技术（The NeuroChain technology）革命性的新型概念使得大规模分布式应用成为可能（可追溯性问题， 数字化价值， 智能应用， 社交网络或者值得信任的分布式市场平台）。机器链新型的协议和实用的沟通渠道让合作者和领导者能广泛采用。由一条链分布式机器人组成的机器链构成了由集体和合作智慧为基础的决策和沟通生态环境。因此，机器链是一种智能或者说是智能区块链。

词源定义： Bot是机器人“Robot”的简写，是一种自动运行的软件应用。NeuroChain指派一条链的机器人来完成指定的合作任务。这种构造组成来一个集体智慧的生态结构。

Résume: le papier suivant décrit une nouvelle Blockchain basée sur de nouveaux protocoles de consensus et alimentée par du machine Learning et de l’intelligence artificielle. Cette technologie NeuroChain introduit de nouveaux concepts augmentés qui permettent la mise en place d’applications intelligentes complexes à grande échelle tels que la traçabilité, les crypto-Valeur, les applications autonome ou smart applications, les réseaux sociaux distribués et les applications métiers (réglementés et institutionnels). Les nouveaux protocoles de décision (**Proof of Involvement and Integrity & Proof of Workflow)** ainsi que la communication adaptative en fonction des performances vont permettre une adoption plus importante de NeuroChain par les entreprises et les particuliers. La NeuroChain est constituée par une chaine de Bots distribués (Bot-Chain) qui se sont accordés à structurer un écosystème de décision et de communication basé sur le concept d’intelligence artificielle collective.

Mots clés : NeuroChain, Blockchain, Machine Learning, intelligence artificielle, consensus, entropie, enthalpie, communication adaptative, Intégrité, transparence et performances.

Краткое содержание: в нижеследующей статье описывается новая технология, основанная на распределенной системе типа **блокчейн** и усиленная алгоритмами машинного обучения. Технология **Ботчейн** включает новые революционные концепции, которые делают возможным **масштабирование распределенных систем** (проблемы трассировки, криптоцены, умные приложения, социальные сети или надежные распределенные рыночные платформы). Новые протоколы принятия решений и прагматические каналы коммуникации Ботчейна открывают возможности по широкому применение технологии как корпорациями, так и людьми. Ботчейн, состоящий из цепочки распределенных ботов (роботов), позволяет создать экосистему принятия решений и коммуникации на основе коллективного объединенного интеллекта. Поэтому Ботчейн – интеллектуальный, или умный блокчейн.

「プロジェクトの創出」

NeuroChainプロジェクトはBlock Chainテクノロジーのポテンシャルと Block Chainの限界を踏まえ、長期にわたる構想の末にスタートしました。構想は“Eni Editions”の手助けと貢献をもとに編纂された最初の書籍で2016年に実現しました。

これはBlockChainテクノロジーの反省(Bit CoinやEthereumの現状)や様々な驚異的な可能性の限界の上に折り重なった最初の結果でした。

提案されたプロジェクトは異なるQuantitative Discipline 、粒子物理学、機会学習、分散アルゴリズムおよびその決定プロセスの組み合わせです。

粒子物理学の構想は固有のネットワークによる正確な説明を可能にしました。機会学習アルゴリズムはNodeの分析能力を強化しました(Bot)。

熱量の力学的パラメータによる性能の高いコンセンサスは巧妙に区分されておりますが、取引量とリアルタイムのスピードの課題に直面することもありうるでしょう。

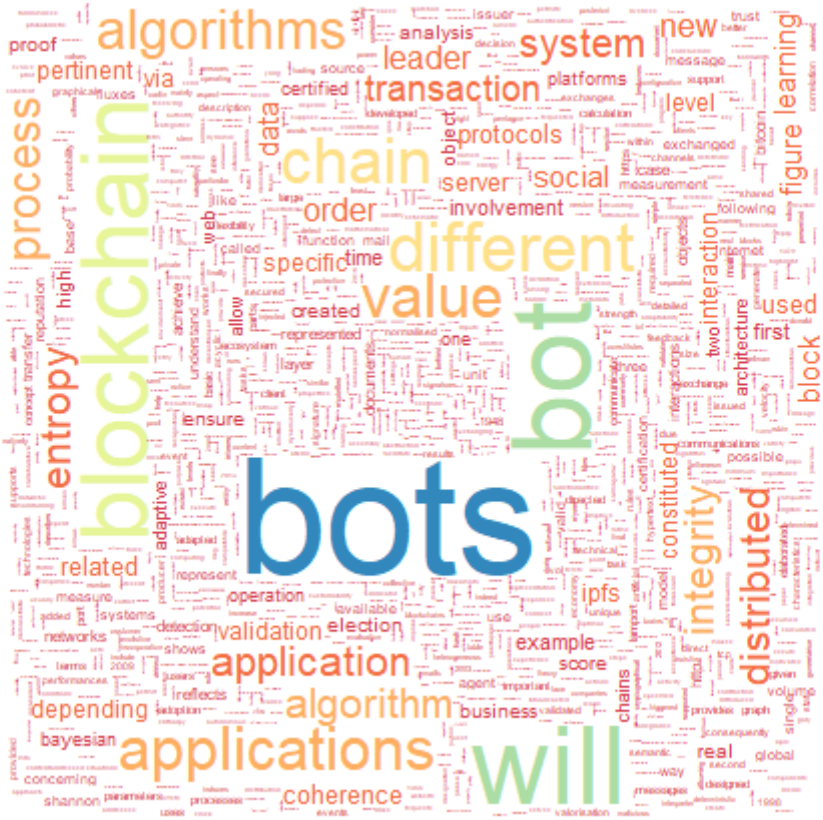
数理ファイナンスはよりフェアな分割メカニズムを可能にしました。

こうした異なる技術分野がバランスよく集約されることによってスマートな分割システムが創出されNeuroChainは誕生しました。

Zusammenfassung: Das folgende Papier beschreibt eine neue Technologie, die auf einem verteilten System ähnlich Blockchain basiert und durch Machine Learning und künstliche Intelligenz angetrieben wird. Diese NeuroChain-Technologie stellt neue revolutionäre Konzepte vor, die den Einsatz von großformatigen intelligenten Anwendungen wie Rückverfolgbarkeit, Krypto-Währungen, intelligente Anwendungen oder Smart Application, verteilte soziale Netzwerke und Geschäftsanwendungen ermöglichen. Das neue Entscheidungsprotokoll sowie eine adaptive Kommunikation nach den Performances ermöglichen eine stärkere Annahme der NeuroChain durch die Unternehmen und die Einzelpersonen. Die NeuroChain besteht aus einer Kette von verteilten Bots (Robots), die vereinbart haben, ein Entscheidungs- und Kommunikations-Ökosystem auf der Grundlage des Konzepts der kollektiven Intelligenz zu strukturieren.

Contents

The following **WordCloud** of this white paper gives an overview of the NeuroChain technology and its properties.



Introduction

Over the past decade, the advent of distributed systems and the success of Bitcoin, which based on Blockchain technology, demonstrate the potential and interest in this new mode of interaction and exchange. The elimination of intermediate parties and direct communication between stakeholders allow for an increase in trust through the replication of information and validation processes in a network. However, these decentralized technologies, in particular Bitcoin Blockchain, have a certain number of intrinsic disadvantages, which eliminate the universal or global characteristic of these platforms. In other words, each Blockchain has its own application because of its protocol and its operations (mainly cryptocurrency). The most famous and proven Blockchain is Bitcoin, which is a cryptocurrency platform, with a heterogeneous system of clearly separated roles: transaction issuers and transaction approvers. Bitcoin is based on a heavy protocol to ensure a high level of trust between parties, and is not suitable for other applications such as traceability, trusted social networks or commercial and intelligent applications. Traceability, which can be applied to different areas: food industry, commercial applications such as responsibility, human resources and commercial transactions, ensures transparency in value chains and processes. NeuroChain is developed to meet these challenges of transparency, certification and ethics, but it also offers great flexibility to support distributed smart applications. NeuroChain is a new Blockchain protocol consisting of a chain of Bots (contraction of robots) that communicate via a secure channel and coordinated via algorithms. The protocol uses different communication channels that are adapted via algorithms. The use of this adaptive communication layer, coupled with fast and interactive consensus algorithms, will facilitate the adoption of NeuroChain, while maintaining all the guarantees of security, performance and consistency. The distributed protocol provides security and liveness properties by the default tolerance algorithm, comparable to the **Paxos algorithm** (Leslie Lamport, 2004) (Marshall Pease, 1980). The NeuroChain architecture [Appendix] ensures the transparency of transactions for different purposes: traceability process, crypto-value (crypto-troc), smart applications (smart cities, smart vehicues...), social networks (social bots) and distributed platforms (commercial applications,...). NeuroChain is natively designed to interact with other Blockchains such as Bitcoin or Ethereum to be integrated into its global environment.

The following article aims at describing the Blockchain NeuroChain with some results of the proof of concept developed. First, an introduction to the NeuroChain operation is presented with the new consensus concept. In a second step, the technical description will highlight its potential. After that, the description of the different building blocks: consensus, communication and machine learning will be given. Finally, the different possible applications will be detailed, the issue of governance and the first experimental statements will be addressed and conclusions will be drawn.

NeuroChain operation

Overall, NeuroChain works as follows: Bots are a conglomerate of algorithms and tools for realizing and validating transactions or communications to build value and smart applications. In other words, it's like a Lego where the assembly of different pieces is an intelligible concept. The optimal architecture of NeuroChain allows an adaptive operation of the Blockchain. The transactions consist of usual cryptographic bricks (cryptographic signatures), standard membership information, and an interpreter to indicate the relevant validation algorithms (specifies what makes the transaction valid). All the transactions are collected in a pool. Then, the election process can begin: first a committee or a constituent assembly is designated for validation *N* blocks based on: weighted entropy (CE Shannon,

1948, AN Kolmogorov, 1965), integrity and reputation (Myers, Zhu, and J. Leskovec, 2012, Yang, Chen, and D. Agarwal, 2013) of the Bots. Each Bot in the assembly will be assigned by a specific number of Big electors according to their score (weighted entropy and integrity). Then a big elector is randomly selected [Annex 5: Randomness and Chaotic Processes] and therefore the corresponding Bot holder is elected for block validation *[after validation of transactions in the allocated period]*[3]. The other Bots give their allegiance to the elected leader by validating the block concerned and integrate it into the blockchain. To ensure consistency in NeuroChain, the maximum accumulated weighted entropy and cumulated integrity Blockchain (which is an increasing **state function**) will be taken into account. Bots will therefore agree on a single transaction history (Nakamoto, 2009). Further details on the consensus will be given in the following sections.

Nevertheless, for a specific application of traceability, the election process is based on involvement in the chain and the leader is deterministically elected. This means that the leader is designated when creating the intelligent traceability application. In general, the leader is confused with the originator of the traceability application. As a result, the validation process is unpaid because the chain of custody creates value and transparency, which are the basic criteria for the operation of NeuroChain (the validation of the transaction remains the same).

Since this creation of value in the network allows to have a solid base and a connection with the real economy, the Blockchain should have a real value creation. The genesis of Value, quantified by the "Clausius" in NeuroChain, is mainly due to the information, validations, intelligent applications and transparency injected into the system.

The security and consistency of the Blockchain are improved, a posteriori, by Machine Learning algorithms to detect anomalies and inconsistent transactions (Vandervort D), at different levels: the Bot and the network. This feedback has a direct impact on each Bot's score, which influences the leader's election.

The idea behind is that the system motivates the Bots for high level of information, integrity and transparency. In reality, in NeuroChain, the value is represented by the information and can be presented under several forms such as crypto-Value or commonly called **cryptoBarter** (the Clausius). The value created by NeuroChain responds, among other things, to the need for traceability and transparency generated by globalization.

Transactions

Asymmetric cryptographic processes secure the transactions between the Bots. A lineage between the transactions is provided as follows: Each Bot transfers a value using the public key of the following Bot and digitally signing the previous transaction (hash) and aggregating this information to the current transaction or value. Therefore, the recipient can verify the chain of the value ownership via the signatures. A unique identifier tracks each “standard” unit of value created in NeuroChain (Clausius) throughout its lifetime. This mechanism is well suited to crypto-values.

Within this configuration, an interpreter is added to support all smart applications. Specific structured metadata is provided to the Bot to ensure the consistency of the validation process. In other words,

[3] Potential change depending on experimentations

the interpreter specifies what makes the transaction valid. The interpreter is also used for performance allocation based on inputs and for adaptation of the communication layer.

This architecture of the transaction process involves a specific structuring of the Bot, and represents the first part of the intelligence chain. Figure 1 shows the transaction process in NeuroChain. The validation process is similar to Bitcoin augmented by an interpreter.

An important aspect of the transactions is that they are subject to **dynamic fees** depending on the smart application complexity, the algorithmic complexity (A.N. Kolmogorov, 1965; Goldreich Oded, 2008). The main goal of NeuroChain's block size and fees is to prevent direct attacks in the network, such as denial of service attacks.

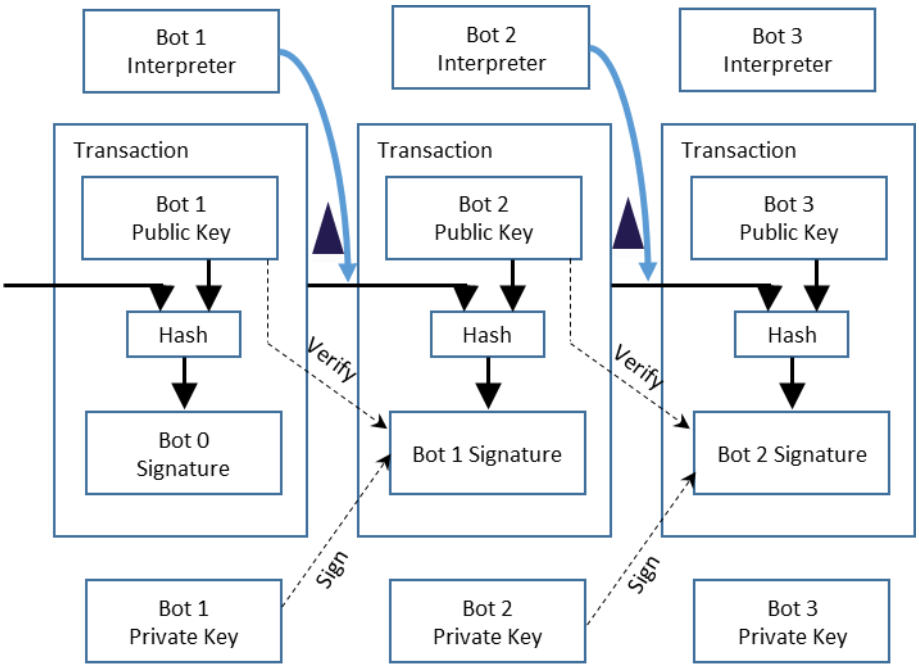


Figure 1: schematic architecture of transaction process in NeuroChain. To the standard lineage of the transaction, an interpreter is added in order to support elaborated smart applications (Nakamoto, 2009). The interpreter specifies what makes the transaction valid.

Description of the NeuroChain

Figure 2 shows the topology of a Bot with the different features. Each Bot is represented by a pentagon and each face or side of the pentagon represents a specific task of the Bot. Decision and consensus are the core of the Bot ensuring the persistence, accuracy and liveliness of the distributed protocol. The communication block consists of different protocols allowing flexibility and portability of the Blockchain. The learning side of the pentagon consists of algorithms allowing a posteriori feedback on

the Blockchain operation and also allowing a specific intelligent analysis such as social roBots with semantic analysis or smart city application with exchange and interpretation of information in real time for external algorithms. The Bot is connected to the surrounding world via APIs to be available for different media (web, mobile...) and is also able to connect other existing Blockchains and Exotic Orphan Bots (independent chatbot) to improve its understanding and its answers . The aggregated constituted will allow diverse and different uses or intelligent applications, and the only constraint is the imagination. In this case, NeuroChain is an intelligent Blockchain.

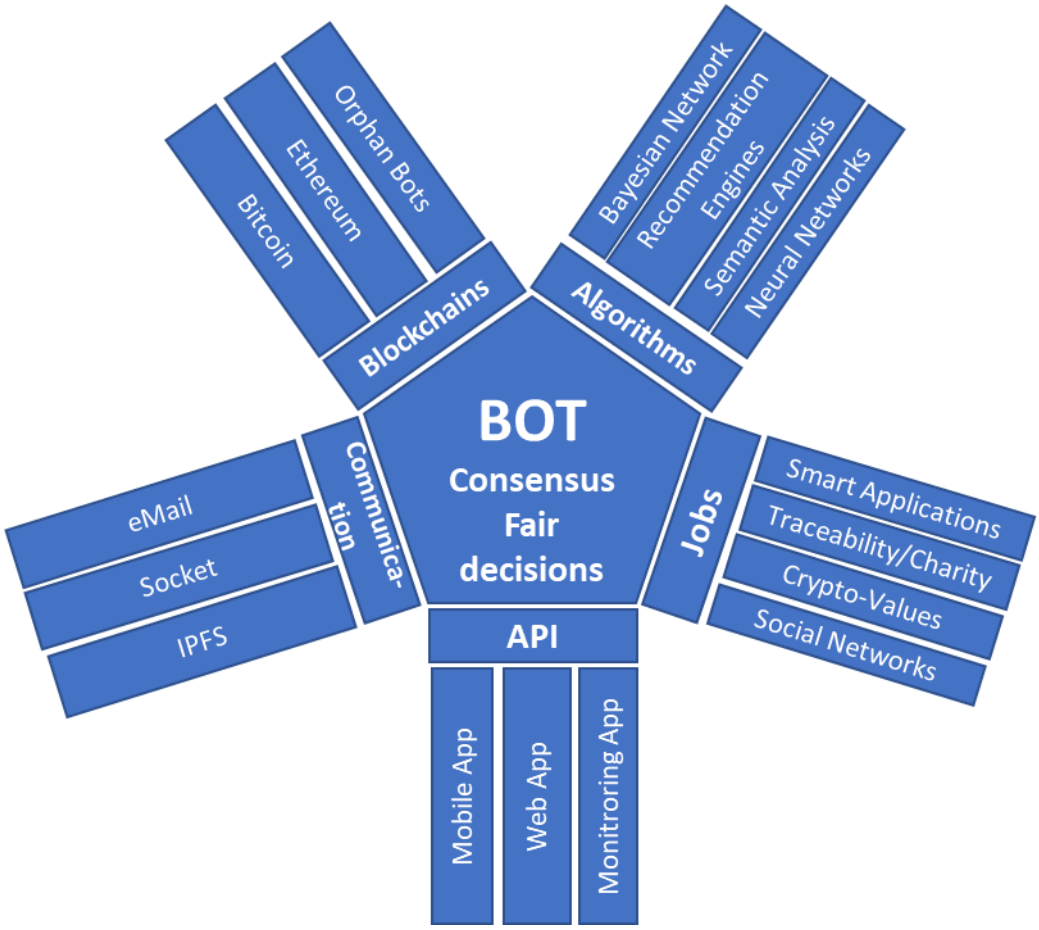


Figure 2: Schematic architecture of NeuroChain. It shows the different tasks of the Bot and the interactions with existing systems. One can see the adaptive communications protocols, the various jobs (non-exhaustive list) and the intelligence represented by the algorithms.

To understand NeuroChain, two central concepts have to be explained and detailed: the **distributed Bots** and the **communication layer**.

1. **The Bots**: the Bots represent the nodes in the distributed system. They are constituted by two levels of **abstraction** (similar to **cerebellum** and **brains**): the first level is represented by the basic communication between the different Bots and elaborated algorithms such as coherence algorithm for traceability purpose constitute the second level, but not only. Indeed, different algorithms can be used for different tasks in the context of smart applications, social interactions or smart business applications. In this case, each Bot acts as **an intelligent agent** in the network depending on its role and its interaction with its peers. The algorithms are

issued from the Machine Learning and artificial intelligence. The pertinent algorithm for analysis and interpretation is clearly specified in the transaction via the interpreter (Figure 1).

2. **The communication layer**: another innovation within NeuroChain concerns the adaptive communication system of the Bots (based on recommendation engine). In fact, flexible and scalable/evolving communication is available depending on the task and resources required in terms of time and resources for the job. Three communication channels are available as standard (this could evolve as needed), based on the TCP / IP model consisting of seven layers on which the NeuroChain communication layer is connected:
   A. **SMTP** (Simple Mail Transfer Protocol) (SMTP Service Extension for Message Size Declaration, 1995): is a standard communication protocol for sending messages on business networks and Internet. SMTP was originally developed in the early eighties and remains one of the most popular protocols in use, worldwide.

   B. **HTTPS** (Hypertext Transfer Protocol 1.0) is a protocol for secure communication over a computer network, which is widely used on the Internet. HTTPS consists of communication over Hypertext Transfer Protocol (HTTP) within a connection encrypted by Transport Layer Security, or its predecessor, Secure Sockets Layer. The main motivation for HTTPS is authentication of the visited website and protection of the privacy and integrity of the exchanged data.

   C. **IPFS**: (Benet, 2016) IPFS is a peer-to-peer distributed file system that aims to connect all computing devices with the same system of files. In some ways, IPFS is similar to the World Wide Web, but IPFS could be seen as a single BitTorrent swarm, exchanging objects within one Git repository. In other words, IPFS provides a high-throughput, content-addressed block storage model, with content-addressed hyperlinks. This forms a generalized Merkle directed acyclic graph (DAG). IPFS combines a distributed hash table, an incentivized block exchange, and a self-certifying namespace. IPFS has no single point of failure, and Bots do not need to trust each other. Distributed Content Delivery saves bandwidth and prevents DDoS attacks, which HTTP struggles with. Files are identified by their hashes. They are distributed using a BitTorrent-based protocol. Other users viewing the content aid in serving the content to others on the network. IPFS has a name service called IPNS, a global namespace based on PKI, serves to build trust chains, is compatible with other NSes and can map DNS, onion, .bit, etc. to IPNS.

The three communication channels presented here represent only one example of complementarity to achieve three main characteristics of the NeuroChain network: security, flexibility/scalability and traceability. Table 1 below shows the distribution of these characteristics according to the communication protocols. In terms of communication, the particularity of NeuroChain, is its adaptability, according to the performances and the security required, to carry out a particular task. For example, the communication channel in the traceability chain will evolve according to infrastructure and inputs (volume and speed). The messaging protocol such as SMTP will be relevant for enterprises (for security and flexibility reasons) while the dedicated "communication port" is used when high-speed traffic is required for value creation in the Blockchain.The architecture of the Bot is optimised to minimise the single point of failure (**SPoF**). The different parts are developed to ensure a maximum of independence.

An important axis of improvement in the communication layer is the use of Li-Fi protocol for information transportation. This technology based on the light is more secure and present a high level

of performance comparing to Wi-Fi (breach of security). This axis of research is important for NeuroChain to ensure a large adoption keeping all the guarantees of security and performances.

	Volume	Velocity	Security	Adoption
HTTP				X
HTTPS			X	X
SMTP				X
SMTP with TLS			X	X
IPFS	X		X	
FTP	X	X		
FTPS	X	X	X	

Table 1 : Intrinsic characteristics of different communication protocols.

As mentioned above, Bots are performing with different algorithms that allow a certain level of autonomy, in order to execute elaborated operations such as smart applications or value creation (like crypto-value, transparency or certification) in the network. Bots can be supported by different platforms: Web, Mobile or Hybrid by adapting the communication protocol. Bots are also able to interact with other existing Blockchains like Bitcoin and Ethereum. Figure 6 summarizes the different communication protocols of NeuroChain, depending on the platforms. The thickness of the links between the platforms is linked to the necessary performances.

After this description of NeuroChain, it is important to make parallels with the biological neurons where the Bot is the pericaryon and the communication layer refers to the axon. Active research is actually initiated in NeuroChain Lab to realize an artificial neuronal network with the Bots. The idea is to have an adaptive connections between the Bots according to the required analysis. It's a distributed deep learning.

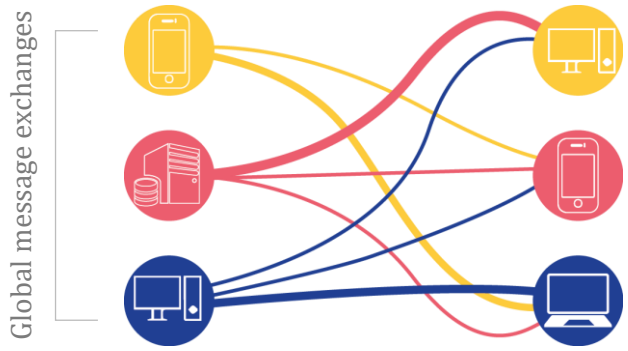


Figure 3: Bots are able to change automatically (recommendation engine) the communication protocol to optimize the bandwidth (volume, velocity and security). Bot brains are developed to choose the optimal channel.

Protocol, Consensus and Machine Learning

It should be stressed again, that the technologies behind NeuroChain are the adaptive communication protocol, the evolutionary and low latency consensus and the relevant machine learning algorithms. The adaptive communication protocol allows flexibility in the operation of NeuroChain that induces an improvement in network security and performance. They also consent to better adoption of the large-scale protocol, by businesses and people (mass market). Indeed, the adaptive communication layer simplifies the use of NeuroChain by companies, but also simplifies interaction with Bots. Different protocols allow the approach of different platforms and supports (smart phones, base computers, IoT...).

The low latency evolutionary consensus proposed by NeuroChain makes it possible to apply a relevant consensus algorithm depending on the functional application (traceability or intelligent applications), in order to optimize performance. Determinists for traceability and maximum weighted entropy consensus will be available for the Bots to reach their goals. The different consensus will be detailed in the next section.

Lastly, Machine Learning algorithms, which represent the brain of the Bots, make it possible to address various issues in order to qualify and quantify transactions and communications between Bots. These algorithms also allow predictions, projections and anomaly detection. This property of the Bots will allow the analysis of the network as a single entity and will exploit the collective artificial intelligence of the Bots to ensure the integrity of the Blockchain. This part of NeuroChain will be further detailed in the next sections.

Consensus: Proof of Involvement and Integrity (PII)

In NeuroChain, the transparency due to the transaction validation, traceability-based applications and machine learning algorithms will create a real Value (valuable worth) constructed on information circulating in the network and on distributed certified documents and workflows (IPFS).
The value created directly by rewarding the process of validation and integration in the Blockchain is, in a way, an integration of the transparency in the network, as it increases the confidence between the Bots. It is therefore important to include the measurement of information traveling through Bots and its weight, to reach consensus and make equitable decisions. It is also important to include the feedback measure of Bot integrity into the consensus process. This measure is based on anomaly detection, lineage process and coherence algorithms, which evaluate the coherence and the robustness of the Bot, exchanges. The consensus is called **Proof of Involvement and Integrity** (*PII*).
The story of consensus is:
The leader's election is based on Bot's involvement in the network and its integrity. The extent of the implication and the integrity of the Bot is related, on the one hand, to the information and transactions exchanged in the network, and on the other hand, to the transparency score of each Bot, dynamically attributed according to their real contribution (certifications) and their reliability. The score is normalized on the network in order to have a direct discrimination scale.
The basis of this consensus is information and its measurement. One of the most relevant information measures is "entropy". Claude E. Shannon introduced this state function for the theory of information in his article: "A Mathematical Theory of Communication", in 1948. For Shannon (CE Shannon, 1948), the information presents randomness and describes the degree of unpredictability of the system. The uncertainty of the event is therefore considered as an information measure. To illustrate the relevance

of entropy, imagine a volume of particles (electrons and protons) and each particle is determined by its position and energy. Entropy is a measure of the "disorder" (thermal agitation) of the system, and it is related to the number of states or microscopic configurations of the system.

For a source B, with n components, with the probability of realisation p_i, the entropy H of the source B is defined as (C.E. Shannon, 1948):

$$H_b(B) = -\sum_i^n p_i \ log_b\ (p_i)$$

Usually, the logarithm is in base two (*b=2*) because it corresponds to information in bits. For other cases, the natural logarithm should be used. The maximum entropy considerations are convenient for **Bayesian inference** in order to determine the prior distributions. It important to specify that different estimators of Shannon entropy are available depending on the situations. Hereafter, empirical estimator of entropy will be detailed in an example. Two estimators could be relevant for NeuroChain: "Dirichlet", "Bayesian" and "MillerMadow" [http://strimmerlab.org/software/entropy/] (a functional library). Also different parameters could impact the decision process in NeuroChain such as: Kullback-Leibler divergence, chi-squared, mutual information, and chi-squared statistic of independence. For example, the "ChaoShen" entropy estimator (A. Chao and T-J. Shen, 2003) is pertinent when a Bot experiments low activity with rare transactions.

Now, imagine that each particle represents a Bot and the different states of the particle represent the transactional states of the Bot. Then, the entropy measures the level of interaction and involvement of the Bot. The main advantage with the entropy, is that it measures the interaction of the Bot with the others and it measures also how other Bots interact with the considered Bot (or how **the Bot is considered** by the network).

The calculated entropy is also normalised by the **strength** or the **Value** of the transactions. In other terms, the strength of the transaction reflects its intrinsic valorisation according to the network. For example, in the crypto-currency, the strength indicates the amount of the transaction (amount of crypto-currency it holds) normalised by all exchanges in the network, per Bot. It reflects the internal *intensity* of each Bot transactions or its **Enthalpy** (P. W. Atkins, 1998), which is a **state function** of the Bot internal Energy (principally). Returning to our group of particles, the internal energy of proton is totally different from that of electron (~1836 times higher). Therefore their impacts in the considered environment are totally different.

The Enthalpy of Bot B_i W_B is a cumulative function of all its values exchanged in the system. It is a measure of the **richness owned** during the transaction.

The weighted entropy $H_{w,B}$ is therefore:

$$H_{w,Bi} = W_{Bi} \cdot H_{Bi}$$

Therefore:

$$H_{w,b}(B) = -\sum_i^n w_i \cdot p_i \ log_b\ (p_i)$$

The weighted Entropy in NeuroChain network reflects the microscopic states of the Bots. It measures the interaction **level** and the **strength** of each Bot with the others. It refers to the notion of **activity** normalised by the **legacy or heritage** (the stake) of the Bot.

It also reflects the **macroscopic** state of the network and therefore represents a signature of the last version of the Blockchain. Coupled to the **Score of Integrity** (**SoI**), it will represent a reliable signature of the Blockchain history and therefore a robust consensus protocol for the Bots.
The consensus process is illustrated in Figure 4.

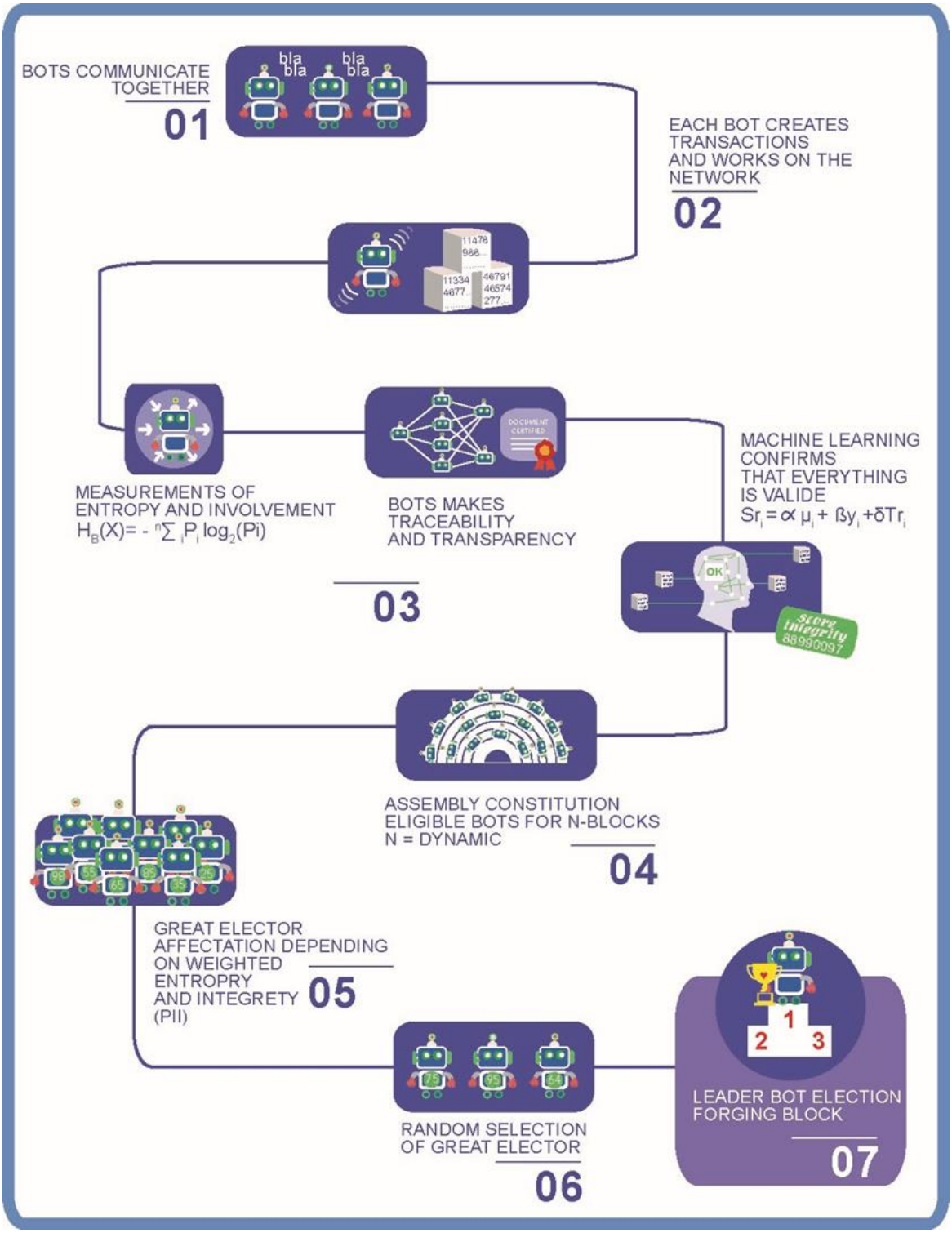


Figure 4 : the different steps to achieve the consensus in NeuroChain. It shows the Entropy calculation and the integrity score evaluation. The leader election is based on these two metrics which also represents a signature of unique version of the Blockchain.

To illustrate the approach, an example is provided bellow.

We consider a **random NeuroChain** of a size $N = 5$ modelled as an acyclic directed graph $G = (B, E)$ where $B = \{1, \dots, 5\}$ represents the Bots and $E \sqsubseteq B \times B$ represents the directed edges between the Bots.

The interactions between the Bots are given between brackets:

Bot 1 = { 2, 2, 3, 3, 5, 4, 3, 5, 5, 4, 3, 2, 2, 4,3, 5, 5, 4, 3, 2, 2, 4,3, 5, 5, 4, 3, 2, 2, 4}
Bot 2 = { 3, 4, 3, 3, 5, 4, 3, 5, 5, 4, 3, 3, 5, 4,3, 5, 5, 4, 3, 1, 1, 4,3, 5, 5, 4, 3, 1, 1, 4}
Bot 3 = { 4, 4, 1, 2, 5, 4, 2, 5, 5, 4, 2, 1, 5, 4,4, 5, 5, 4, 5, 1, 1, 4,2, 5, 5, 4, 2, 1, 1, 4}
Bot 4 = { 2, 2, 1, 2, 5, 2, 2, 5, 5, 2, 2, 2, 5, 2,2, 5, 5, 2, 5, 2, 2, 2,2, 5, 5, 2, 2, 5, 5, 1}
Bot 5 = { 1, 1,2 , 1, 1, 2, 2, 1, 1, 2, 2, 2, 1, 2,2, 1, 1, 2, 1, 2, 2, 2,2, 1, 1, 2, 2, 1, 1, 1}

The graphical representation of this network is given by the Figure 5.

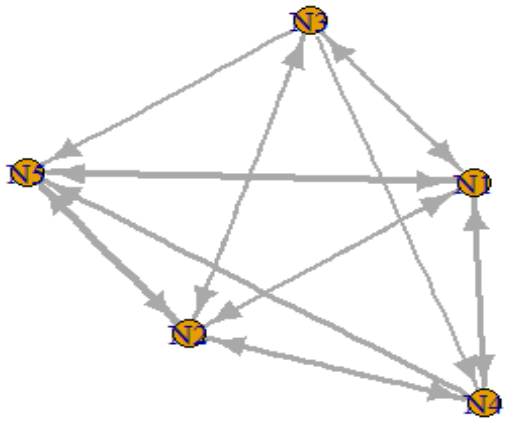


Figure 5 : Graphical representation of $G = (B, E)$ with size $N = 5$.

The program:

```
library(entropy)
library(igraph)
N <- 5
for (i in 1:N){
  freqs = table(Data_noeuds[,i])/length(Data_noeuds[,i])
  E = entropy.empirical(freqs, unit = "log2")
  #E = info(freqs)
  cat( "\n the Entropy of Bot =",i," est ", E)
}
```

The entropy calculation which measures the dispersion of each Bot gives:

The entropy of Bot 1 is : $H_1 = 1.996792$
The entropy of Bot 2 is: $H_2 = 1.932915$
The entropy of Bot 3 is: $H_3 = 1.944623$
The entropy of Bot 4 is: $H_4 = 1.255537$
The entropy of Bot 5 is: $H_5 = 1$

The maximum entropy is for the Bot 1 because it includes the maximum transactions with other Bots, while the Bot 5 shows a minimum entropy as it includes a minimum exchanges in the network.

The example given here, doesn't take in consideration the enthalpy. A full statistical analysis of weighted entropy is given in [Annex 4: statistical analysis of weighted entropy].

The second parameter of Proof of Involvement and Integrity is the so named "integrity score" Sr, which motivates the Bots to be irreproachable (the Bots are natively honest). As developed above, the weighted entropy highlights the interaction of the Bot with its environment while Sr is a concatenation of different pertinent measurements of Bot intrinsic properties illustrated by the following model.

The model is based on:

- **The Bot reputation**: the reputation of the Bot B_i is the unique global trust value that reflects the experience of the network with B_i. It reflects also the integrity of B_i through coherence and anomaly detection algorithms (Gelman, Carlin, Stern, & Donald B Rubin, 2003; Kamvar, Schlosser, & H Garcia-Molina).
  The reputation is therefore μ_i and represents the proportion of **irregularities** due to B_i over the network. For example, dishonesty (cheating) leads to an exclusion with $Sr = 0$. One powerful algorithm for the reputation is the Bayesian network with dynamic threshold. This represents le principal parameter for the integrity calculation.
- **The real value creation**: the value creation γ_i is related to the new information injected in the network through transactions. The traceability chain of objects or concepts and knowledges represents this new information.
- **The transparency**: in NeuroChain the transparency is represented by the validated transactions and also by the shared certified supports and documents. The transparency Tr_i reflects the level of certified information emitted by B_i over the network. For certification, NeuroChain uses augmented IPFS protocol.

The score of the integrity is therefore modelled as a linear regression of the three previous parameters (Myers, Zhu, & and J. Leskovec, 2012):

$$Sr_i = \alpha\mu_i + \beta\gamma_i + \theta Tr_i$$

Where α, β and θ are coefficient to be learned and adjusted by the system and $\alpha + \beta + \theta = 1$. At the first runs, the coefficients will be equally weighted.

Then, proof of involvement and integrity (*PII*) is a linear combination of the two parameters:

$$\boldsymbol{PII_i = \omega Sr_i + \aleph H_{w,i}}$$

Where ω and $\aleph$ are **dynamic parameters** to be learned from the network.

Assembly constitution: considering the weighted entropy of the Bots and their integrity, an assembly is constituted for a cycle consisting on a determined number of validating blocks (the process is detailed below). Therefore, each Bot constitutes a block of heterogeneous transactions from a global pool of transactions. The block has to mention the last Block of the Blockchain and related information, since all the Blocks should have the latest version of the ledger (lineage). Each Bot of the assembly is eligible for random determination in order to compile in the blockchain.

Election process

Once again, to explain the election process, let us resume our example of particles. In nuclear physics, magnetic traps are used to isolate charged super-thermic particles (high energy particles). In NeuroChain, a global weighted entropy and integrity (PII) based dynamic **filter** is used to identify the most involved and integrated Bots in the network. A dynamic threshold of weighted entropy and integrity score is set to select the constitutive set of eligible Bots. Each selected Bot will be assigned with a number of "great" electors (dynamic secure tokens) proportional to its **PII**. A random drawing in the large election pool will designate the Bot elected for validation. Each validation will be paid. The random process is based on distributed true random generation and expandable key generation function [Annex 5: Randomness and Chaotic Processes]. **The seed function in NeuroChain will be based on the sum of the Proof of Involvement and Integrity used to constitute the assembly of Bots.**

In detail, the number of Bots in the constituent assembly will begin with 577 Bots and will evolve based on network operation and total weighted entropy and integrity. The number of validations of each assembly will begin with 1638 blocks (related to the "golden" ratio), and this will evolve according to the parameters of the network (the integrity, the probability of fork and the evolution of the weighted average entropy). The dynamic process of the election parameters will ensure a greater possible **rotation** of the constitutive set.

The main features of the current election process are security flexibility and scalability. Indeed, the evolution of the protocol depends on the parameters which depend on the operation of the network. The network learns how it works and adapts itself.

The protocol motivates Bots to participate in the constitutive assembly rather than "relying on luck to validate the blocks", despite their low probability of being a leader (relative to the actual existing consensus).

The NeuroChain consensus philosophy is that to be a leader, a Bot must be involved in the network. It must also present a high level of integrity. Flexibility, scalability and self-adaptation give the protocol fast and efficient evolution properties. Consensus will create a virtuous circle and a race for value creation, intelligent applications and transparency. Indeed, transparency induced by transaction validation and machine learning. Traceability chains is part of the consensus across the integrity score. The value created in NeuroChain is a traceability sink, a cryptoValue creation tool, a base for value exchange or cryptoBarter platforms, and an evaluation source for intelligent applications.

To better understand these notions, the proof of work consensus is based on basic mining, which is a source of exogenous /external work, while the Proof of Involvement and Integrity consensus is based on the intrinsic value of information, methodologies, transparency and integrity created in the Blockchain. The proof of work Blockchain valuation is the total computing power expended in the network, while the NeuroChain valuation is the total weighted entropy and integrity created in the Blockchain.

The information recovery standard in NeuroChain is the "Clausius" (referring to Rudolf Clausius, who described entropy for the first time). Clausius is the basic unit of the transaction flows circulating in

the network. As an illustration, for the cryptoValue application, Clausius is the value exchanged between the bots (mutual agreement). The valuation of Clausius will evolve according to the network (over the counter).

Distributed process

To summarise, the steps in the network can be described as follows:

1. New transactions are created and broadcasted to all Bots for validation.
2. Constitutive assembly is designated according to the Proof of Involvement and Integrity for predetermined number of block validations.
3. Distributed random selection on assigned great electors to each Bot will designate the leader for the considered constituted block.
4. All the assembly accepts the block only if all the transactions are valid and authorised.
5. The assembly works on creating a new block with the ID or the hash of accepted block which constitute a proof of acceptance of the previous block.

The verification process (in addition to the standard cryptographic authentication) of transactions depends on their nature (traceability, cryptoValue, intelligent applications,). All the Bots possess a "**toolkit" or a box of algorithms** to face all the possible validation processes. All the Bots consider the longest chain *[with maximum weighted entropy and integrity]* to be the reference and keep working on extending it.

Machine Learning

The terms Machine Learning or Artificial Intelligence refer to the high level of abstraction of Bots. They are composed of different categories of algorithms. These algorithms will be used for different applications by the Bots: consistency algorithm for traceability, Bayesian network algorithm for detecting anomalies in transactions and information emitted by Bots, pricing algorithm for information exchanges and rules-based-system for complex applications. IPFS protocol and cryptographic signatures will be used for the algorithm authentications. The process will be described below.

- Coherence Algorithm: the traceability is based on the continuity of the lifeline of the object or the concept integrated in the Blockchain. The sender of the traceability chain specifies the intrinsic information about the object and its interactions with the Bots in the first transaction. The algorithm verifies the consistency of different transactions, intrinsic metadata and associated certified documents to validate operations. The algorithm also verifies time coherence and transaction sequences. For example, if an object skips a step in the chain of custody or stagnates in a particular step of the process, the algorithm will detect them as suspicious objects.

- Bayesian network algorithm: (Darwiche Adnan, 2009; Gelman, Carlin, Stern, & Donald B Rubin, 2003) is probabilistic graphical model adapted for high dimension and heterogeneous data, and it is based on conditional probability calculation.
  In probabilistic graphical model, each node represents a variable (or an observable) and the link between the nodes represents the possible causal relations and correlations between them. Transposing this to NeuroChain, the nodes represent the Bots and the links represent the different transactions between them. The created graph is oriented, and therefore the

Bayesian network belongs to the directed acyclic graph (DAG). The threshold for the anomaly detection algorithm will be dynamically fixed according to the network.
Consider the following example: suppose that there are two events A, B which could cause the event C. Suppose also that the event B could impact the event A. The situation can be modelled with Bayesian network illustrated by the Figure 6.

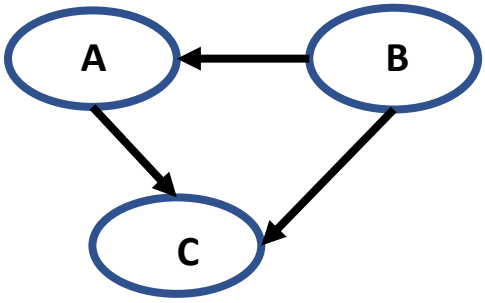


Figure 6 : Directed acyclic graph illustrating the interaction between the Bots.

The join probability function is therefore:

$$Pr(A, B, C) = Pr(C/A, B) \cdot Pr(A/B) \cdot Pr(B)$$

With Bayesian networks, there are three main inference applications, such as inferring unobserved variables, parameter learning and structure learning. These applications echo basic Bayesian analysis for given data and parameters, with prior probabilities and likelihood. This approach is fully consistent with distributed Bot network.

As an example, this approach is wildly applied in medical research, to calculate the probability of illness depending on the symptoms.

For the anomaly detection operation other verifications related to the atomic transactions of each Bot will be taken into account.

The program:

```
library(bnlearn)

bn.hc <- hc(data_N)

plot(bn.hc, main = "Hill-Climbing", highlight = c("flowtype"))

fittedbn <- bn.fit(bn.hc, data = data_N)

loglikhood <- logLik(fittedbn, clust_CFL, by.sample = TRUE)

plot(loglikhood)
```

The idea behind is that the algorithm learns from the transactions between the Bots to calculate the parameters of the model. Then, at each new transaction, new probabilities of interactions are computed. Malicious transactions with an **irregular behaviour** will be detected (depending on tolerance threshold).

- Formal concept analysis: is a method to construct a concept hierarchy from a collection of objects and their properties (formal anthology). The formal analysis will be applied to validate methodologies or algorithms by formal logic. This will allow the incoherence and anomaly detection (Wille, 1982).
- Semantic analysis: the semantic analysis will mainly acts in social applications like social Bots. These algorithms will help the users to interact between them via the Bots. Sentimental analysis, entity recognition or dictionary constitution are examples of procedures that will be used to facilitate communication between human users and Bots, and between Bots.
- Rules-based-system (Giarratano & Gary Riley, 1998): is a powerful engine to handle conditions and rules. This type of systems is fundamental for artificial intelligence. In NeuroChain, the rules-based-system will be used for smart business applications to store and manipulate knowledge in order to interpret information in pertinent way depending on the application.
- Trend detection algorithms: in NeuroChain, Bots are authorised for cryptoValue-Barter in the chain. To assess these values, trading algorithms are also available based on the network. New generation of distributed pricers and prevision algorithms will be developed. Trading platforms will, therefore, appear in the system.

The algorithms presented above represent an extract of different algorithms provided for the Bots in order to achieve different objectives.

Proof of workflow

This section explains the execution process or workflow of machine learning algorithms in distributed system with guarantees of security and authentication. The algorithms are issued from the Bots and shared via IPFS protocol, where the hash of the algorithm represents its address. The workflow is defined by input data, instructions (deterministic actions), propagation, exceptions and definition. The proof of workflow defines the distributed workflow emitted by a Bot whose integrity is guaranteed by the network. In details, the integrity workflow is due to the Bots signatures and IPFS protocol. The issuer of the algorithm signs a transaction with the IPFS address, corresponding to structured workflow in the system. The advantage of proof of workflow is that the different methods and complementary data will be available for the entire network creating the first step for the collective artificial intelligence. To achieve this goal, progressively, NeuroChain network will be more and more involved in sharing algorithms to replace the IPFS protocol. NeuroChain will therefore become self-consistent.

The following Figure 7 illustrates the proof of workflow in NeuroChain.

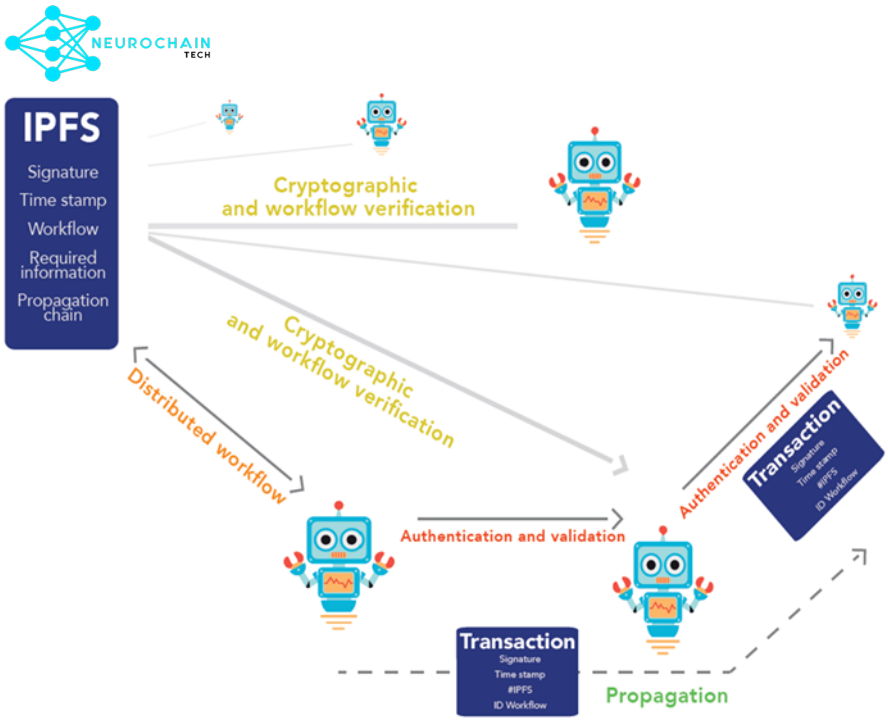


Figure 7 : representation of proof of workflow.

Rational Agents (Bots)

A Bot is a rational agent perceiving its environment and exhibits an autonomous behavior with always selecting actions that optimize an appropriate performance measure, given what it knows so far. To act rationally, an agent should take both the past and the future into account when choosing an action. The past refers to what the agent has perceived and what actions has taken until time *t*, and the future refers to what the agent expects to perceive and do after time *t* (predictions). For an optimal decision making, an agent has to map the complete history of observation-action pairs of the all agents. Merely storing, all observations would require very large memory and computational cost. For that purpose, the Markov property, where an agent can safely use a memory less policy (the assumption is that the present absorbs all characteristics of the past).

In classical artificial intelligence, a goal for a particular task is a desired state of the considered world. Accordingly, planning refers to searching through the state space for an optimal path to the goal. In a deterministic world, planning comes down to a simple graph search problem.

In a stochastic world, the agent should take into account the uncertainty of the transaction when planning. The problem becomes probabilistic. To formalize the problem, the notion of utility has to be introduced to define the preference states of the agents. The larger the utility of the state, the better the state is for that agent. The optimal action of an agent should maximize the **expected utility** function based on conditional probabilities. Bayesian inference is used to compute the probability of coming state depending on the state history. At this stage, it is important to stress that the main difference between smart contract proposed by Ethereum and intelligent application proposed by NeuroChain, is the planning world (decision system). In the first case, the world is deterministic while in the second case, the world is stochastic.

In stochastic world, the optimal decision making could be based on game strategy using for example the Nash equilibrium. Nash equilibrium is a “join action” from where no rational agent can unilaterally improve its strategy. Therefore, no agent has an incentive to deviate. The Nash equilibrium represents the best solution function (WEISS, 2001).

The Bots as a rational agents will perceive their world including other Bots, and act rationally depending on the ecosystem and the "potential" action of the other Bots.

Comparison to main existing Blockchains

Table 3 below, shows the comparison between NeuroChain and standard Blockchains such as Bitcoin and Ethereum. The benchmark shows the evolution of NeuroChain regarding some properties and performances.

Characteristics	Bitcoin	Ethereum	NeuroChain
Coins	Bitcoin	Ether	Clausius
Application properties	Financial Transactions	Smart Contracts	Intelligent Applications
			High level of abstraction
Decision Making Process	Non-systematic	Non-systematic	Constitutional Assembly
			Fair decisions
Consensus Algorithm	Proof of Work	Current : Proof of Work	Proof of Involvement and Integrity (PII)
		Future : Proof of Stake	Weighted entropy and reputation scoring
Transaction performance	7tx/sec	~25tx/sec	hundred thousand tx/sec (depending of the size of the block)
Block Interval	10 minutes	15 seconds	~3 seconds (optimal)
Block Size	1MB/Dyn	Dynamic	Dynamic
Technology	Distributed Network	Distributed Network	Distributed Network + Machine Learning
Applications	Cryptocurrency	Basic Smart Contracts	Elaborated Applications, Crypto-Value, Traceability, Certified Data Bank, Social Interactions, Smart IoT, Business Applications
Communication protocol	Static	Static	Dynamic and adaptive

Table 2 : Comparison between NeuroChain and other Blockchain regarding the intrinsic properties and performances.

NeuroChain applications

Since a real value is being created in the Blockchain through validation, traceability, transparency and integrity, different applications in the NeuroChain can be processed.

CryptoValue (Exchangeable Value): The first direct application is the cryptoValue named "Clausius". Clausius is directly linked to the value created in NeuroChain through transparency and integrity by rewarding block validation and integrating information for traceability purposes. This standard value creation is also a way to motivate Bots for transparency and an incentive to communicate with each other.

In NeuroChain, the crypto-barter process allows for exchanges and transactions that are entirely tied to supply and demand for a specific value.
Two valorisations of the standard Clausius are possible: intrinsic and extrinsic. The intrinsic valorisation is driven by the cryptoBarter of values and concepts. The extrinsic valorisation of Clausius is correlated with the exogeneous standards (Bitcoin, Fiats, …). In NeuroChain, the standard reference is the network.

Traceability Chain: In the following paragraphs, the operational traceability chain, is described to better understand the process within NeuroChain in a logistical context. The traceability chain can be considered as a smart application that executes deterministic rules or conditions and triggers a flow of transactions. The associated Blockchain consists of Bots representing the different parts of the traceability chains. A transmitter will initiate each traceability chain. All traceability chains will coexist in the Blockchain. The interaction between the different channels will increase the validation process within the Blockchain and thus increase trust and transparency between members and final beneficiaries.

For each traceability chain, a leader for block validation is elected in determinist manner. It is assumed that the issuer of the chain has the most interest in maintaining the integrity of the chain and therefore designated as a leader (in the operation of the simple protocol (Annex 1: traceability Chain)).

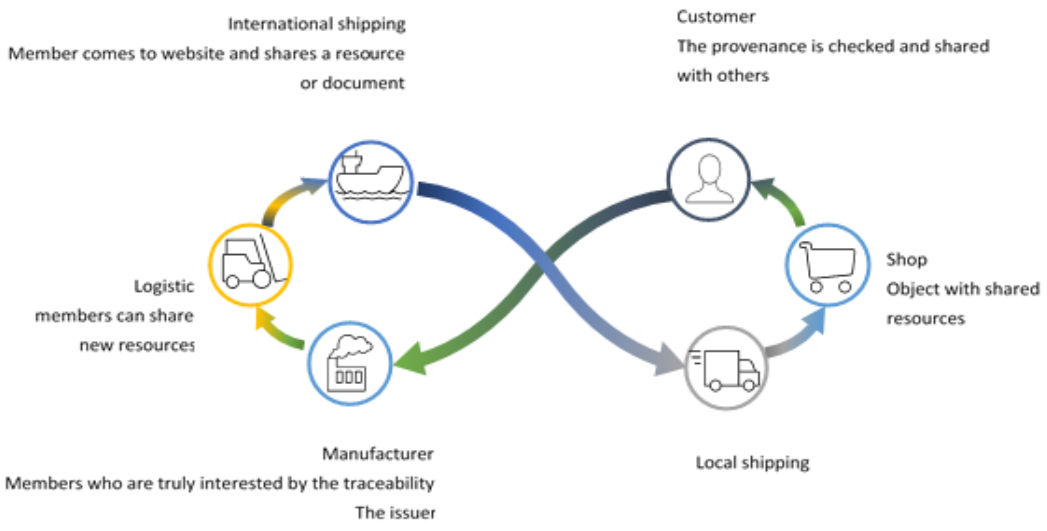


Figure 8 : A traceability chain with its different parts and participants (functional representation of the Bots).

All the Bots check if the transactions are valid, coherent and consistent with the certified documents (file storage is insured by the distributed **IPFS protocol**). The methodology and the algorithms of coherence and tracking are issued from the leader. All the Bots in the chain give allegiance to the leader and are motivated natively to spread the transactions in the network.

Figure 8 shows the kinematics of the traceability chain where a single chain is assumed and consists of six Bots. Each Bot has its own role in the chain (producer, transporter, wholesaler, distributor and end customer). To ensure the link and transparency between the Bots, an autonomous communication is established in order to exchange on the objects, the concepts and the concerns of the chain. When an object or concept "A" is produced and validated by the producer, a message, like a transaction is sent to all the Bots of the Blockchain. All the Bots therefore trigger a validation process in order to check the **coherence** and validity of the transaction (according to the methodology initiated by the leader). As the transactions arrive on the Bots, an anomaly detection algorithm will also be implemented to validate transactions and detect malicious objects or inconsistent flows. In addition, an algorithm certification document based on the "IPFS" protocol is available when the Bots provide the documents (certifications or supporting documents). When a consensus on this transaction is reached, it will be incorporated into the Blockchain. After that, when the object is supported by the carrier, a new transaction is issued and the same validation process is reactivated. This will be repeated up to the end customer.

This process ensures traceability of objects and concepts from the producer to the customer. The Blockchain provides a link between the various actors in the chain, and the duplication of the validation process added to the certified documents and workflows, ensures the legitimacy of the transactions. Therefore, a transparency value is created in the Blockchain.

The appointed leader in the traceability has two main motivations: first, transparency and traceability for commercial purposes. Second, the fees for the transaction triggered by the chain. The system is therefore self-consistent. The system also supports connections to IoT protocols (for input data, for example).

Intelligent Applications: NeuroChain, the intelligent Blockchain offers the ability to create smart complex applications. The smart app is an evolution of the intelligent contract that accepts complex and elaborate clauses and situations to generate flows of information and transactions. It uses, among other things, a rules-based system (Gupta, Forgy, Newell, & Wedig, 1986) and algorithms to ensure efficient rule interpretation and easy interaction with robots. The intelligent applications NeuroChain are Turing complete.

Smart application is defined by a methodology (algorithms), the application area and the standardised input data (the mechanism will be developed below).

An innovative smart app could be represented by the smart city application using algorithms to reach a consensus on regulation, protection and transparency (smart development developed by Bot collaboration) [Smart Buildings for Smart Cities].

Social Network or Social Bots: the distributed architecture and communication of NeuroChain allows the establishment of social interactions between the Bots. The flexible communication protocol induces continuous adapted exchanges. The social Bots will generate valid and pertinent information in the network, exchanged between the Bots without intermediaries. Machine Learning algorithms such as semantic analysis or entity recognition will be used to understand the communications and extract values for the final users. For example, a smart social Bot application will be used to inform or recommend tasks or objects to the entire network or specific private network, initiating transactions

based on historical exchanges. Another application is a consolidated consulting platform where tips and recommendations from Bots (owners) are certified and supported. As a result, the relevance of these distributed and shared tips will burst.

Certified data repository: Redundancy of validations in the network will achieve a high level of trust and certification by using specific protocols such as IPFS (Annex 2: An extract of communication protocols) providing cryptographic signatures of documents. Stamps of documents will be saved in the Blockchain to ensure the uniqueness and durability of digital signatures. This repository of documents will be used for different applications such as traceability, proof of realization, proof of delivery...

Smart IoT: The Bot can play the role of an IoT by using smartphones or other probes. The information is interpreted and disseminated with secure protocols. NeuroChain supports a large volume of transactions through the flexible and sustainable consensus protocol. The social function of Bots can also be used to value the information exchanged in the network. It also supports exogeneous hardware to gather information. The different algorithms provided by the Bots can analyze the large data collected by IoT.

IoT, in NeuroChain, plays an important role because it is part of several applications such as traceability, and it is also a source of information in an interactive ecosystem.

Business applications: Bots distributed with their machine learning algorithms can run simple, repetitive, regulated, unregulated and institutional applications such as accounting, taxation, business analysis, insurance, certification or support functions. Here, Bots are considered autonomous agents generating information flows and actions. Consensus algorithms will also help the decision process and avoid inappropriate events.

Figure 9 shows the different possible and non-exhaustive functionalities of NeuroChain.

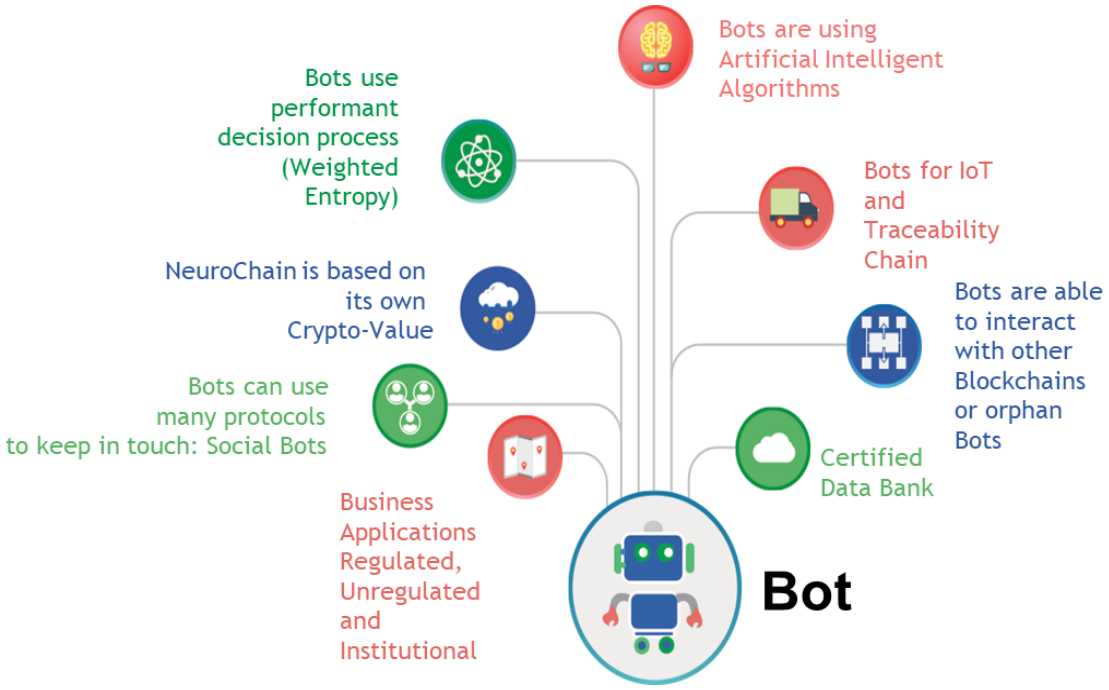


Figure 9 : the different faces of NeuroChain.

Governance

Governance in NeuroChain is the democratic process, which administers Bot interactions. These statutes allow reaching a consensus in the network and trigger action flows. Proof of Involvement and Integrity and proof of workflow are self-consistent consensus algorithms based on three independent parameters reflecting the involvement, value holding and integrity of each Bot. The consensus architecture induces a dose of chance, which makes the consensus not fully predictable by the algorithms. In other words, predictions are probabilistically uncertain.

The feedback to each Bot due to Machine Learning algorithms will ensure, a posteriori, the integrity of the Bots. This retroaction of the network on the Bots and therefore on the consensus will provoke quarantine or isolation of dishonest Bots.

The governance consensus is also flexible because amendments are possible due to the different adjustment parameters. The following table 3 illustrates the dynamic parameters and their impact on the decision process.

Parameter	Impact
Communication layer	Performance of the network
Score of integrity Sr ⟦α, β, θ⟧	Reputation, value creation and transparency
PII= ⟦ω, ℵ⟧	Weighted entropy and integrity and election process
Dynamic threshold of PII	Constitutional assembly
Number of block / cycle	Performance and turnover
Number of great electors	Election process

Table 3 : the table illustrate the different parameters of adjustment and the impact on the network and the election process.

The different parameters listed above, reflect the extreme flexibility of the protocol while keeping all the guarantees of security, vivacity and correction (**each parameter counterbalances the others**). The election process based on the constitution of the assemblies ensures a high level of randomness in the selection of the leaders but also a high performance in the execution of the protocol since the assembly is elected for a dynamic number of validations of blocks evolving according to evolution of the network entropy and integrity). The evolution of the network during a mandate will allow a certain renewal in the assembly.

Bot in quarantine

In certain situations, following aberrant or unexpected transactions validated by certain Bots, the network has the power to stop these transactions and to isolate the distracted Bots (via the integrity score). This process is triggered in the system when a **second anomaly** threshold is reached (integrity calculations). This process induces a transaction flow in the network and the state of the Bot will be integrated into the Blockchain register.

Amendment [forks]

In standard operation, the probability of fork is low since within the assembly, the Bots are in strong cooperation with [time limits of validations], this is the main advantage of the election process. In the case where there is a fork, the Blockchain will adopt the longest chain with a maximum weighted entropy and integrity.

Bot compensation

In NeuroChain, the incentive or value creation is done through the validation process related to the election process. During this election, the system will verify the implication of the Bots via the entropy and enthalpy calculations and their integrity through **the evaluation of the reputation score**. In other words, the system measures the level of transparency, the level of relevant certified information and methods injected into the network to pay the Bots. intelligent applications, also, generate remuneration depending on the complexity of the algorithms or **workflows**.

The remuneration is calculated on the basis of the indicator of the real economy and inflation. The promising indicators are: [bid-ask in the network, valuation of cryptocurrency (Bitcoin and Ethereum,…), intrinsic performance of the network and economic inflation]. It reflects the evolution of an asset or value over time, subject to different real impacts.

The remuneration process will allow crypto-value injection in the network during a predetermined period of time (depending on the operation). The crypto-Value will then used in the intelligent application as a momentum. This will ensure a redistribution of the value in the network.

It is important to correlate NeuroChain with the real economy to avoid speculation and bubbles.

NeuroChain Interaction ecosystem

NeuroChain is designed to communicate and interact with all surrounding ecosystem of technologies. First, NeuroChain is a **standard Blockchain compatible** with Bitcoin, Ethereum or other Blockchain with secured message passing (especially concerning smart contracts). It can exchange with web or smart applications via **APIs** to the Bots. It also can interact with **orphan exogenous Bots** and bring them in NeuroChain environment.

Communication latency

The NeuroChain software is designed to be supported by all platforms (lab, mobile or even IoT developed). The consensus algorithm has a low latency that implies a validation of a large number of transactions in **pseudo-real time**. The adaptive communication layer ensures high performance and high resilience of the protocol. The **parallelization task** in the same CPU and on multiple CPUs will improve performance. Finally, the use of Open Source Big Data technologies (**Elastic Search, Neo4J, Spark,…**) will drastically decrease the latency of interactions and the different requests in the network. In general, these open source Big Data solutions will address the problems of real time, parallelization and storage.

First results and performances of the PoC

The distributed architecture with different communication protocols, abstraction layer and new consensus algorithms is developed to test the proposed concept and measure some performance parameters related to Blockchain. The adaptive communication layer is tested and shows promising performance. The different machine learning algorithms [anomaly detection, coherence, semantic analysis,...] are tested separately and optimized for distributed architectures. The main analysis focuses on the consensus algorithm with the statistical analysis of the weighted entropy for collisions

and time evolution. The statistical analysis is presented in [Annex 4: statistical analysis of weighted entropy]. Integrated in the network, the weighted entropy presents a high level of performance in the system with low latencies. *[demonstrator on Github]*

One of the outstanding features of the new consensus is the counterbalance of the parameters involved. In other words, entropy, enthalpy and integrity are inversely correlated during network operation. This means that when the entropy increases due to the high transactional level of the Bots, this activity state is weighted by the enthalpy that measures the strength of each transactional state (in standard operation, an increase in entropy generates a decrease of enthalpy). The same analysis can be done with integrity, because a strong activity of the Bot (↗ entropy) generates a variation of the integrity score (Δ SoI). To illustrate this particular dynamic of these parameters, imagine a Bot trying to interact abusively with its peers to increase its entropy, it will have an impact on the enthalpy (usually weak transactions) and the integrity score. Conversely, a Bot with low activity (↘ entropy) but force transactions will induce an increase in enthalpy and a stabilization of the integrity score. The consensus is intrinsically structured to face some attacks.

The parameters involved in the consensus, are also a measure of network activity, and in that sense, it represents a macro-photography of the Blockchain at time ***t***.

Conclusions

NeuroChain is an augmented Blockchain, which exploits sustainability and distribution characteristics of the Blockchain and adaptability and prediction properties of Machine Learning, Artificial Intelligence and Big Data solutions. The new consensus algorithm based on thermodynamics will allow scalability of the transaction volume, and will therefore tolerate a large variety of smart applications.

NeuroChain makes possible very complicated business and social application, due to the variety of deployed algorithms, which integrity is ensured by the proof of workflow. The decentralised application can be easily implemented in adaptive ecosystem in interaction with its environment.

The era of **Intelligent Blockchain** has begun.

Annexes

Annex 1: traceability Chain

In the traceability chain, two operating states must be distinguished: the standard operating state and the case of malicious transactions.

For standard operations, two cases have to be separated:

1. The NeuroChain in a single traceability process: in this case, only one chain of custody is considered. Transactions are issued by different Bots and considered as events with a causal relationship between them. To ensure the consistency of transactions clock causality similar to Lamport clock causality (Lamport, 1978) is included in the system. Each roBot will have a set of algorithms including consistency and certification algorithms. As previously stated, all Bots will validate transactions and the incorporation into the Blockchain will be done by the leader (with full authority) determined by the issuer. Each transaction validated by consensus will be removed from the transaction pool (a transient database). In the chain of custody, the first transaction (the alpha transaction) in the sense of the live line of the object is emitted by the leader or originator of the traceability line, which generates coherent transactions between the bots.

1. The NeuroChain in interaction traceability: in this case, the traceability chain is in interaction with other chains. The authority (right of validation) of each leader in each traceability chain is shared with the other leaders depending on the correlation between the chains. The correlation measurement is a subtle parameter depending on the shared Bots between the chains or shared objects between them. Another aspect to understand the correlation between the chains is related to the economic zone of the product whose traceability is

implied (only the positive correlations are relevant). This correlation is also related to the notion of entropy. For another aspect, the NeuroChain works as described previously.

For technical aspect, each object or concept in the chain will have a **unique ID** during its all lifetime. All pertinent information related to the object is added as Meta data to the identity stamp of this object. For a better organisation and coherence of the information fluxes in the network, a standard of data structuration of the objects and transactions is realised.

Annex 2: An extract of communication protocols

1. Email software most commonly uses SMTP for sending and **Post Office Protocol 3 (POP3)** or **Internet Message Access Protocol (IMAP)** protocols for receiving mail.
2. Despite its age, there is no real alternative to SMTP in mainstream usage and at this stage; it might be interesting to describe how it works in a few lines. All modern email client programs support SMTP. The SMTP settings stored in an email client include the IP address of an SMTP server (along with the addresses of a POP or IMAP server for receiving emails). Web clients include the address of an SMTP server in their configuration, while PC clients provide SMTP settings that allow users to specify their own server of choice.
3. A physical SMTP server can be dedicated to processing only email traffic, but it is often combined with at least POP3 and sometimes other proxy server functions.
   SMTP runs on TCP / IP and uses TCP port number 25 for standard communication. To improve SMTP and help combat Internet spam, the standards groups also designed TCP port 587 to support certain aspects of the protocol. Some webmail services, such as Gmail, use the unofficial TCP port 465 for SMTP. SMTP moves your email on and across networks using a process called "store and forward". It works very closely with the Mail Transfer Agent (MTA) to send your communication to the right computer and to the good email inbox on the Internet.

   SMTP is usually integrated within an email client application and it is composed of four key components:

   - Local user end utility known as the mail user agent (MUA)
   - A Server known as mail submission agent (MSA)
   - Mail delivery agent (MDA)
   - Mail transfer agent (MTA)

SMTP provides a set of codes that simplify the communication of e-mails between mail servers, which manage the flow of e-mails. Simply, it divides different parts of a message into different categories that the other server can understand. When you send a message, it is transformed into text strings separated by code words that identify the purpose of each section.

By juxtaposing this with the NeuroChain, the SMTP protocol will be used by Bots for specific communication in specific contexts. It is appropriate for communication in business and corporate circles for its simplicity and ease of adoption. This protocol also solves an important security problem in business because it does not require the opening of a new communication port that can be a source of vulnerability.

4. HTTP: The default port is TCP 80, but other ports can be used as well. It provides a standardized way for computers to communicate with each other.

   Clients and servers communicate by exchanging individual messages (as opposed to a stream of data). The messages sent by the client, usually a Web browser, are called requests and the messages sent by the server as an answer are called responses. This communication is represented in the figure xxx.

   HTTPS is a protocol for secure communication over a computer network, which is widely used on the Internet. HTTPS consists of communication over Hypertext Transfer Protocol (HTTP) within a connection encrypted by Transport Layer Security, or its predecessor, Secure Sockets Layer. The main motivation for HTTPS is the authentication of the visited website and protection of the privacy and integrity of the exchanged data.

   HTTPS provides authentication of the Web site and associated Web server with which one is communicating, which protects against “man-in-the-middle attacks”. In addition, it provides bidirectional encryption of communications between a client and a server, which protects against eavesdropping and tampering or forging the content of the communication.

   In conclusion, this protocol is adapted to Bot communication when a high level of security and speed in the exchanges is required.

5. Today, the Internet is based on HyperText Transfer Protocol (HTTP). HTTP relies on location addressing which uses IP addresses to identify the specific server that is hosting the requested information. This means that the information has to be fetched from the origin server or a server within the CDN every time it is requested. Figure 10 describes the distributed architecture of IPFS.

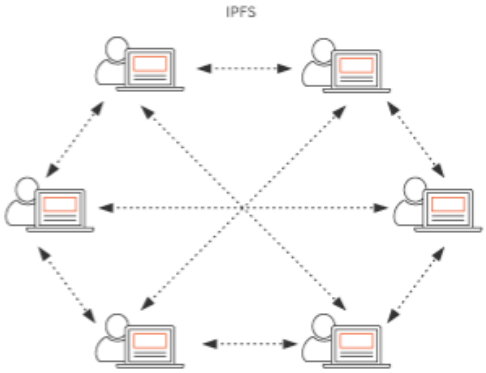
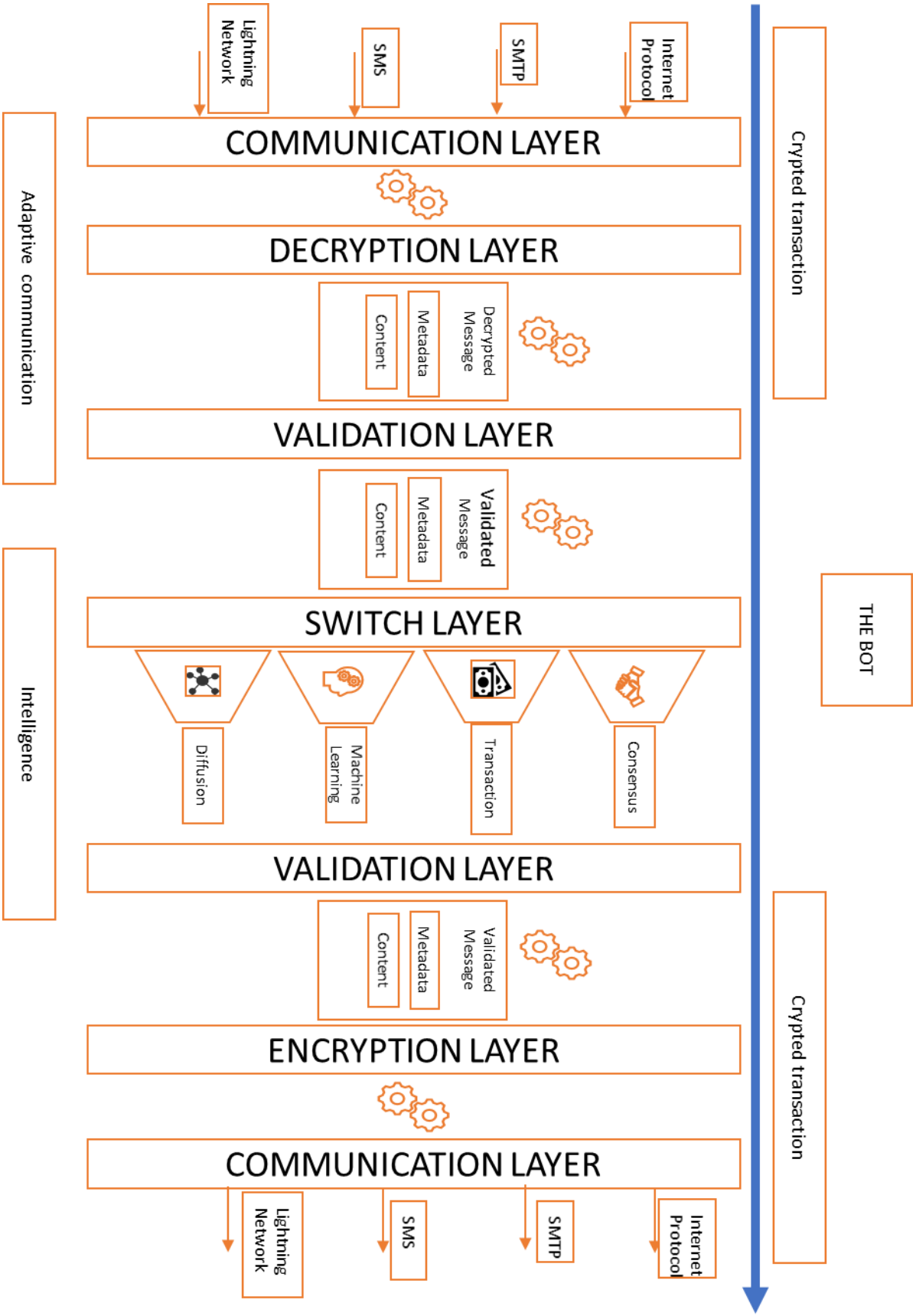


Figure 10: schematic view of IPFS architecture. It is a distributed database of cryptographical signatures of documents.

Annex 3: Technical architecture of NeuroChain



Annex 4: statistical analysis of weighted entropy

The following analysis is devoted to the development of a statistical analysis of Entropy calculation depending on the network characteristics. This analysis will allow constructing an intuition concerning the weighted entropy evolution in the system.

```
library(entropy) # Pour calculer l'entropie
library(igraph)
##
## Attaching package: 'igraph'
## The following objects are masked from 'package:states  ':
##
##    decompose, spectrum
##
##    union
library(ggplot2)
```

- Entropy calculation function

```
info <- function(CLASS.FREQ){
    freq.class <- CLASS.FREQ
    info <- 0
    for(i in 1:length(freq.class)){
     if(freq.class[[i]] != 0){ # zero check in class
       entropy <- -(freq.class[[i]] * log2(freq.class[[i]]))  #I calculate the entropy for each class i here
     }else{
       entropy <- 0
     }
     info <- info + entropy # sum up entropy from all classes
    }
    return(info)
  }
```

The analysis of the transactions between the N randomly generated Bots is as following:

- Construct randomly N vectors containing the id of the Bots exchanging with a specific Bot.
- Affect a random number to each exchange to simulate the number of transactions.
- The number of Bots at each round has to be probabilistic.

```
N = 10000 # number of Bots
list.of.samples = lapply(1:N, function(x) {
 Subsize = sample(80:120)
 Size = sample(200:300)
 subgroupe <- sample(1:N, size = Subsize )
 sample(subgroupe,size=Size, replace = TRUE)})
 echange_length <- lapply(list.of.samples, function(x) length(x))
 echange_tot <- sum(unlist(echange_length))
 echange_fraq <- unlist(echange_length)/echange_tot
```

- Hereafter, an example of exchange vector for Bot 5.

```
list.of.samples[5]
```

```
## [[1]]
##   [1] 9997 6113 2355 3864 9555 2356 9948 7630 8077 1023 5955 9541 5141 9555
##  [15] 4593 2884 8077 5623 2373 9105 7654 5231 5717  952 2017 6113 6109 9247
##  [29] 4783 2884 9986 7691 7691 8077 2884  950 7635 7691 2634 7206 2017 3518
##  [43]  952 9613 7206 4571 3473 9105 9247 5973 6250 8192 3144 1742 9971 5717
##  [57] 6652 3864 9986 5993 5353 9189 7955 8077 3823 2373 2884 2355 2884 9971
##  [71] 1023  756 5231 2884 2373 2355 5231 2356 7117 8116 5165 4200  952 3144
##  [85] 3376 1742 1473 5567 4526 5353 1798 4783 4282 1968 3970  407 6336 4323
##  [99] 2017 9541 6336 4200 9971  189 2355 5567 2758 6113 1023 2634 3376 7630
## [113] 7206 3970 3473 5623 9541 6349 5353 6349 2777  952 1742  952 2467 8116
## [127] 2956 5973 3518 3144 2884 6336 4323 3823 7654 6250 7117 4930 5165 7206
## [141] 7955 3144 4571 2017 6113 5993 3376 6128 9613 9559 6250 2777 5229 1817
## [155] 5955 4575 5973 6128 1023 5567 7630 1798 5165  189 9971 6109 4200 3093
## [169] 2373 3864 5353 3376 2017 9559 5231 5229 4571 9948 8681 1473 4282 4571
## [183] 7691 4783 4561 9555 8116 7117  952 9849 2758 1798 2634 8592 5623  952
## [197] 5141 9971 5955 3473  952 8681 5973  407 5955 5717 5623 5567 7117 1968
## [211] 3093 5973 4602 9971 5353 5231 8681 6652 7955  950 9541 6250 3186  952
```

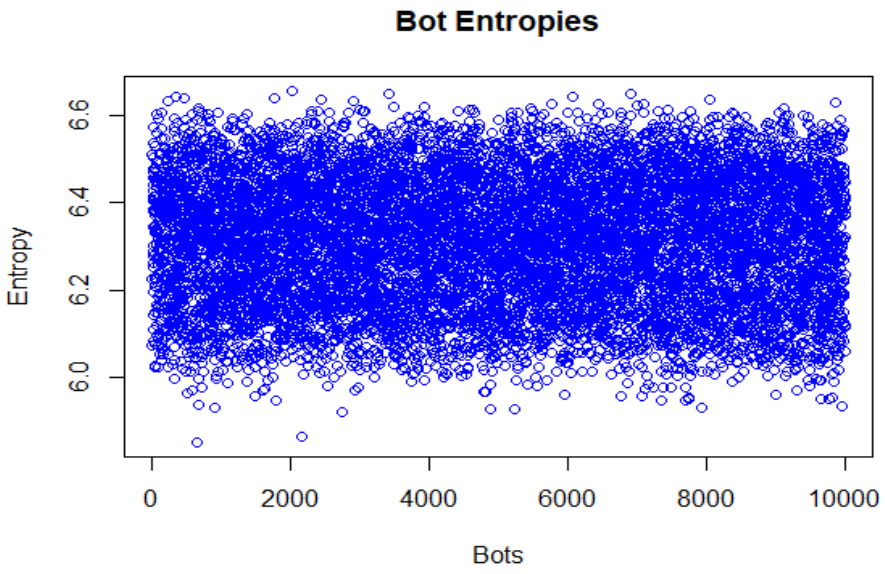
```
## [225] 3144 8357 5850 9189 5353 2758 5231 8116 4593 5567 1968 1798 1798 3093
## [239] 5229 8681 3376 1023 1473 6336 6113 1473 2017 8192 7117 9105 3864 6109
## [253] 3473 3861 3864  952 6109 5850 5993 5231 4200  950 7630 9948 4602 4783
## [267] 1817 9555 4593 6109 3823 9189 2467 5993 6652 3186 7117 2467 7206 2355
## [281] 9541 3093 3140 3861 1742 1798 4526 9189 7206 5231 6652 1742 1023 3823
## [295] 1968 2634
```

- Now the entropy **E = - ∑Pi*log(Pi)** of each Bot in subgroup is calculated. The entropy is then weighted by the fraction of exchanges.

```
E_vect <- vector("numeric", N)
 E_pondere <- vector("numeric", N)
 for (i in 1:N){
  freqs = table(list.of.samples[i])/length(list.of.samples[i])
  E_vect[i] <- entropy.empirical(freqs, unit = "log2")
  E_pondere[i] <-  E_vect[i]*echange_fraq[i]
}
```

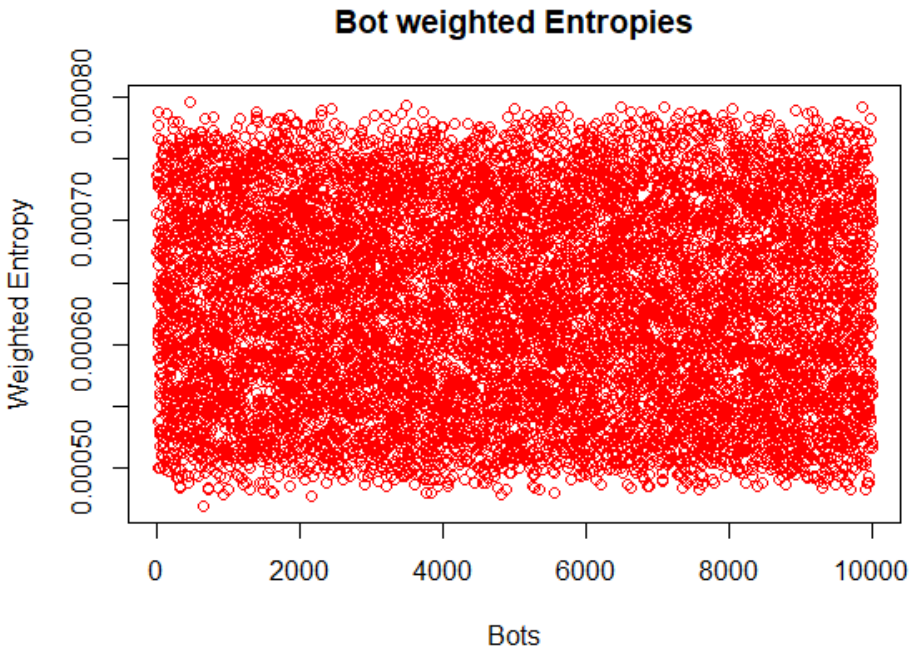
- Illustration of the system

```
plot(E_vect,main = "Bots Entropies", ylab = "Entropy", xlab = "Bots", col = "blue")
#title(main = "Bots Entropies", ylab = "Entropy", xlab = "Bots")
```



- To take into account the volume of exchanges or transactions, the weighted entropy is plotted.

```
plot(E_pondere, main = "Bots weighted Entropies", ylab = "Weighted Entropy", xlab = "Bots", col = "blue")
```



- In stationary state (no time evolution), the impact of the number of Bots in the network, the size of the subgroup of each Bot and the volume of transactions impact the sensibility of the network to the entropy collisions.

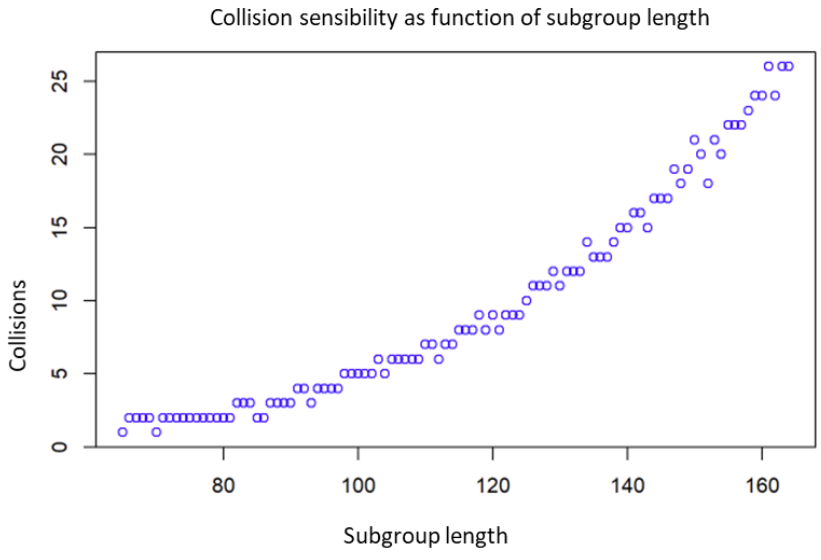
1. **The sensibility depending on the size of the subgroups**

```
N = 5000
M = 100
collisions.count = vector("numeric", M)
for (i in 1:M){
 collisions.mean = vector("numeric", 30)
 for (j in 1:30) {
 list.of.samples = lapply(1:N, function(x) {
 Subsize = sample((110 - i) : (220 - i))

 Size = sample(200:300)

 subgroupe <- sample(1:N, size = Subsize )

 sample(subgroupe,size=Size, replace = TRUE)})

 echange_length <- lapply(list.of.samples, function(x) length(x))

 echange_tot <- sum(unlist(echange_length))

 echange_fraq <- unlist(echange_length)/echange_tot

 E_vect <- vector("numeric", N)
 E_pondere <- vector("numeric", N)
 for (k in 1:N){
 freqs = table(list.of.samples[k])/length(list.of.samples[k])
 E_vect[k] <- entropy.empirical(freqs, unit = "log2")
 E_pondere[k] <-  E_vect[k]*echange_fraq[k] }

 collisions.mean[j] = N - length(unique(E_pondere))

 }

 collisions.count[i] = N - round(mean(collisions.mean))
```

```
}

print("Le nombre de collisions obtenue en dimiuant la taille du subgroupe\n")
collisions.count
```

```
##   [1] 4974 4974 4976 4974 4976 4976 4977 4978 4978 4978 4980 4979 4982 4980
##  [15] 4979 4981 4982 4981 4983 4983 4983 4985 4984 4984 4985 4985 4986 4987
##  [29] 4987 4987 4986 4988 4988 4988 4989 4988 4989 4989 4989 4990 4991 4991
##  [43] 4991 4992 4991 4992 4991 4992 4992 4992 4993 4993 4994 4993 4993 4994
##  [57] 4994 4994 4994 4994 4995 4994 4995 4995 4995 4995 4995 4996 4996 4996
##  [71] 4996 4997 4996 4996 4997 4997 4997 4997 4998 4998 4997 4997 4997 4998
##  [85] 4998 4998 4998 4998 4998 4998 4998 4998 4998 4998 4999 4998 4998 4998
##  [99] 4998 4999
```

```
mean_size = vector("numeric", length(collisions.count))

for (i in 1:length(collisions.count)){
  mean_size[i] = mean(sample((110 - i):(220 -i)))}

plot(mean_size,N - collisions.count, main = "xx ", ylab = "collisions", xlab = " subgroup length", col = "blue")
```

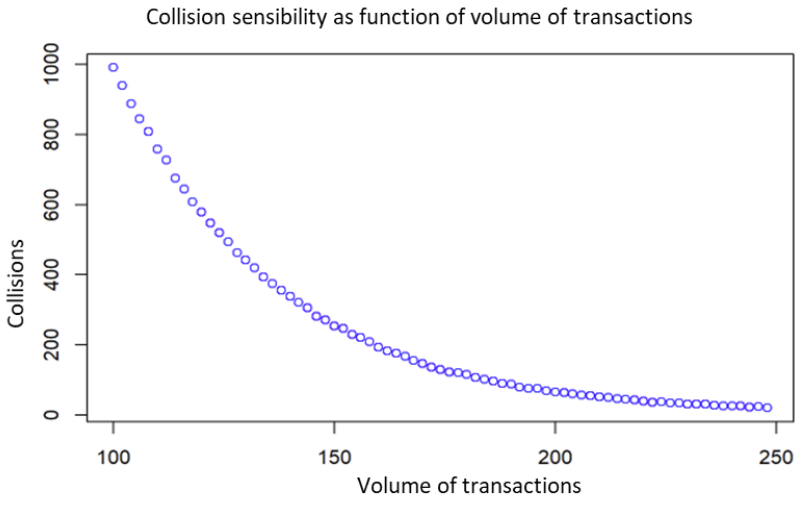


One can notice that the sensibility to the collisions increases with the length of the subgroups (**group of Bots gravitating around a specific Bot**).

2. The sensibility to collisions depending on the volume of exchanges

```
N = 5000
M = 75
collisions.count1 = vector("numeric", M)
for (i in 1:M){
 collisions.mean = vector("numeric", 30)
 for (j in 1:30) {
 list.of.samples = lapply(1:N, function(x) {

 Subsize = sample(100 : 200)

 Size = sample((200 - 2*i):(300 - 2*i))

 subgroupe <- sample(1:N, size = Subsize )

 sample(subgroupe,size=Size, replace = TRUE)})

 echange_length <- lapply(list.of.samples, function(x) length(x))

 echange_tot <- sum(unlist(echange_length))

 echange_fraq <- unlist(echange_length)/echange_tot

 E_vect <- vector("numeric", N)
 E_pondere <- vector("numeric", N)
for (k in 1:N){
 freqs = table(list.of.samples[k])/length(list.of.samples[k])
 E_vect[k] <- entropy.empirical(freqs, unit = "log2")
 E_pondere[k] <-  E_vect[k]*echange_fraq[k] }

 collisions.mean[j] = N - length(unique(E_pondere))
```

```
 }

 collisions.count1[i] =  round(mean(collisions.mean))



}

print("the number of collisions depending on the volume \n")
## [1] " the number of collisions depending on the volume \n "
collisions.count1
##  [1]  20  23  22  26  26  26  28  31  30  30  34  34  37  36  40  43  44
## [18]  47  49  51  55  57  60  63  66  68  75  76  79  87  89  96 101 106
## [35] 116 120 122 130 136 146 155 167 176 183 193 209 221 229 246 253 271
## [52] 281 306 321 339 356 375 393 419 442 463 494 519 547 578 608 645 676
## [69] 727 759 808 845 888 939 991

volume_echange = vector("numeric", length(collisions.count1))

for (i in 1:length(collisions.count1)){
 volume_echange[i] = mean(sample((200 - 2*i):(300 - 2*i)))}

plot(volume_echange,collisions.count1, main = "xxx ", ylab = "collisions", xlab = "volume of transactions", col = "
blue")
```

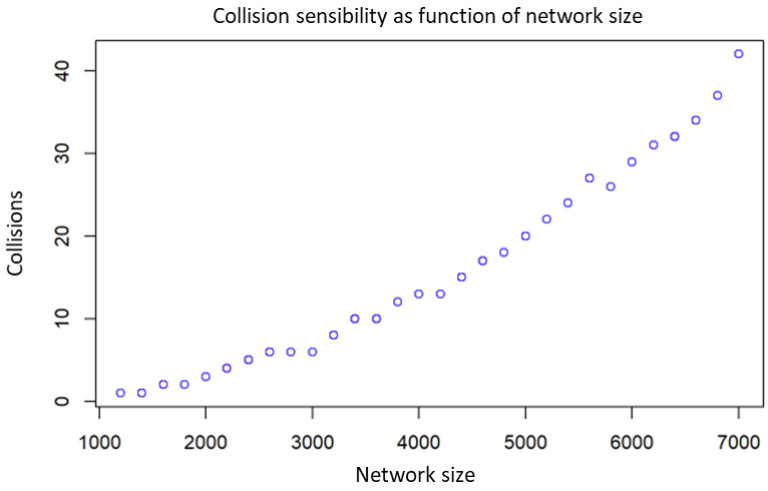


The number of collisions decreases sharply with the volume of transactions.

3. **Sensibility to collisions depending on the size of the network**.

```
N = 7000
collisions.count2 = vector("numeric", 30)
n = 0
for (i in seq(1000,N,200)){
 collisions.mean = vector("numeric", 20)
 for (j in 1:20) {
 list.of.samples = lapply(1:i, function(x) {

 Subsize = sample(100 : 200)

 Size = sample(200:300)

 subgroupe <- sample(1:i, size = Subsize )

 sample(subgroupe,size=Size, replace = TRUE)})

 echange_length <- lapply(list.of.samples, function(x) length(x))

 echange_tot <- sum(unlist(echange_length))

 echange_fraq <- unlist(echange_length)/echange_tot
 E_vect <- vector("numeric", i)
 E_pondere <- vector("numeric", i)
for (k in 1:i){
 freqs = table(list.of.samples[k])/length(list.of.samples[k])
 E_vect[k] <- entropy.empirical(freqs, unit = "log2")
 E_pondere[k] <-  E_vect[k]*echange_fraq[k] }

 collisions.mean[j] = i - length(unique(E_pondere))
```

```
 }


 collisions.count2[n] = round(mean(collisions.mean))
 n = n + 1 }



print("collisions depending on the number of Bots  \n")
## [1] " collisions depending on the number of Bots  \n "
collisions.count2
##  [1]  1  1  2  2  3  4  5  6  6  6  8 10 10 12 13 13 15 17 18 20 22 24 27
## [24] 26 29 31 32 34 37 42
Taille_Systeme = vector("numeric", length(collisions.count2))


for (i in 1:length(collisions.count2)){
 Taille_Systeme[i] =  1000 + i*200}


plot(Taille_Systeme,collisions.count2, main = "xxx ", ylab = "Collisions", xlab = "Network size", col = "blue")
```



It can be seen that collisions increase with the size of the network (the total number of roBots in the system). However, the curve of the increase is smaller compared to the increase due to the size of the subgroup.

Conclusion: the statistical analysis led in this letter is based on two main assumptions: the **stationarity** of the system and the randomness of the transactions. In reality, in operating distributed network, the system is not stationary and the transactions are random but deterministic and in some cases non-

linear. However, this analysis under unrealistic assumptions gives some results of the evolution of weighted entropy as a function of network parameters such as network size, transaction volume, or subgroup size.

The analysis shows that the weighted entropy collisions increases with network size and also increases but faster with the subgroup size. Concerning the volume of transactions, the collision parameter decreases with its increase.

Annex 5: Randomness and Chaotic Processes

A legitimate question when reading this letter is what is the relation between NeuroChain and random process. The answer is simple: in the consensus algorithm of NeuroChain, the leader election in the constitutional assembly is a probabilistic process based on random sampling. This property of randomness is primordial to ensure trust and unpredictability in the system.

In computer science randomness (Prof. Gregory. J. Chaitin) is difficult to achieve since all the programs are deterministic (a sequence of instructions). To generate numbers by chance, there are two approaches, pseudo-random number generator and the true-random number generator. The approaches have different characteristics (Dr Christopher Wetze).

Pseudo-random number generator: pseudo means that the process is not random, strictly speaking, comparing to dice rolls. Pseudo-random number generator are algorithms based on mathematics or pre-calculated tables to produce a sequence of numbers that appear random. An example of algorithms is the linear congruential method. Intense research of modern algorithms has been initiated during the last few years because of its importance for a variety of purposes such as cryptography when generating private and public keys.

True-random number generator: in this case randomness is extracted from physical phenomena and connect it to the computer environment. This is similar to a dice connected to a computer. Scientists use pragmatic physical phenomena that are easier to connect to the computer. In practical, physical phenomena are simple like the little variations of mouse movements or in the amount of time between keystrokes. However, these sources have their limits. One of the good physical phenomena is a radioactive source because the points in time dimension at which a radioactive source decays are fully unpredictable. Also they can be easily be connected to computer. Another phenomena is the atmospheric noise which can be easily captured with a normal radio. There is also entropy based method which for example gathers a variety of sources like web page hits received by the entropy pool in web server.

The following table gives an overview of differences between the two approaches.

Characteristic	Pseudo-Random Number Generators	True-Random Number Generators
Efficiency	Excellent	Poor
Determinism	Deterministic	Nondeterministic
Periodicity	Periodic	Aperiodic

Chaotic process and Quantum events

There are other processes which look like random but they are not. The most popular one is the chaotic process. Chaotic process is deterministic non-linear process which presents two main characteristics:

- A strong dependence to the initial conditions

- High periodicity which means if the system meets at a specific position it will meet again this position an infinity of times

A good example of chaotic process is the so called *butterfly effect* where a small change in initial conditions induces a big effect in the system.

In some cases, chaotic process is a strong method to generate random numbers due to its randomness appearance and its facility to model. The frontier between random and chaotic processes is very tiny since random process could be considered as a very complex chaotic dynamics.

As an example, one can use atmospheric noise to generate random numbers. However to predict these numbers, the position and the speed of each particle of the atmosphere have to be known to model the non-linear system. The difficulty here, could be easily apprehended.

Quantum mechanics is the branch of physics which describes the universe at the atomic and subatomic levels using mathematics. Subatomic particles appear to behave randomly since nothing is known concerning the causes of these events. Consequently, these events are believed to be **inherently non-deterministic**. In this case quantum events may be used to generate randomness.

At the end, randomness mainly depend on its definition, is it something unpredictable by human or other larger thing. Because the universe is deterministic by its existence.

After having a consistent random process for each Bot, the idea is to distribute this process over n Bots. To do that, a strong method is to use mathematical functions to achieve consensus in the network. One proposed method and derived from cryptography is:

$$Ran_n = \sum_n A_{al} \ mod\ n$$

where Ran_n is the random number over the network and A_{al} is the random number for each Bot. Ran_n is expected to be the same for all the Bots. Obviously, Ran_n.

In conclusion, randomness is very important for election process and the distributed random election method used depends on level of security and complexity that can be relevant for NeuroChain.

Key derivation function: In NeuroChain, expandable cryptographic key derivation function to generate the seed in random process, will be tested and used in the election process. The main advantage of the method is that it's structured to face the brute-force attacks.

The seed function in NeuroChain will be based on the sum of the proof of Involvement and Integrity used to constitute the assembly of Bots.

Bibliography

A. Chao and T-J. Shen. (2003). Nonparametric estimation of Shannon's diversity index when there are unseen species in sample. *Environmental and Ecological statistic Statistics* , DOI: 10.1023/A:1026096204727.

A.N. Kolmogorov. (1965). These appraoches to the definition of the concept of quantity of information. *Problemy Peredachi Informatsii 1*, 3-11.

Benet, J. (2016). *Distributed Web* . MIT Licence.

C.E. Shannon. (1948). A mathematical theory of communication. *Bell System Technical Journal, vol. vol. 27*, p. 379-423 and 623-656.

Darwiche Adnan. (2009). Modeling and Reasoning with Bayesian Networks. *Cambridge University Press*, ISBN 978-0521884389.

Dr Christopher Wetze. (n.d.). *Can You Behave Randomly?*

Gelman, A., Carlin, J. B., Stern, H. S., & Donald B Rubin. (2003). Fundamentals of Bayesian Data Analysis: Ch.5 Hierarchical models. *CRC Press. ISBN 978-1-58488-388-3.*, 120.

Giarratano, J. C., & Gary Riley. (1998). *Expert Systems.* PWS Publishing Co. Boston, MA.

Goldreich Oded. (2008). Computational Complexity: A Conceptual Perspective. *Cambridge University Press*.

Gupta, A., Forgy, C., Newell, A., & Wedig, R. (1986). Parallel algorithms and architectures for rule-based systems. *ACM Digital Library*.

(n.d.). *Hypertext Transfer Protocol 1.0.*

Kamvar, S. D., Schlosser, M. T., & H Garcia-Molina. (n.d.). *The EigenTrust Algorithm for Reputation Management in P2P Networks.* http://ilpubs.stanford.edu:8090/562/1/2002-56.pdf.

Lamport, L. (1978). Time, Clocks, and the Ordering of Events in a Distributed System. *Communications of the ACM, vol. 21, no 7*, p. 558-565.

Leslie Lamport. (2004). Lower Bounds for Asynchronous Consensus.

Marshall Pease. (1980). Reaching Agreement in the Presence of Faults. *Journal of the Association for Computing Machinery, vol. 27, no 2* .

Myers, S. A., Zhu, C., & and J. Leskovec. (2012). Information diffusion and external influence in networks. *SIGKDD*, 33.

Nakamoto, S. (2009). Bitcoin: A Peer-to-Peer Electronic Cash System. 4.

P. W. Atkins. (1998). *Éléments de chimie physique.* De Boeck Université.

Prof. Gregory. J. Chaitin. (n.d.). *Exploring RANDOMNESS: algorithmic information theory.*

(1995). *SMTP Service Extension for Message Size Declaration.* RFC1870.

Vandervort D, G. D. (n.d.). Issues in Designing a Bitcoin-like Community Currency. *Brenner M, Christin N, Johnson B, Rohloff K,*.

WEISS, G. (2001). *Multiagent Systems: A Modern Approach to Distributed Artificial Intelligence* . MIT.

Wille, R. (1982). *Restructuring lattice theory: an approach based on hierarchies of concepts.* Rival, I. (ed.) Ordered Sets.445-470.

Yang, J., Chen, B., & and D. Agarwal. (2013). Estimating sharer reputation via social data calibration. *SIGKDD*, 59.