Article

Not peer-reviewed version

The New CAP Theorem on Blockchain Consensus Systems

Aristidis G. Anagnostakis * and Euripidis Glavas *

Posted Date: 18 February 2025

doi: 10.20944/preprints202502.1163.v1

Keywords: blockchain consensus; blockchain optimization; autonomous systems; distributed systems; consensus engineering; consensus optimization; consensus cost; blockchain entropy; consensus entropy

Article

The New CAP Theorem on Blockchain Consensus Systems

Aristidis G. Anagnostakis [1] and Euripidis Glavas [2,*]

[1] School of Finance and Administrative Sciences, University of Ioannina Greece, 45110 Ioannina, Greece; arian@uoi.gr

[2] School of Informatics and Telecommunications, University of Ioannina Greece, 47100 Arta, Greece; eglavas@uoi.gr

* Correspondence: E.G. eglavas@uoi.gr

Abstract: One of the most emblematic theorems in the theory of distributed databases is the Eric Brewer's CAP theorem. It stresses the tradeoffs between Consistency, Availability and Partition and states that it is impossible to guarantee all three of them simultaneously. Inspired by this, we introduce the new CAP theorem for autonomous consensus systems, and we demonstrate that of the three elementary properties, Consensus achievement (C), Autonomy (A) and entropic Performance (P), two at the most can be optimized at any given time. To formalize and analyze this tradeoff, we utilize the IoT micro-Blockchain as a universal, minimal, consensus-enabling framework. We define a set of quantitative functions relating each of the properties to the number of event-witnesses in the system. We identify the existing mutual exclusions, and we demonstrate that (A), (C), and (P) cannot be optimized simultaneously. This imposes an intrinsic limitation on the design and the optimization of distributed Blockchain consensus mechanisms.

Keywords: blockchain consensus; blockchain optimization; autonomous systems; distributed systems; consensus engineering; consensus optimization; consensus cost; blockchain entropy; consensus entropy

---

1. Introduction

In an ideal world, unanimity upon an *event* is implied: everyone agrees if *it took place* or *not* by default. The mechanism for reaching a decision is also common and divine, and works in the same way for all: *true* is *true* and *false* is *false* always and for everyone. Seen under the Aristotelean perspective, the "*harmony of true*" prevails and is always treasured and guarded by all [1]. Autonomy and consensus are absolute with no effort.

In the real world though, this unanimity is not always given. Even if everyone carries the same mechanism for telling *true* from *false*, and even if everyone always acts *rationally*, *consistently*, and *in good faith*, the fact of the atoms' *finiteness* suggests that *subjectivity is not guaranteed* [2]. Someone might eventually reach a contradictory conclusion, even upon a commonly observed *event* (e.g. while I *see a cup*, you *see a pot*). However *true* for the others, things may escape his view, due to limitations in memory, power, processing capacity, communications' latencies and deficiencies [3].

The process of reaching and proving consensus in Blockchain systems is known to be significantly *energy demanding*: *intensive processing, information exchange*, and *storage* has to take place among and within the atoms (the nodes). Reaching and proving *consensus* comes at a significant *cost* [4–7].

In the world of Blockchain systems, the process of *consensus* semantically coincides with that of *witnessing* [8]. The number of the event-witnesses that are required every time varies with the details of the architecture of each system. It largely defines the consensus dynamics of the system as well as its overall *performance* traits. To reach the desired level of *consensus*, without compromising the *availability* and without overburdening the resources, the Blockchain systems tend to adopt *probabilistic over absolute finality* and *eventual* over *strong consistency* practices [9].

Still, as we demonstrate in this work, there are fundamental and universal tradeoffs between *consensus*, *autonomy* and *entropic performance*, irrespective of the details in the implementations of each Blockchain system.

1.1. Motivation

During the early stages of the evolution of cloud computing, a number of significant traits and constraints were revealed. One of the most definitive ones is described in the E. Brewers' CAP theorem [10,11], which highlights the tradeoff between *Consistency*, *Availability*, and *Partition* in distributed database systems.

In our time, Blockchain incarnates the dream of modern technologists for autonomous peer and inclusive system operation of virtually *infinite distribution*. The limits and the constraints of Blockchain consensus mechanisms are beginning to reveal and set boundaries on the feasibility of infinitely distributed systems [12,13].

Up to now consensus mechanisms are typically studied in the context of the containing Blockchain systems they serve. Yet, consensus begins to be perceived as a *distinct* self-contained mechanism and its *performance* (like all mechanisms), to be measured under the prism of *entropy* [14].

Consensus mechanisms come with significant tradeoffs. In this work, we explore the intrinsic tradeoffs among the fundamental properties of *Autonomy*, *Consensus achievement* and *entropic Performance* in the distributed consensus systems.

1.2. Contribution

In this work we introduce the new CAP theorem in the context of distributed consensus engineering.

In semantic analogy to the Eric Brewer's CAP theorem, which formalizes the tradeoff between Consistency, Availability and Partition tolerance in database systems, the *new CAP theorem* highlights the fundamental constraints existing between *Autonomy*, *Consensus achievement* and *entropic Performance* in distributed collective consensus systems.

In this work we study the process of consensus as a distinct self-containing process and we introduce a set of novel quantitative definitions:

- *Autonomy* is defined at the *atomic scale*, as the fraction of the memory of each node reserved to serve local operations.
- *Consensus achievement* is defined at the *system scale* as the fraction of nodes required to reach agreement (i.e., consent) on new events for the system to function.
- *Entropic Performance* is introduced, following [14], as a metric for the efficiency of the consensus process quantified by the overall reduction in the information entropy of the system per unit of consumed energy.

Throughout this work we recognize the notion of *witness* as the essential link among the *Autonomy* (A), the *Consensus* (C), and the *entropic Performance* (P) of the system.

- We prove that <u>of these three essential properties, two at the most can be optimized simultaneously at any given time</u>.
- We demonstrate that this trait comes in a direct analogy and has the same semantic origins as Eric Brewer's CAP theorem.

The generality of the findings in this work is reinforced by the universality of the witnessing process, which appears to underlie every blockchain system, irrespective of its architectural considerations and implementation details [2,14,15].

2. Materials and Methods

To keep our approach universal and architecture agnostic, our analysis relies on two major pillars:

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 18 February 2025 doi:10.20944/preprints202502.1163.v1

- The deployment of the IoT micro-Blockchain, defined in [8], as a fully distributed and neutral framework. This constitutes the link of our analysis to the physical world.

- The quantitative representation of (A), (C) and (P), with respect to the number of event-witnesses in the system.

Following in this section we quote their basic traits.

2.1. The IoT Micro-Blockchain Framework

The IoT micro-Blockchain implements a *neutral universal consensus framework* that exists in every atom of the realm and governs its primeval functionality. In this framework, Blockchain is considered in its most generic form, as the aggregation of interconnected atomic hash-chains which are stored in the finite local memory of the tiny IoT nodes. Its source code can be found in https://github.com/arianagnostakis/IoT_Blockchain (accessed on 12/2/2025).

Our system is self-inclusive, and new events are only taking place inside the nodes. Consensus relies on common *event-witnessing*. The process of witnessing is demonstrated in Figure 1, which is given here after [8]. It demonstrates a new local event which is raised and stored locally in node N_2, as well as transmitted and witnessed by 2 sibling nodes.

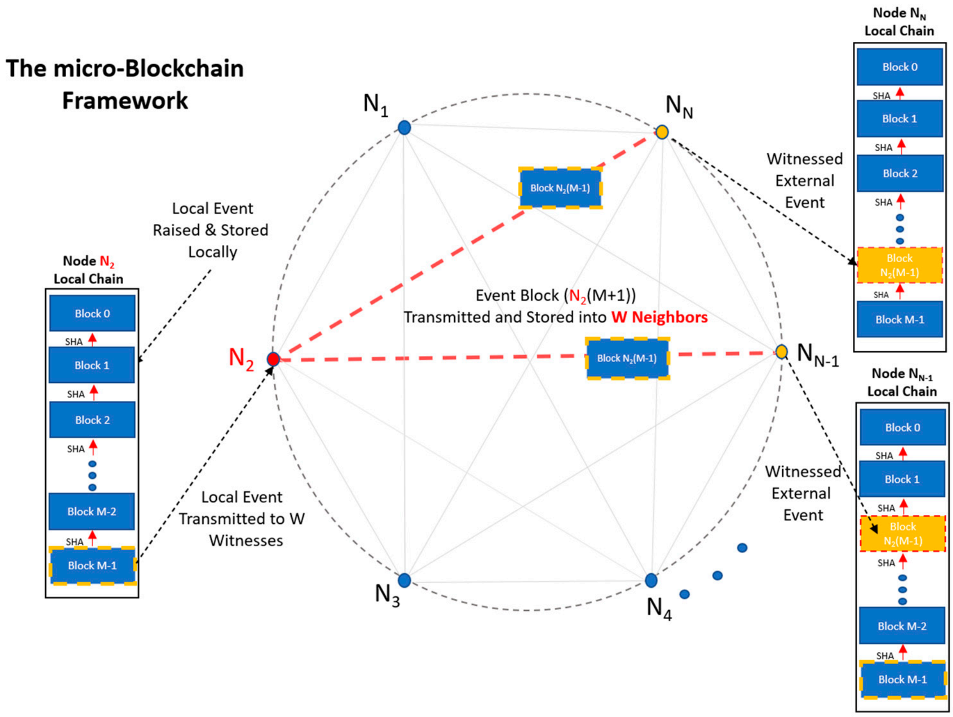

Figure 1. The process of witnessing in the IoT micro-Blockchain network (published after [8]).

Generally, in the light of a *new event, W* siblings are being contacted to validate and record it in their local memories. The whole consensus process relies on this simple *witnessing* function: *every witness* of an event *agrees* (i.e., *consents*) with *every other* witness of the *same event*. This way the collective agreement mechanism that governs consensus is built.

The desired number of event witnesses *W* varies with respect to the needs of each application. In an absolute-finality Blockchain system, such as the ones that are developed exclusively to facilitate monetary transactions, (e.g. early-stages Bitcoin) [16], an absolutistic requirement of *W*=*N* at all times (where N is the number of the peer nodes in a system) is often raised. In such a case, the content of the memories of all nodes becomes identical; the redundancy of the Blockchain, as well as the consensus over it becomes absolute and maximum.

In a system consisting of finite capacity entities, such as the IoT world, as well as every *digital world*, this absolutistic requirement cannot always be guaranteed. This induces the consideration of

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 18 February 2025 doi:10.20944/preprints202502.1163.v1

consensus as a scalar attribute in its generality. A distributed world of finite-capacity entities has often got to work with W<<N.

The memory of the autonomous node can be abstractly modeled as in Figure 2. Here, the memory of each node consists of two parts: (a) the "ROM-like" part in which the "inherited code" implementing the *common consensus process* is stored, and (b) the active "RAM" part in which the event data are stored and has the form of hash-chain. The *active memory* (b) can be further logically divided into two fractions, storing either *local*, or *external* events' data. The events that are validated and stored in each node are either *local*, i.e. the event was raised within the node or *external* i.e. the node *witnesses* an event that came up on a sibling node.

We utilize this simple model to define the Autonomy of the node as the ratio of the fragment of the local memory utilized by the node to store its *local events* (*M_Local*) to the *total active memory* of the node (*M*) later in Section 3. At a fundamental level, this also represents the fraction of the total resources of the node dedicated to processing the local events.

The logical fragmentation of the memory of the node is depicted in Figure 2.

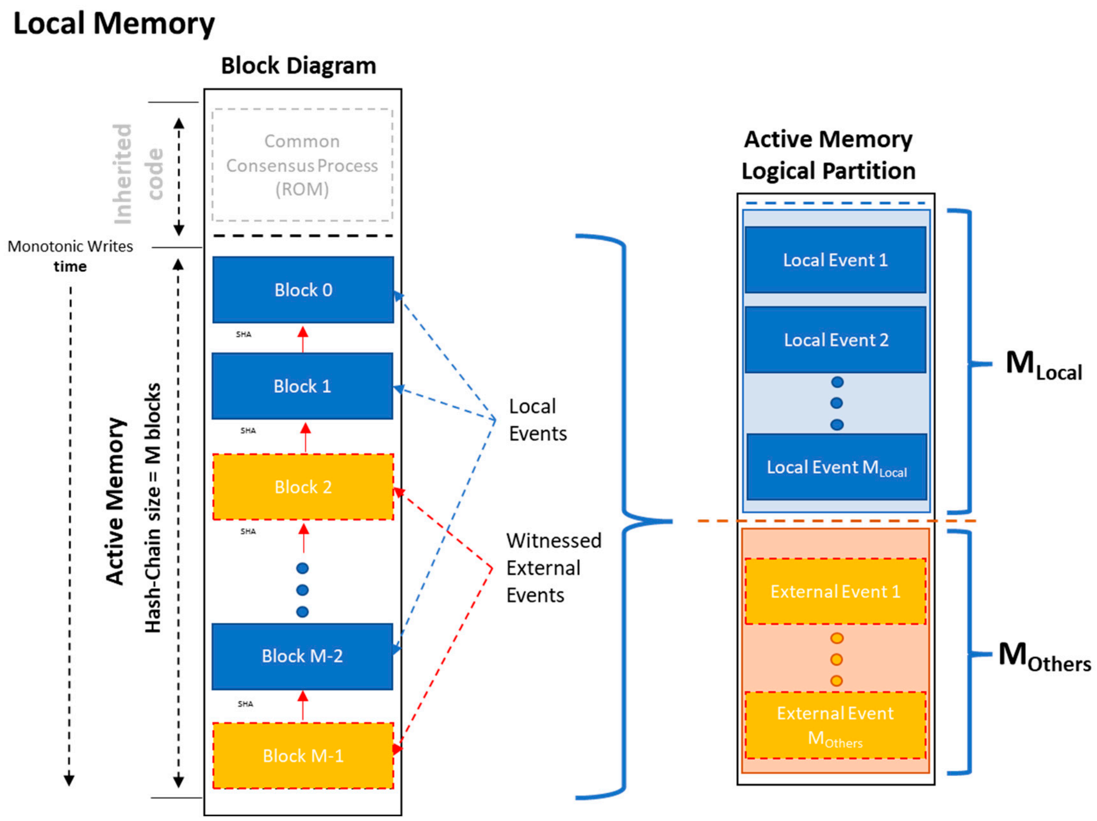

Figure 2. A simple model of the memory of the autonomous node ($M=M_{Local}+M_{Others}$).

2.2. Consensus, Autonomy and Entropic Performance

The basic traits of *Consensus (C)*, *Autonomy (A)* and *entropic Performance (P)* of a universal consensus mechanism are given here, to facilitate their formal definition in the next section.

***Consensus (C):* Constitutes a collective attribute and can thus be defined only within a system of atoms**. In its essence, consensus is a binary function: in the end, every atom may either *agree* or *disagree* with *the others* over an *observable event*.

In its general form, consensus constitutes metric of the awareness of "how close" the separate perceptions of the atoms in a realm are upon an *observable event*. If only infinitesimal events are considered (i.e. single-bit events like the opening of a door, the crossing of a temperature threshold, or the validity of a monetary transaction), then it is easy to identify consensus as the fraction of the

 doi:10.20944/preprints202502.1163.v1

population that *witness* a specific event (i.e. they *are aware of* and *agree on the event*). In every case though, the outcome of consensus remains a binary agreement/disagreement flag: e.g. "Does the hash produced in block X with nonce Y start with 5 zeros in a row?" (yes/no). This is analyzed in [14].

Autonomy (A): **Autonomy is a metric of the independence of the atom.** It constitutes an atomic trait, defined here as the fraction of its memory each atom can dedicate for serving itself. As mentioned earlier, at a fundamental level this also represents the fraction of the resources dedicated to process and store the nodes' local events, (considering the fact that all events are alike, and that the cost of processing and storing *local* and *external* events is the same).

At an abstract level, in a peer distributed system, every peer node operates in *conformity* with its siblings. To achieve this, part of its resources are contributed to serving the others (the community). Conceptually seen, the *existence* and the *coherency* of the community itself rely on this tribute from the part of each peer node.

In a peer collective consensus environment, autonomy is expected to increase, when the number of per event required witnesses (*W*) is reduced. A fully independent atom should operate consistently without the necessity of any event-sharing with *the others*.

In a fully autonomous system, as *witnessing* tends to zero (*W->0*), the atom tends towards absolute autonomy (*A->1*) allocating all its resources to serve itself. Still, to comply with the notion of *system*, *W*, however small, can never be *equal* to zero, since this would lead to a system of totally isolated nodes with zero external awareness: a set of unconnected, lonely nodes.

Entropic Performance (P)**:** At the cost of running the consensus mechanism, the observed information entropy of an isolated autonomous system decreases as *W* increases. Higher witness counts reduce the diversity of information stored in each node, leading the system to overall lower entropy states. This is discussed in detail in [14] where in addition it is demonstrated that this decline is steepest for low values of *W* (i.e. when fewer overall witnesses exist in the system).

In this work we introduce the *entropic performance* of the system as the ratio of the observed *information entropy reduction* to the overall *energy consumed* to achieve it, with respect to *W*.

Following we formalize the definitions, and we highlight the tradeoffs existing between the *Autonomy (A)*, the *Consensus achievement (C)* and the *entropic Performance (P)*.

3. Analysis and Results

In this work we intend to highlight the principles. For keeping our analysis simple, we consider the *average node* as the representative unit of the system in the sense that the nodes in a peer distributed environment are considered equivalent, and following the same principles. Building on the definitions in [8] and without hurting the generality of our approach, we consider that the nodes in our system exhibit uniform behavior. The probabilities of new events' occurrences and witnessing are evenly distributed among the nodes.

3.1. Formal Definitions

Autonomy (A): Autonomy is formally defined here as the fraction of the memory of the node that is kept to serve the "self" with respect to the total memory of the node and is given by:

$$A = \frac{M_{Local}}{M} \Rightarrow A = \frac{M_{Local}}{M_{Local}+M_{Others}} \quad (1)$$

where M the active memory of the node, M_{Local} the memory dedicated to the local events and M_{Others} the memory used to store external events and as it is defined in Figure 2 and discussed in detail in [8].

Considering the *witnessing requirement* of the system, every event in the *local memory* of the node must be witnessed by *W* siblings. With respect to the uniformity of the nodes, this as well suggests that every node in the system witnesses on average the events existing in *W* local memories of other nodes, leading to:

$$M_{Others} = M_{Local} * W \quad (2)$$

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 18 February 2025 doi:10.20944/preprints202502.1163.v1

This quantifies the witnessing requirement and defines the fragment of M_{Others} to M_{Local} within each node with respect to the number of event-witnesses W.

Since

$$M = M_{Local} + M_{Others}$$

we get

$$M = M_{Local} + M_{Local} \cdot W \Rightarrow M_{Local} = \frac{M}{1+W} \quad (3)$$

where again, M is the total active memory in a node and W the number of witnesses on each event.

By substituting M_{Local} and M_{Others} in the Autonomy formula (1) we get:

$$A = \frac{\frac{M}{1+W}}{M} = \frac{1}{1+W}, \ W \in (0, \mathrm{N}-1] \ (4)$$

where N is the total number of nodes in the realm.

This defines A with respect to W and indicates that as W increases, A decreases. It aligns with the intuitive assertion that more witnessing, reflects increased participation in the consensus process and thus reduces the *Autonomy* of the node.

Consensus (C): We define the *quantity of consensus* as the fraction (the normalized count) of the autonomous nodes that agree over an observable event every time. This corresponds to the *average number of witnesses* of each new event (W), with respect to the overall population (N). It takes the form of a scalar ranging $1/N$ (no one but the introducer of the event witnesses it) to 1 where every node witness every event. This is analyzed in depth in [14].

We model the overall collective consensus of the system as the fraction of the nodes witnessing each event, to the total population:

$$C = \frac{W+1}{N}, \quad W \in (0, N-1] \quad (5)$$

where again, W is the number of witnesses on each event and N the total number of nodes in the system. The (+1) component in the numerator accounts for the node initiating the event, which is always part of the consensus process. C approaches 1 as W+1 approaches N, i.e., when all nodes participate in the consensus process for every event. C becomes minimal ($C \to \frac{1}{N}$) when $W \to 0$, indicating a system tending to zero consensus, (i.e. a system of isolated nodes).

Substituting W in eq.(4) with respect to eq.(5), leads us to:

$$A = \frac{1}{C \cdot N} \quad (6)$$

which relates the *Autonomy (A)* with the *collective Consensus (C)* and reveals the constraint between A and C.

Entropic Performance (P): is defined here as the entropy drop (bits) occurring per unit of energy consumed in the system (Wh). The consensus process, while seen as a distinct mechanism from the perspective of the second thermodynamic law, is an entropy conversion mechanism.

The entropic traits of the consensus process have been studied in [14]. There, the information entropy in the system is defined with respect to W as:

$$H = -\log_2\left(2^{-\left[N \times \frac{M}{1+W}\right]}\right)$$

which simplifies to:

$$H = \frac{N \times M}{1+W} \quad (7)$$

where N is the number of peer nodes in the realm, M the active memory capacity of each node and W the number of per event-witnesses.

Eq. (7) describes the information entropy of the system with respect to the number of per event witnesses (W) and highlights that increasing the number of witnesses proportionally reduces the information entropy.

The rate of the *information entropy reduction* with respect to W is also given in [14] by the derivative of H over W:

$$\frac{dH}{dW} = -\frac{N \cdot M}{(1+W)^2} \qquad (8)$$

where again, N is the number of the nodes, M the active memory of each node and W the number of per-event witnesses in the system. Eq. (8) designates that the *information entropy reduction* in our system is steepest for low (still >0) values of W.

Let's now consider the energy required for introducing an additional witness of an event in the system (E_w).

While this depends on the details of each architecture and setup (i.e. Bitcoin, Ethereum PoW/PoS, Hyperledger Fabric, etc.) [16–18], several common factors can be identified. These include the total number of nodes, the per memory-cell power consumption, the per-witnessing data transfer and processing required, the existence/absence of a broadcasting channel among the nodes, and the conflicts rising due to the new events' frequency and distribution among the nodes.

The exact value of the energy consumption of the various real-world systems is out of scope of this study, and subject to other on-going work. Still the average per-witnessing energy consumption in one node of our testbed (micro-Blockchain framework running on a setup of Arduino Nano 33 IoT devices) is given here as a reference point. In a sparce neighborhood of nodes operating at a relatively low frequency of local events (i.e., *N=100*, local events frequency *~10⁻² Hz*) the energy consumption per event-witnessing is approx. 55 μWh. This considers the energy required for *one event block* to be transmitted, received, validated, and stored, *in one witness*. The technical specifications of the reference hardware can be found in [19].

(Please note that the above data are only given as a reference point to the reader, and do not affect the findings, the results and the conclusions of this work.)

The total energy invested in the system to acquire W event-witnesses is then given by:

$$E_{event} = W \cdot E_w$$

This corresponds to the total amount of energy consumed in the system per event in order to support the consensus process and is equal to the energy consumed by all W witnesses of the event. Consequently, the smaller the energy consumption in a node per event witnessing (E_w), the more efficient the system is overall.

Following, the *entropic Performance* of the consensus mechanism is defined as the occurring entropy drop dH in the system per unit of energy E_w consumed:

$$P_{erformance} = \frac{1}{E_w} \cdot \frac{dH}{dW} \xrightarrow{(8)} P_{erformance} = -\frac{1}{E_w} \cdot \frac{N \cdot M}{(1+W)^2} \quad \text{(bits/Wh)} \qquad (9)$$

where again, E is the energy consumption per event-witnessing, M the active memory of the node, N the number of nodes in the system and W the number of per-event witnesses.

This corresponds to the entropy reduction occurring in the of the system per unit of consumed energy.

4. The New CAP Theorem

Theorem 1. *Of the three elementary properties of the autonomous systems, Consensus achievement (C), node Autonomy (A) and entropic Performance (P), two at the most can be optimized at any given time.*

Proof of Theorem 1. To prove the assertion, *following we investigate the traits of the properties. In* sections 4.2 and 4.3 we demonstrate the two mutual exclusion *conditions* holding among of the properties in the form of mutual constraints.

 doi:10.20944/preprints202502.1163.v1

4.1. Autonomy and Consensus as Functions of W

Figures 3a,b depict Consensus (C) and Autonomy (A) as defined in eq. 4 and 5 respectively, with respect to W in a dynamic system of 100 nodes.

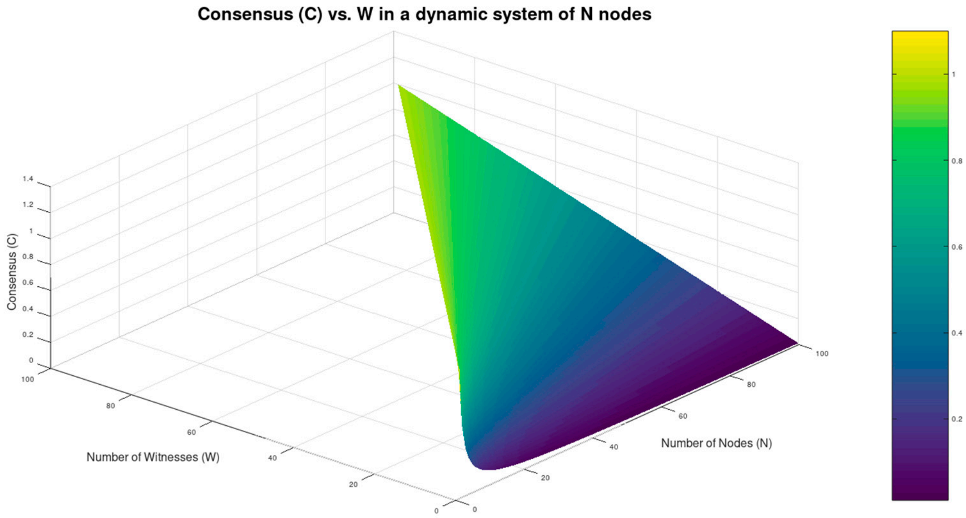

Figure 3. a: Consensus as a function of W in a dynamic system of 100 Nodes, W varying from 0.1 to N-1.

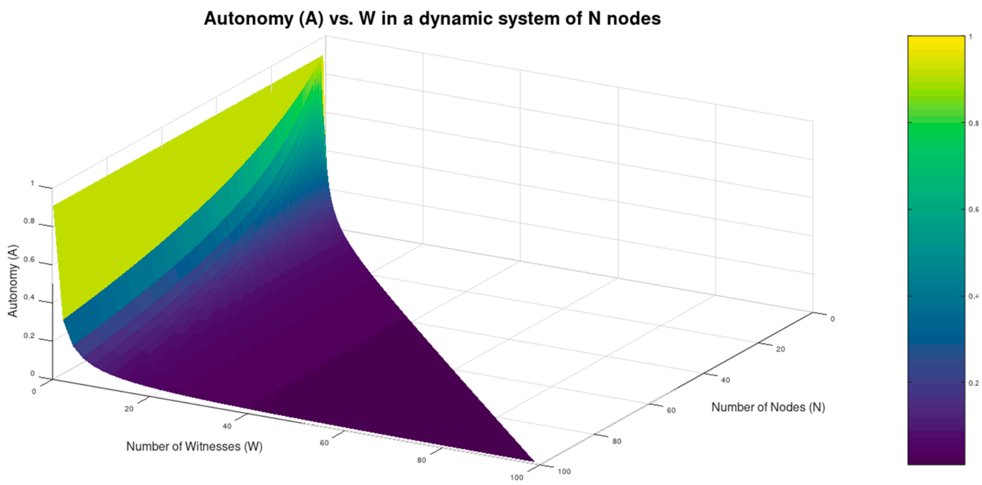

Figure 3. b: Autonomy as a function of W in a dynamic system of 100 Nodes, W varying from 0.1 to N-1.

The macroscopic behavior of C and A as demonstrated in Figures 3(a,b), is in strict conformance with the definitions: as expected, Consensus increases monotonically with W, while Autonomy declines rapidly. In Figures 3.a and 3.b the axes are left in their original scale to demonstrate the boundaries as well.

Based on eq. (4) and (5), we have $\frac{1}{N} < A < 1$ and $\frac{1}{N} < C \leq 1$ respectively.

Both the Consensus and the Autonomy become optimal as they approach 1 (see definitions in the previous section).

Still, as we demonstrate below in Figure 4 this cannot happen simultaneously for both.

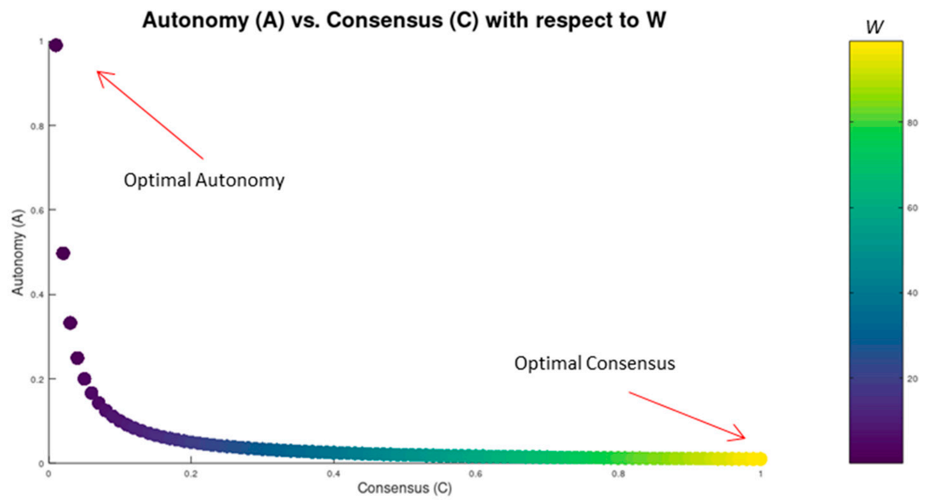

Figure 4. Autonomy (A) vs Consensus (C) as functions of W.

4.2. Autonomy (A) vs Consensus (C)

The tradeoff between (A) and (C) is evident through the eq. (6). This imposes a condition of Mutual exclusion among them in a requirement for the concurrent optimization of the two properties.

Figure 4 shows the existing tradeoff between C and A with respect to the number of event-witnesses (*W*).

The scatter plot in Figure 4 depicts the outcome of eq. 4 and eq. 5 with respect to *W* and demonstrates that Autonomy and Consensus cannot be optimized simultaneously: While consensus increases with *W*, the resources remaining to serve the local events in the nodes decrease, and along with them, the Autonomy of the node declines as well.

Figure 4 clearly demonstrates <u>this mutual exclusion condition</u> and constitutes the first proof of the initial assertion (Theorem 1).

Still, this is not the only one. As we demonstrate following in Figure 5, neither *Consensus* and *entropic Performance* (P) can be optimized simultaneously, raising the second mutual exclusion condition.

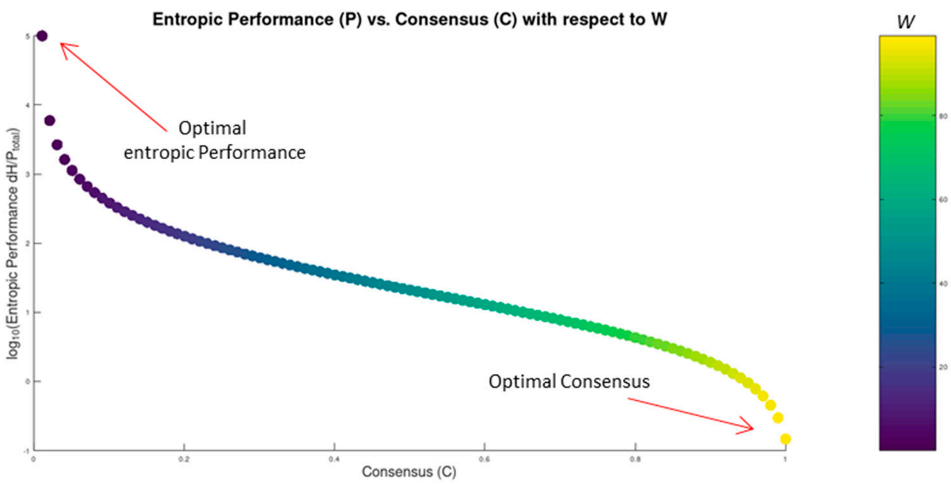

Figure 5. Consensus *(C)* vs entropic Performance *(P)* given as functions of *W*.

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 18 February 2025 doi:10.20944/preprints202502.1163.v1

4.3. Consensus (C) vs Entropic Performance (P)

The entropic Performance (P) (i.e. the efficiency of the system) is defined in eq.9 as the observable decrement in the information entropy of the system per unit of consumed energy, and extends the definitions of [14].

The scatter plot in Figure 5 depicts the tradeoff between the entropic Performance (P) (eq.9) vs Consensus (C) (eq.5) as functions of W.

As demonstrated in Figure 5, Consensus and entropic Performance cannot be optimized simultaneously: While the higher Consensus occurs for high values of *W*, the entropic Performance is optimal (maximizes) for low values. This also constitutes a solid proof of the generic intuitive assertion that "higher consensus mandates higher power consumption". The relation of the entropic performance with respect to the degree of replication in a system was first revealed in [14] and raises here a mutual exclusion condition among C and P in a request for concurrent optimization. **Again, this also constitutes proof of Theorem 1.**

It also mandates that in order to increase the efficiency of the Consensus mechanisms, the systems need to operate on the lower possible number of per event witnesses with respect to the overall stability and consensus requirements every time.

4.4. Autonomy (A) vs Entropic Performance (P)

The entropic performance of a system increases with the increment of the autonomy in the system. This is depicted in Figure 6.

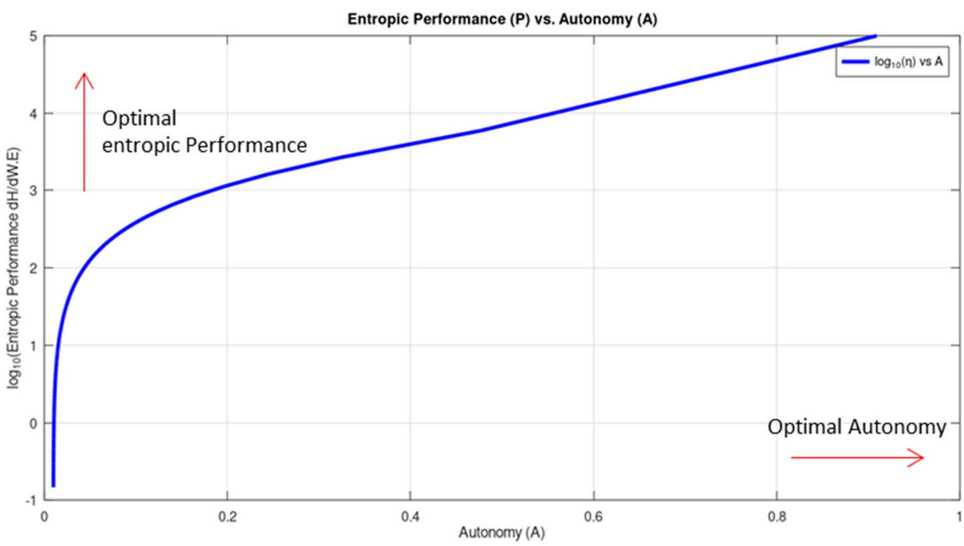

Figure 6. Entropic Performance vs. Autonomy.

The Autonomy of the system increases decreasing the number of per-event witnesses, while at the same time the entropic performance of the system becomes optimal. Figure 6 demonstrates that both *autonomy* and *entropic Performance* can be simultaneously optimal.

4.5. Entropic Performance (P) as a Function of W

In Figure 7 we see the entropic Performance of a dynamic system of 100 nodes. It presents the outcome of eq. 9 and demonstrates once more that the *steepest entropy reduction* per *energy unit paid* occurs for *low values of W*. This also highlights that the entropic Performance tends to optimal (maximizes) as $W \rightarrow 0$.

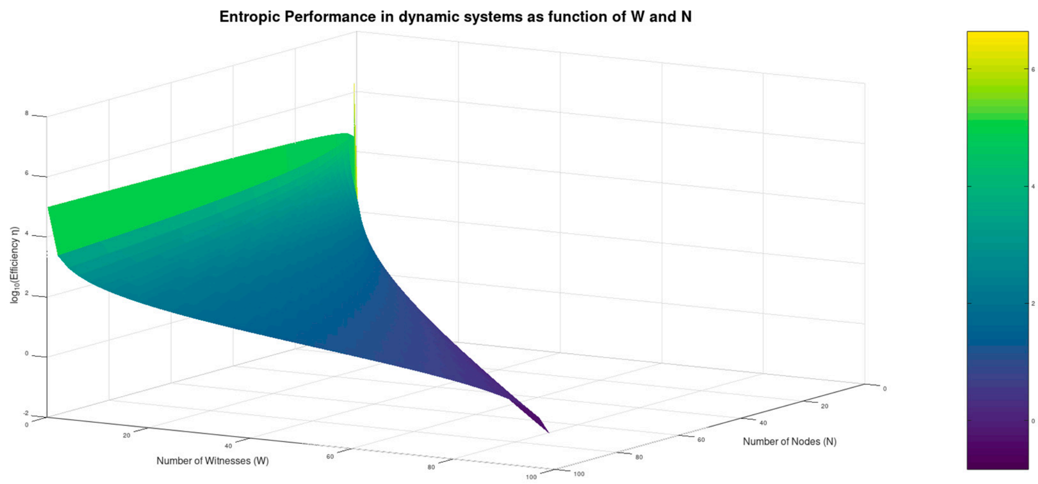

Figure 6. 3-D plot of the entropic Performance (efficiency) of a dynamic system of N nodes for *N*=1:100, *W*=0.1:*N*-1, *E*=1 and *M=1,000* bit.

As it is proved in [14], the steepest decline in the information entropy of the system occurs for low values of *W*, meaning that the act of witnessing is more efficient in terms of information entropy reduction per unit of consumed energy while the fewer witnesses exist in the system.

This also suggests that *diversity* in a system increases its overall *entropic Performance*.

5. Discussion

Seemingly independent properties in nature often come with inherent constraints. This principle is elegantly captured in the great conservation laws where it relates to the existence of underlying *symmetries* [20,21]. These laws, fundamental to theoretical physics and mathematics, highlight an intrinsic truth: some properties are *constrained* and cannot vary without affecting others. Distributed systems are no exception to this trait.

One of the most influential theorems in the field of distributed database systems, Eric Brewer's CAP theorem, reveals such a constraint among the properties of *Consistency*, *Availability* and *Partition tolerance*: it asserts that these cannot be simultaneously optimal. This assertion, initially introduced by Brewer as a conjecture in a keynote speech in 2000 [10] was formally proved by S. Gilbert and N. Lynch two years after [11].

Building on these concepts, in this work we introduce the *new CAP theorem* for distributed Blockchain systems. We demonstrate that *Consensus achievement* (C), *Autonomy (A)*, and *entropic Performance (P)* in distributed collective consensus systems cannot be optimized simultaneously.

The initial assertion is proven through two mutual exclusions:

(a) Between *Autonomy (A)* and *Consensus achievement (C)*

In an attempt to optimize one of two, the other is sacrificed. This derives from eq.4, eq.5 and eq.6 in section 3.1 and is demonstrated in Figure 4, section 4.2. An attempt to optimize *Autonomy* (moving $A \rightarrow 1$) would leave the nodes without any resources to serve the community: *W* moves close to 0, leading *Consensus achievement* to minimum and the system degrades down to a set of isolated nodes. Again, trying to optimize *Consensus* ($C \rightarrow 1$), we would have to withhold resources from the atoms to serve the system, sacrificing *Autonomy*.

This intrinsic constraint was revealed in this work by starting from two apparently independent starting points: while we define *Autonomy* in the micro-scale of the atom, *Consensus* is defined in macro, with respect to the properties of the realm. This strengthens the hypothesis for the existence of an inherent constraint among *A* and *C*.

(b) Between *Consensus achievement (C)* and *entropic Performance (P)*

Consensus achievement is an energy-consuming process: it relies on data *transmission, processing* and *storage,* all of which are known to be energy consuming tasks. The constraint between (C) and (*P*) is revealed in section 3.1 through eq. 5 and eq. 9 and is demonstrated in Figure 5 section 4.3. Again, trying to optimize one of the two, the other is forced away from optimal. Our consideration for the two properties has independent starting points as well: while C is defined with respect to the macroscopic traits of the system, P is defined based on Shannon information entropy principles. This further strengthens the finding of the inherent constraint among C and P. The *entropic traits* of distributed consensus systems are studied extensively in [14].

In this work we extend the scope by applying the notion of energy consumption in our considerations. We treat *Consensus achievement* as a distinct, self-contained process, and we exploit its entropic features to introduce a new metric on its efficiency, as the entropy reduction it induces to the system per unit of consumed energy. This paves the way for further Research and Development aiming on Blockchain optimization.

In this work we exploit the architectural neutrality of the IoT micro-Blockchain framework to formalize (A), (C) and (P) with respect to the primitive traits of distributed autonomous systems.

Throughout this work a great amount of effort was put in keeping our approach as simple as possible. In many points we opted for the simpler applicable approach to highlight the principles and to guard the universality of the findings.

One such is the *uniformity of the nodes*: our findings are proved in this work under a *uniform system consideration.* Still, the same principles underly and the same assertions hold for every finite-capacity system and bounded-probability distribution consideration. Further exploration of these cases is part of other ongoing work.

5. Conclusions

In this paper we introduce the new CAP theorem for Blockchain consensus systems.

We demonstrate that from the three essential properties *Autonomy (A), Consensus achievement (C) and entropic Performance (P)*, two at the most can be optimized at any given time. This imposes an intrinsic limitation on the design and the optimization of distributed consensus mechanisms.

How far can the decentralization of Blockchain systems go? This work sets a solid conceptual framework for the understanding of the limits of distributed consensus mechanisms.

This study discloses two fundamental mutual exclusions in collective consensus systems: (a) in the requirement for concurrent optimization of *Autonomy* and *Consensus achievement* and (b) in the requirement for concurrent optimization of *Consensus achievement* and *entropic Performance.*

Through this work, an essential intrinsic limitation on the way towards distributed consensus systems' optimization is exposed. This proclaims the need for scalability in the consensus process: whether prioritizing the achievement of high consensus levels, autonomous operation, or high energy performance, this work lays the foundation for the modeling and development of adaptive mechanisms to dynamically balance these requirements in distributed consensus systems.

References

1. Aristoteles (284~322 B.C.) "Ηθικά Νικομάχεια" Book 1, p. VII. Available online: https://www.greek-language.gr/digitalResources/ancient_greek/library/ browse.html?text_id=78&page=8 Last (accessed on 10/5/2024).
2. Anagnostakis, A.G.; Naxakis, C.; Giannakeas, N.; Tsipouras, M.G.; Tzallas, A.T.; Glavas, E. Scalable Consensus over Finite Capacities in Multiagent IoT Ecosystems. *IEEE Internet Things J.* **2022**, 10, 6673–6688. https://doi.org/10.1109/JIOT.2022.3162103 .
3. Dwork, C.; Lynch, N.; Stockmeyer, L. Consensus in the Presence of Partial Synchrony. *J. ACM* **1988**, *35*, 288–323.
4. Alon, N.; Benjamini, I.; Lubetzky, E.; Sodin, S. Communication Cost of Consensus for Nodes with Limited Memory. Proc. Natl. Acad. Sci. USA 2019, 116, 1912980117. https://doi.org/10.1073/pnas.1912980117
5. Pereira, E.J.; Fonseca, E.S.; Mendonça, J.A.; Neto, J.B.; Pinto, A. Sustainable Consensus Algorithms Applied to Blockchain. Sustainability 2024, 16, 10552. https://doi.org/10.3390/su162310552
6. González, A.; Schülke, C.; Ehlers, M. Beyond Bitcoin: Evaluating Energy Consumption and Environmental Impact of Various Cryptocurrencies. Energies 2024, 16, 6610. https://doi.org/10.3390/en16186610

7. Liu, Y.; Chen, H.; Sun, X. Concerning Consensus Fairness and Efficiency with Uncertain Costs. Mathematics 2023, 12, 1266. https://doi.org/10.3390/math12081266
8. Anagnostakis, A.G.; Giannakeas, N.; Tsipouras, M.G.; Glavas, E.; Tzallas, A.T. IoT Micro-Blockchain Fundamentals. *Sensors* **2021**, *21*, 2784. https://doi.org/10.3390/s21082784.
9. Vogels, W. Eventually Consistent. *Commun. ACM* **2009**, *52*, 40–44. https://doi.org/10.1145/1435417.1435432.
10. Brewer, E.A. Towards Robust Distributed Systems. Proceedings of the Annual ACM Symposium on Principles of Distributed Computing (PODC), Portland, OR, USA, 16–19 July 2000. https://dl.acm.org/doi/10.1145/343477.343502
11. Gilbert, S.; Lynch, N. Brewer's Conjecture and the Feasibility of Consistent, Available, Partition-Tolerant Web Services. ACM SIGACT News 2002, 33, 51–59. https://doi.org/10.1145/564585.564601.
12. Rožman, N.; Corn, M.; Škulj, G.; Berlec, T.; Diaci, J.; Podržaj, P. Exploring the Effects of Blockchain Scalability Limitations on Performance and User Behavior in Blockchain-Based Shared Manufacturing Systems: An Experimental Approach. Appl. Sci. 2023, 13, 4251. https://doi.org/10.3390/app13074251
13. Bulgakov, A.L.; Aleshina, A.V.; Smirnov, S.D.; Demidov, A.D.; Milyutin, M.A.; Xin, Y. Scalability and Security in Blockchain Networks: Evaluation of Sharding Algorithms and Prospects for Decentralized Data Storage. Mathematics 2024, 12, 3860. https://doi.org/10.3390/math12233860
14. Anagnostakis, A.G.; Glavas, E. Entropy and Stability in Blockchain Consensus Dynamics. Information 2025, 16, 138. https://doi.org/10.3390/info16020138
15. Zhang, Y.; Xu, C.; Wang, X.; Yang, F. A Witness-Based Consensus Algorithm for Private Blockchains. IEEE Access 2019, 7, 38874–38883. https://doi.org/10.1109/ACCESS.2019.2906022
16. Nakamoto, S. Bitcoin: A Peer-to-Peer Electronic Cash System. Bitcoin.org. 2008. Available online: https://bitcoin.org/bitcoin.pdf (accessed on 15/5/2024).
17. Buterin, V. A Next-Generation Smart Contract and Decentralized Application Platform. Available online: https://blockchainlab.com/pdf/Ethereum_white_paper-a_next_generation_smart_contract_and_decentralized_application_platform-vitalik-buterin.pdf (accessed on 10/1/2025).
18. Hyperledger Fabric Documentation. Available online: https://hyperledger-fabric.readthedocs.io/ (accessed on 10/1/2025).
19. Arduino. Arduino ABX00027 Datasheet. Available online: https://docs.arduino.cc/resources/datasheets/ABX00027-datasheet.pdf (accessed on 1/2/2025).
20. Noether, E. Invariante Variationsprobleme. Nachrichten von der Gesellschaft der Wissenschaften zu Göttingen, Mathematisch-Physikalische Klasse 1918, 235–257. Available online: https://eudml.org/doc/59024 (accessed on 14/2/2025).
21. Noether, E. Invariant Variation Problems. Translated by M.A. Tavel. Transp. Theory Stat. Phys. 1971, 1, 186–207. Available online: https://arxiv.org/pdf/physics/0503066.pdf (accessed on 14/2/2025).