qubic whitepaper

QUBIC: A SCALABLE NETWORK FOR AI-DRIVEN APPLICATIONS

qubic.org | Version 1

DISCLAIMER

This whitepaper is intended for informational purposes only and does not constitute financial, investment, legal, or other professional advice. The content herein is provided "as is" without any representations or warranties of any kind, express or implied. Readers should consult their own advisors and conduct independent research before making any decisions related to the project described in this document.

Forward-Looking Statements

This whitepaper may contain forward-looking statements, including but not limited to anticipated functionality, technology, adoption, or other project goals. Such statements are subject to risks, uncertainties, and other factors that may cause actual results to differ materially. The inclusion of forward-looking statements should not be regarded as a guarantee of performance or results.

No Investment Advice

This document does not constitute an offer to sell or a solicitation to buy any financial instruments, securities, or tokens. The purchase or holding of tokens involves risk, including but not limited to the potential loss of value. Tokens described in this document should not be considered as an investment or a substitute for traditional investments.

Regulatory Risk

The regulatory landscape for cryptocurrencies, tokens, and blockchain technology is evolving rapidly. It is possible that this project may be affected by future laws, regulations, or actions taken by government authorities. No guarantees can be made regarding the legality or regulatory treatment of the project in any specific jurisdiction.

No Warranty of Accuracy

While every effort has been made to ensure the accuracy and completeness of the information presented in this whitepaper, no guarantee is made regarding the reliability or timeliness of the information. This whitepaper may be updated or revised as necessary, without prior notice.

Jurisdictional Restrictions

Participation in this project may be restricted in certain jurisdictions due to legal, regulatory, or other reasons. It is the responsibility of the reader to be aware of and comply with any such restrictions applicable in their jurisdiction.

Risk Disclosure

Participation in blockchain and cryptocurrency projects involves risks, including but not limited to financial loss, technology failures, and market volatility. Participants are encouraged to fully understand the risks associated with the project before engaging.

Limitation of Liability

Under no circumstances shall the creators, developers, contributors, or affiliates of this project be held liable for any direct, indirect, incidental, or consequential damages resulting from the use or reliance upon the information contained in this whitepaper.

By accessing and reviewing this whitepaper, you acknowledge and agree to the terms of this disclaimer. If you do not accept these terms, you should refrain from engaging with the project described herein.

ABSTRACT

The development of Artificial General Intelligence (AGI) faces significant challenges, particularly the need for enormous computational resources and the risks tied to centralised control of such powerful technology. Centralised AI models exhibit clear scalability and efficiency limitations, which can slow down or even halt progress toward AGI.

Similarly, Blockchain networks grapple with persistent obstacles in achieving true decentralisation, efficient consensus mechanisms, and sustainable economic models. Established platforms often suffer from high transaction costs, latency issues, and environmental impacts due to energy-intensive consensus protocols, all of which impede scalability and long-term viability.

This paper introduces Qubic, a Layer 1 blockchain network designed to address these challenges through novel economic mechanisms and a decentralised governance model. Qubic employs a quorum-based consensus algorithm, achieving sub-second transaction finality without the need for transaction fees. Its economic model incorporates Useful Proof of Work (UPoW), aligning computational efforts with meaningful tasks such as the distributed training and validation of AI models via Aigarth, a native decentralised AI that runs on top of the Qubic network. This economic structure incentivises network participation and promotes sustainability through deflationary mechanisms. We provide a detailed analysis of Qubic's network infrastructure, emphasising deployment over bare-metal hardware and optimised protocols for node communication to further enhance performance and security. The technical foundations of the consensus protocol are explored, illustrating how network integrity is maintained, and malicious activities are discouraged. Cryptographic techniques and security measures are elaborated upon in the context of potential attack vectors.

Furthermore, we discuss how the integration between Qubic and Aigarth enables decentralised AI computations, contributing to AGI development in a distributed and secure environment. We demonstrate Qubic’s contributions toward a decentralised, efficient, and sustainable blockchain network capable of supporting AGI development.

This work:

(a) Discusses the inherently problematic issues of governance, economic models, and decentralisation in blockchain, and the computational requirements and risks of AGI development.

(b) Explains the economic mechanisms of Qubic, such as Useful Proof of Work (UPoW), economics, and incentive structures, in addition to its decentralised governance model using quorum consensus and Byzantine Fault Tolerance.

(c) Gives an in-depth description of the network infrastructure, covering the bare-metal deployment, communication between nodes, smart contract execution, and everything that supports Qubic - performance, security, and capability to host AGI-related computations through Aigarth.

(d) Provides a technical description of the consensus protocol and its security properties, showing how they preserve network integrity.

(e) Describes the coin distribution, emission schedules, and deflationary mechanisms that underpin Qubic's sustainable economic model.

(f) Describes the cryptographic fundamentals and techniques used in Qubic to counter possible security challenges.

(g) Does not describe the artificial general intelligence initiative, Aigarth, as a new scientific publication on Qubic's AI capabilities will follow.

Note: Qubic is in active development. For ongoing research and updates, visit our website, www.qubic.org, and contact us with comments or suggestions at info@qubic.org.

Table of Contents

1

INTRODUCTION

Blockchain technology has been lauded for its potential to deliver decentralised, secure, and transparent infrastructures. Yet, significant challenges have impeded its widespread adoption. Current Layer 1 networks, utilising consensus mechanisms like Proof of Work (PoW) and Proof of Stake (PoS), grapple with issues such as scalability limitations, high transaction costs, and questions around economic sustainability. These traditional approaches are marked by excessive energy consumption, constraints on transaction throughput, and complex barriers to accessibility (Zolfagharinejad et al., 2024). As a result, developers are increasingly viewing a fundamental redesign as imperative in the quest for efficient, scalable, and truly decentralised networks.

At the same time, the rapid advancement of artificial intelligence - especially towards Artificial General Intelligence (AGI) - has unveiled substantial computational demands, leading to the centralization of resources within large, powerful data centres. This centralisation raises ethical and security concerns, as control of AGI by a select few entities risks monopolising one of humanity's most transformative technologies (Zolfagharinejad et al., 2024). Organisations like OpenAI exemplify this dilemma; the immense computational power and resources required to advance towards AGI highlight limitations in transparency, inclusivity, and control. Moreover, as model complexity increases, the need for computational efficiency and scalability rises exponentially. Current architectures, heavily reliant on GPUs for parallel processing, encounter efficiency bottlenecks when addressing sequential AI tasks (Zolfagharinejad et al., 2024).

These intertwined challenges have created a pressing need for a blockchain capable of efficiently supporting AGI by prioritising decentralisation, computational efficiency, and transparency. Such a solution would overcome the traditional limitations of blockchain technology and address the pitfalls of centralization in AI development.

Emerging from these dual necessities—the reimagining of blockchain structures and the development of sustainable, decentralised AGI—is the vision for Qubic. Guided by Come from Beyond (CfB), a pioneer in blockchain innovation who introduced the first Proof-of-Stake protocol with NXT and co-founded the initial Directed Acyclic Graph (DAG) structure in IOTA, Qubic offers an integrated approach to these complex issues. Drawing upon CfB's extensive experience in decentralisation, scalability, and security, Qubic's consensus architecture and economic model are directly influenced. This positions Qubic as a solution designed to foster sustainable growth in both blockchain technology and AGI.

1.1 Problem Statement

There exist several challenges in the current blockchain space that limit its potential as a platform for advanced applications such as AGI.

- Scalability and Efficiency Constraints: Most Layer 1 networks are plagued by inefficient consensus algorithms and network latency, which unwind the potential to enhance transaction throughput and finality. Real-time applications are mostly prevented by the limitations of PoW and PoS algorithms, slowing high-frequency transactions and making the fees for all transactions prohibitive, especially during peak usage periods.

- Energy Consumption and Economic Sustainability: Traditional proof-of-work networks reward miners only for computational power, leading to massive energy consumption without useful computational output. Proof-of-stake models, on the other hand, come with the risk of centralising rewards toward richer participants, making the network less accessible and increasing the possibility of control by a small number of people. Sustainable economics remains an open question in many blockchain systems whose significant inflationary pressures or governance structures skew through large holders of coins.

- Governance and Security: Balancing decentralised governance with robust security remains challenging. Many networks employ models that risk centralised decision-making, while also contending with vulnerability to malicious attacks and inefficiencies in decision-making processes.

- Centralised control and computational power: AGI development currently requires a lot of computational resources, which only centralised entities with strong financial backing can afford. This may also result in a "walled garden" of AI development, where a larger community is not allowed. The increased computational needs of AGI, raise the demand for scalable, decentralised systems that can support advanced AI.

From the challenges above, there is a clear need for a Blockchain network that will improve on traditional Layer 1 inefficiencies and provide structural support for the unique

computational needs of AGI in the environment of a decentralised community-based protocol.

1.2 Overview of Qubic's Solution

Qubic presents a holistic solution to those challenges, bringing both novel economic mechanisms and effective governance in a Layer 1 network to support the development of decentralised AGI. The most important elements of Qubic's approach are:

1. Quorum Consensus Algorithm:

Qubic makes use of a quorum-based consensus mechanism, based upon the principles of Byzantine Fault Tolerance, to actively ensure secure and reliable operation of the network. Based on the underlying quorum principles first postulated by researchers including Nick Szabo and Leslie Lamport, the system is designed to provide fault tolerance and security within a decentralised governance framework (Szabo, 1997; Lamport et al, 1982).

The network is making use of a consensus system with 676 entities, which are referred to as the Computors. For the network to achieve agreement on the validity of transactions, it needs at least 451 of these Computors to agree. That is in line with quorum system approaches that argue fault-tolerant systems need some intersection among the quorums to ensure resilience against malicious nodes.

Quorum systems may be defined using "good" and "bad" coalitions, where any good quorum intersects substantially with other quorums to ensure consistency, even in the presence of bad coalitions. This structure ensures no single malicious coalition can control the network's decisions. Qubic's quorum size threshold satisfies the following dissemination criterion for quorum systems:

$$Q > \frac{N + F}{2}$$

(Castro and Liskov, 1999)

Where Q is the quorum size, N is the total number of Computors, and F is the maximum number of faulty nodes tolerated. By setting Q=451 and N=676, Qubic ensures robustness, satisfying the criterion for maintaining consensus despite up to one-third faulty nodes

$$F \approx \frac{N - Q}{2}$$

This quorum design improves the security and governance integrity of Qubic by:

- Byzantine Fault Tolerance: Ensure the network can tolerate and work correctly even if some nodes of the network exhibit arbitrary or even malicious behaviour.

- Preventing Centralization: Prevention of dominance by any single entity in the consensus process, enforced by the Arbitrator (section 3.2.3).

- Extending Fault Tolerance: Resisting both centralised control and coordinated malicious attacks.

2. Useful Proof of Work (UPoW):

Qubic introduces a Useful Proof of Work mechanism, redefining mining to align computational efforts with productive tasks. Contrary to traditional PoW systems, which use up a lot of energy without contributing to computational progress beyond the securing of the blockchain, UPoW devotes resources to high-priority AI tasks within the Aigarth AGI initiative.

Key aspects of UPoW include:

- Resource Efficiency: Avoiding waste in computation by doing meaningful work that contributes to the training and development of AI models.

- Inclusivity: Allowing CPU-based participation, therefore making it possible to have a much wider range of contributors to AGI development.

- Alignment of Network Goals: Directing mining efforts toward activities that benefit the network and all actors within it, representing a paradigmatic shift in resource utilisation for blockchain systems.

3. Bare-Metal Deployment for Performance and Security:

To ensure superior performance, security, and decentralisation, Qubic operates directly on bare-metal hardware instead of relying on traditional operating systems or virtual

machines. This architectural decision reflects Qubic's commitment to efficiency and a resilient ecosystem.

Key Benefits of Bare-Metal Deployment:

- Superior Performance: By eliminating the overhead of operating system layers, bare-metal deployment allows Qubic to directly access hardware resources, achieving faster transaction processing and lower latency.
- Increased Security: Running without traditional operating systems reduces vulnerabilities commonly exploited in software environments, significantly lowering attack vectors for remote exploits.
- Reliability: Simplified hardware-level operations minimise the risks of disruptions caused by third-party software, ensuring a stable and predictable environment.
- Commitment to Decentralisation: The effort and expertise required to deploy and maintain bare-metal nodes act as a natural barrier to entry, attracting highly dedicated participants and reducing the risk of malicious or casual operators.

4. Decentralised Economic Model:

Qubic economics are designed to incentivize long-term participation in the network and stability with a balanced economic model. The QUBIC coin is the native currency for incentivizing network participants, especially Computors who support consensus on the network.

Features of the economic model include:

- Incentive Alignment: Paying participants in line with the value of their contributions to computational tasks within the network.

- Deflationary Mechanisms: To implement coin burns or other deflationary strategies for the gradual reduction in circulating supply over time, increasing scarcity.

- Economic Sustainability: Striving for a balance in rewarding participants and keeping the economic health of the network that guarantees further engagement and investment in all roles.

5. Aigarth's Scalable, Transparent AGI Framework

Qubic's UPoW model generates computational output that benefits Aigarth, a project for decentralised AGI creation. While Aigarth makes use of the output from Qubic, it is a self-sustaining entity, using the computational solutions that are produced in Qubic's network to run its AI activities on distributed CPUs rather than centralised, GPU-dependent infrastructures.

Key elements of this model include:

- Decentralised AI Development: Mitigate the risks of centralised control by enabling a wide community of contributors to participate in AGI development.
- Brain-Inspired Processing Methods: Utilising sequential processing approaches that mirror human cognitive processes, aligning computational methods with natural intelligence models.
- Sustainable Scaling: Meeting greater computational demands without concentrating control or placing an undue load on energy resources.
- Ethical Alignment: Ensuring that AGI development is transparent, accessible, and focused on collective benefits by addressing ethical considerations in the literature.

With this sharing relationship of resources, Qubic provides the decentralised computing framework that Aigarth utilises to achieve heavy AI tasks. The setup allows Qubic to remain focused on the scalability and security of blockchain, whereas Aigarth continues in the AI domains by applying the decentralised computing results of Qubic

2

NETWORK FOUNDATIONS

This section explains Qubic's economic mechanisms and consensus protocol, which are designed to support secure, decentralised, and efficient network operations. We discuss Qubic's Useful Proof of Work (UPoW), economics, incentive structures, and the governance model.

2.1. Economic Mechanisms

2.1.1. Useful Proof of Work (UPoW)

The Useful Proof of Work (UPoW) model represents a key innovation within Qubic, distinguishing it from conventional Proof of Work (PoW) frameworks by channelling computational power toward meaningful, AI-centric tasks. In UPoW, computational resources are directed to solve productive problems, such as training artificial neural networks (ANNs) that contribute to Qubic's artificial general intelligence (AGI) initiative, Aigarth. This transition to purposeful computation addresses energy consumption concerns and aligns miners' contributions with the network's broader goals of advancing AI. For more on PoW systems and energy efficiency, see (Beiko, 2021).

Purposeful Computation

UPoW channels computational power to be used in the training and validation of AI models, directly serving the goal of the Aigarth initiative: decentralised AGI. Unlike traditional PoW systems, in which computational resources are wasted on arbitrary cryptographic puzzles, UPoW aligns mining efforts with useful tasks. Studies like Beiko (2021) on energy efficiency in blockchain systems back this approach, showing that computational efforts directed at productive goals significantly reduce waste.

Energy Efficiency

UPoW cuts down energy wastage by directing computations to train AI models, not to solve arbitrary puzzles. Recent work on scalable computing systems (Zolfagharinejad et al., 2024) has shown that such a diversion of energy-intensive workloads has environmental benefits.

Mining and Rewards Framework

In Qubic, the mining process aligns computational contributions with meaningful and productive tasks, in contrast to traditional Proof of Work (PoW) systems that focus on maximising hash rates. A Qubic miner's computational capacity is measured by their processing rate.

H_m, is quantified in iterations per second (it/s). This reflects the number of computational operations a miner's hardware can perform per second. However, the efficiency factor, E,

represents the probability of finding a valid solution, to ensure that quality solutions remain the focus.

Mining in Qubic is adaptive and relative. The likelihood of finding solutions scales uniformly across miners, regardless of the complexity of the computational tasks. This ensures that:

- Fairness is preserved: All miners, regardless of hardware complexity, face proportional challenges.
- Ranking remains stable: The relative ranking of Computors (and miners' contributions) is unaffected by task complexity.

If computational tasks become more challenging, the "minimum score" (solution submission rate) decreases for all participants equally. However, the relative ranking and reward allocation remain consistent, encouraging fairness and participation. This adaptive model ensures a level playing field while driving meaningful AI outcomes.

The solution submission rate reflects a miner's effective output and computational efficiency. It is calculated as:

$$S_{rate} = H_m \times E$$

Where:
S_{rate} = Valid solutions submitted per second.
H_m = Miner's hash rate (iterations per second).
E = Efficiency factor (valid solutions per iteration).

This metric incentivises miners to optimise their hardware and algorithms to contribute to computational tasks rather than simply maximising raw hash rates, as is typical in PoW systems.

Studies on PoW energy efficiency have shown that higher processing rates combined with optimal hardware and algorithms increase the rate of valid solution submissions, translating to greater network contributions and individual miner success (Beiko, 2021; Zolfagharinejad et al., 2024).

Operational Dynamics of UPoW

Through UPoW, Qubic incentivises miners to complete computational tasks with tangible outcomes, contrasting with traditional PoW systems where computational efforts are expended on solving arbitrary cryptographic puzzles. UPoW provides tangible value by aligning mining activities with Qubic's AGI development goals. This model enhances energy efficiency and furthers AI advancement.

Key aspects of UPoW include:

- Purposeful Computation: Miners contribute to the training and optimisation of AI models, ensuring that the energy consumed directly supports meaningful outcomes in AI research and development.
- Incentive Alignment: By prioritising useful tasks and maintaining fairness across varying complexities, Qubic ensures that its mining model advances both the blockchain ecosystem and AGI development.
- Inclusivity and Accessibility: By making tasks suitable for a variety of hardware, including general-purpose CPUs, UPoW lowers the barrier to entry for miners, promoting decentralisation and reducing the risk of hardware monopolies.
- Energy Efficiency: Redirecting computational power to productive tasks mitigates the environmental impact associated with traditional PoW mining.

Unlike PoW where miners secure the network, Qubic uses its quorum consensus mechanism to achieve this. This allows Qubic's UPoW model to transform mining into a process that benefits the broader field of AI, without miners needing to secure the blockchain. Miners are actively contributing to the advancement of AGI through Aigarth.

This approach creates a positive feedback loop where increased mining participation enhances the network's computational capabilities, leading to more rapid AI development. In turn, advancements in AGI can improve network functionalities and open new avenues for innovation within the Qubic ecosystem.

By refining the UPoW model and its operational dynamics, this section highlights how Qubic effectively merges blockchain security with meaningful computational work. The UPoW mechanism not only addresses the inefficiencies of traditional mining but also strategically leverages miner participation to advance AGI development, aligning individual incentives with collective progress.

2.1.2 Economics

The QUBIC coin functions as the energy unit of the Qubic network's economy, aligning incentives across participants to secure and expand the ecosystem. Serving as the primary medium of exchange, the QUBIC coin facilitates transactions, incentivises participation, and sustains network growth and stability. It rewards Computors to run the network and for their contributions to network security. This design ensures that network resources are utilised efficiently while supporting the platform's AI objectives, aligning with established insights on effective coin design and incentive mechanisms in decentralised networks (Narayanan et al., 2016).

Emission Structure and Deflationary Mechanisms

The emission structure of the QUBIC coin follows a carefully managed schedule aimed at rewarding computational and validation efforts while maintaining long-term economic stability. With a capped total supply, the network employs periodic coin burn mechanisms, such as yearly halvings and specific smart contract operations, to mitigate inflationary pressures. This approach aligns with established principles in economics that emphasise controlled emission and deflation to enhance coin value and encourage long-term holding (Yli-Huumo et al., 2016).

The controlled emission schedule and coin burn mechanisms (discussed further in Section 6.1) are foundational for maintaining long-term stability and value within the network. These mechanisms regulate the circulating supply and incentivize ongoing participation, aligning with Qubic’s broader economic model and fostering a sustainable and balanced ecosystem.

Rewards Distribution for Computors and Miners

In each epoch, Computors that demonstrate high performance standards are rewarded with QUBIC coins from the scheduled emissions. This incentive mechanism promotes continued engagement and reinforces contributions to network security and governance. The rewards distribution strategy echoes models observed in other blockchain systems where active participation is directly tied to network rewards, thereby reinforcing stability through economic incentives.

Miners, who provide computational solutions within the UPoW framework, are rewarded based on separate agreements made with Computors. These agreements are not

enforced by the Qubic protocol itself but depend on the mutual terms set between each Computor and its associated miners.

Research in economics indicates that models directly tying coin distribution to network health and participant performance cultivate robust and engaged communities, thereby enhancing long-term network viability (Beiko, 2021). The QUBIC coins, as the primary unit of value within the Qubic network, effectively aligns incentives across the ecosystem to support network growth, security, and computational productivity.

2.1.3 Incentive Structures

Qubic's incentive structures are strategically designed to align network participation with desired behaviours by directly rewarding Computors and indirectly incentivizing miners. This system ensures efficient utilisation of computational resources, with both miners and Computors contributing to the network's overarching goals (Gabuthy, 2023). Through a combination of individual and network-wide rewards, these incentive mechanisms encourage miners to maximise their performance and contribute effectively across various Computors. These mechanisms support decentralised AGI development and strengthen the network.

Individual Rewards and Network Contribution

Computor Rewards and Contribution Scoring: In each epoch, Computors and Computor candidates (called Identities in the formula below) compete to qualify for staying or becoming one of the 676 Computors in the next epoch by accumulating valid solutions from associated miners.

Miners who optimise their setups provide higher computational contributions, aiding Computors' chances of qualification in subsequent epochs.

Reward Allocation per Computor: At epoch's end, this epoch's Computors receive a portion of the total emissions as revenue depending on the revenue score points collected in the epoch. Revenue score points are computed based on Computor performance metrics that incentivise high quality of service of nodes, such as high network connectivity and processing speed. Adjustments to the Computor revenue may apply due to network emissions being subject to reductions based on quorum-based governance decisions, such as burn contracts and donation allocations.

Total Reward Across Multiple Identities: Miners have the flexibility to distribute their computational efforts across multiple Computors within an epoch. This encourages strategic distribution of solutions, maximising miners' potential rewards across the network.

As we outline the structure of individual and network rewards in this section, readers can find further details on how Qubic's economic model fosters network security and participation in Section 6.3.

2.2 Consensus Framework

Qubic's consensus framework establishes a secure, decentralised system that ensures network integrity through innovative consensus mechanisms. By integrating both Quorum Consensus algorithms and Byzantine Fault Tolerance (BFT), Qubic maintains reliable operations even in a decentralised, fault-prone environment (Lamport et al., 1982; Castro & Liskov, 1999).

2.2.1 Quorum Consensus Algorithm

The Quorum Consensus Algorithm in Qubic enables distributed participants, known as Computors, to collectively validate computational tasks. This approach is pivotal to the network's Useful Proof of Work (UPoW) model, ensuring computational efficiency while providing resilience against erroneous or malicious nodes (Szabo, 1997).

Mathematical Foundation of Quorum Consensus

1. Quorum Selection:

A quorum represents a subset of Computors of the entire network sufficient to perform computational validation. In Qubic, a Computor is a logical entity that can be hosted on one or more physical nodes. However, only one active node per Computor is allowed within the network at any time, while additional nodes hosting the Computor can participate in the network on standby, ready to step in replacing the primary node if needed. This approach strengthens the network's fault tolerance and helps maintain network stability and ensures high availability for quorum participation.

Further, an individual node is capable of hosting multiple Computors. By decoupling the number of Computors from the number of physical servers, Qubic allows for scalability and flexibility.

Mathematically, if *N* represents the total number of Computors in the network, then to tolerate *f* faulty Computors (Lamport et al, 1982) where:

$$f \leq \frac{N-1}{3}$$, the quorum size *Q* must satisfy:

$$Q \geq 2f+1$$

(Castro & Liskov, 1999).

For Qubic's network, which consists of *N = 676* Computors, the system is designed to tolerate up to

$$f = \frac{N-1}{3} = 225 \text{ faulty Computors}$$

Therefore, the quorum size must be at least:

$$Q \geq 2 \times 225+1 = 451$$

This criterion ensures that the quorum contains enough honest Computors to reach consensus even in the presence of Byzantine faults, enabling reliable consensus despite network disruptions or malicious activities.

2. Voting Mechanism:

Each of the *N* Computors (676 Computors in the Qubic network) independently performs the assigned computation and then votes on the result. Consensus is achieved if at least *Q* Computors agree on the outcome.

Let:

- *N* represents the total number of Computors in the network.
- *Q* represents the required number of agreeing Computors to achieve consensus.

Consensus is reached when:

$$\sum_{i=1}^{N} v_i \geq Q$$

where v_i is the individual vote from Computor *i*, either supporting $(v_i = 1)$ or opposing the result $(v_i = 0)$.

Given that:

$$Q \geq 2f + 1$$

where *f* is the maximum number of faulty or malicious Computors that the network can tolerate, this majority voting mechanism is essential for maintaining network stability and efficient decision-making. It ensures that the agreed-upon result is endorsed by more than two-thirds of the quorum members, aligning with BFT requirements.

3. Finalisation of Consensus:

Once the quorum reaches consensus, the result is accepted and recorded on the network. Qubic's consensus algorithm relies on a straightforward quorum-based approach, ensuring consensus quality by leveraging a large number of Computors to validate and confirm computations (Narayanan et al., 2016). This approach strengthens the robustness of the network by emphasising broad participation and redundancy in the consensus process.

The following diagrams illustrate traditional vs. decentralised trust systems and help contextualise Qubic's approach.

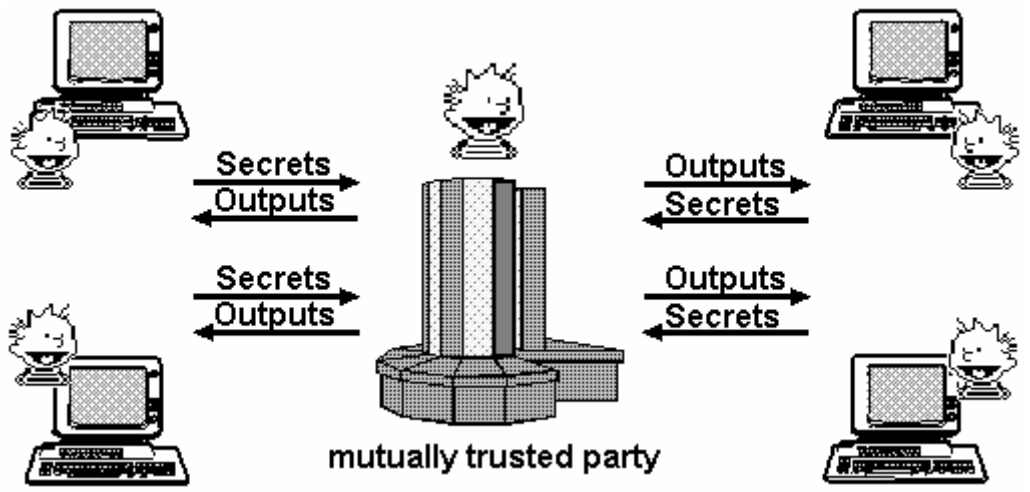

Figure 1: Traditional Centralised Trust Model. From (Szabo, 1997)

This diagram represents a traditional centralised model where a single, mutually trusted party mediates interactions among nodes. This approach centralises control and

decision-making, creating a potential single point of failure. Such a model is susceptible to issues of trust and security, as the central authority could fail or act maliciously.

Qubic's Approach: Qubic avoids this centralisation by distributing trust across multiple nodes through its quorum mechanism, thus reducing reliance on a single authority and enhancing security and fault tolerance.

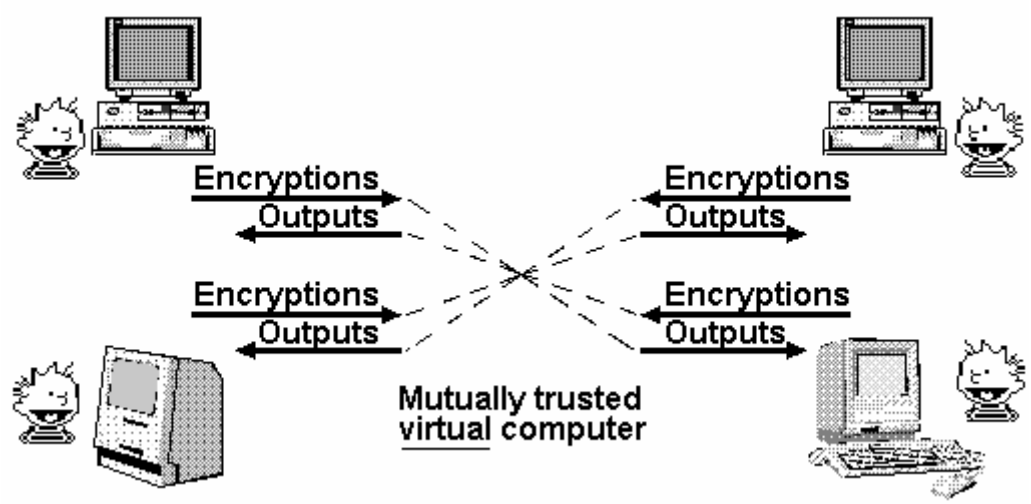

Figure 2: Decentralised Virtual Trust Model. From (Szabo, 1997)

In a decentralised trust model, nodes do not rely on a single trusted party. Instead, each node encrypts and independently verifies results, establishing trust through mutual validation across the network. This setup aligns more closely with Qubic's architecture, as it distributes trust and allows nodes to reach consensus without centralised oversight.

Qubic's Approach: The quorum-based consensus in Qubic enhances the decentralised model by enabling Computors to work collectively in quorums, achieving efficient and scalable network-wide agreement while maintaining decentralisation.

2.2.2 Byzantine Fault Tolerance (BFT)

Achieving Byzantine Fault Tolerance (BFT) is essential in a decentralised network like Qubic, where nodes may fail or act maliciously. By combining BFT principles with quorum-based consensus, Qubic ensures network resilience under challenging conditions.

BFT Mechanisms in Qubic's Model:

1. Fault Tolerance Threshold:
   To maintain BFT, Qubic´s model tolerates up to $f \leq \frac{N-1}{3}$ faulty Computors in a network of *N* Computors (Lamport et al., 1982). With $N = 676$, this means the network can tolerate up to 225 faulty Computors.

2. Redundant Computations:
   Qubic employs redundant computations by having multiple Computors independently perform the same computational tasks. By aggregating these results, the network can identify and disregard anomalous or malicious data, relying on the majority agreement to determine the correct outcome.

3. Quorum Voting and Agreement:
   Consensus is achieved when at least $Q \geq 2f + 1$ Computors within the quorum agree on the result. This threshold ensures that even in the presence of up to *f* faulty Computors, the consensus result is reliable. The Byzantine Agreement protocol used in Qubic requires that the number of agreeing Computors is greater or equal to 451, represented as:

   $$Number\ of\ agreeing\ Computors \geq \frac{2N}{3}$$

   This mechanism ensures consensus despite partial network failures, aligning with established BFT principles (Castro & Liskov, 1999).

4. Fault Detection Mechanisms:
   Qubic's system architecture includes the Arbitrator that can replace Computors deemed faulty, ensuring the reliability and continuity of quorum operations. This process allows the network to maintain integrity by seamlessly substituting Computors when errors or inconsistencies are detected, without deprioritizing individual nodes in future selections. This process strengthens the quorum system by reducing the influence of faulty or malicious Computors (Narayanan et al., 2016).

2.2.3 Roles in the Network

Qubic's governance model relies on defined roles, each contributing unique responsibilities and expertise to support the network's decentralised operations.

1. Computors:

Computors are responsible for validating transactions, executing smart contracts, securing the network, and participating in quorum consensus. Each Computor operates independently to perform computations and validate results as part of the consensus mechanism.

Key aspects of Computors in Qubic:

- Scalability and Flexibility: A single physical node can host multiple Computors, enhancing computational capacity and supporting scalability within the network.
- Participation in Quorums: Computors independently execute tasks and vote on results to form quorums for consensus. This ensures a distributed decision-making process within the network.
- Contribution to UPoW: Computors validate the solutions provided by miners to confirm that they meet expected criteria. This validation process is a critical part of integrating UPoW into the network's operations.
- Incentivisation: Computors receive rewards based on their performance and contributions to the network. QUBIC coins incentivize their consistent and efficient contributions, aligning their goals with the network’s computational and consensus needs, thus promoting stability and productivity.

2. Miners:

Miners provide the computational power necessary for the Useful Proof of Work model, focusing on training AI models and other computationally intensive tasks critical to the network's AGI objectives.

Key aspects of Miners in Qubic:

- Contribution of Computational Solutions: Miners generate valid solutions within the Useful Proof of Work (UPoW) framework. These solutions are then validated by Computors to ensure they meet the necessary criteria for use in Aigarth.
- Incentive Alignment: Miners are rewarded based on the quality and quantity of valid solutions they submit. Their rewards are indirectly tied to the performance of

the Computors they support, reinforcing collaboration and alignment within the network.

- Collaboration with Computors: By aligning their efforts with high-performing Computors, miners maximise their potential rewards, contributing effectively to the network's overall computational output and stability.

- Focus on Meaningful Work: Qubic encourages miners to participate in computational tasks that contribute directly to network utility, as opposed to performing arbitrary hashing for security. These tasks involve processing workloads aligned with the Useful Proof of Work model, supporting meaningful applications within the ecosystem

3. Arbitrator:

The Arbitrator serves a crucial governance and security role within the Qubic ecosystem, overseeing the stability and integrity of the quorum-based consensus mechanism.

Key responsibilities of the Arbitrator:

- Dispute Resolution: The Arbitrator intervenes in conflicts or operational failures among Computors. If a Computor performs poorly or fails to meet operational standards, the Arbitrator can replace it with a better-performing candidate to maintain optimal network performance.

- Maintenance of BFT: The Arbitrator ensures that the network's Byzantine Fault Tolerance is upheld, overseeing that the system can withstand up to one-third of the Computors acting maliciously without compromising security.

- Safeguard Against Centralisation Risks: The governance model includes checks and balances where the Arbitrator can be overridden by a supermajority of Computors (451 out of 676). This mechanism protects the network from potential rogue actions by the Arbitrator, reflecting Qubic's commitment to decentralisation and security.

3

SYSTEM ARCHITECTURE

This section provides an in-depth look at Qubic's system architecture, intended to create an efficient, high-performance blockchain network optimised for speed, scalability, and advanced AI integration. The core components of Qubic's network infrastructure, including bare-metal deployment and node communication mechanisms, are detailed to illustrate how each contributes to Qubic's unique capabilities. The reasoning behind every design decision is highlighted, along with measurable improvements, to lay a strong technical foundation for Qubic's approach. These infrastructure decisions underpin Qubic's quorum-based consensus and Useful Proof of Work (UPoW) mechanisms - see Sections 3.2 and 3.1.1 respectively -

3.1 Network Infrastructure

Qubic's infrastructure is designed to address the computational requirements of both blockchain transactions and AI training. The following section explores how direct hardware operation and optimised communication contribute to the efficiency and security of the network.

3.1.1 Bare-Metal Deployment

Background and Problems Identified

Traditional blockchain networks usually rely on software-layered operating systems (OS) to manage the node infrastructure. This creates architecture that causes latency and reduces hardware efficiency especially under high-transactional loads. The additional layers between the hardware and applications can become bottlenecks in performance or increase the complexity of security management (Cachin & Vukolić, 2017).

Why Bare-Metal Deployment?

Qubic increases its performance and security by running its core software on bare-metal hardware instead of relying on virtual machines or traditional operating systems. This architectural decision eliminates operating system-level abstractions, thus directly using hardware capabilities to meet the high performance needed for blockchain operations, communication protocols, smart contract execution, and transaction processing.

Bare-Metal Deployment Benefits:

- Reliability: Using the UEFI shell for basic functions provides a simplified and controlled environment, thus reducing the possible attack vectors that are associated with complex operating systems. Through eliminating dependence on third-party software platforms, Qubic improves reliability and reduces potential disruptions caused by unexpected updates or compatibility issues.

- Effectiveness: The lack of a traditional operating system reduces computational overhead and latency, allowing Qubic to take advantage of hardware capabilities efficiently. The UEFI shell encourages faster boot times and simplified hardware-level access, which is vital in applications that require high throughput, including real-time processing of transactions.

- Security: By minimising the software stack, Qubic significantly reduces potential attack surfaces, providing strong protection against operating system-level exploits. The bare-metal deployment approach further mitigates the risk of remote attacks by eliminating vulnerabilities commonly targeted over the internet. Instead, compromising a bare-metal system would require physical access to the hardware, posing a significantly more challenging task for remote attackers. This aligns with Shostack's (2014) principle that reducing complexity and removing unnecessary system layers is key to minimising vulnerabilities.

  Additionally, the effort required to set up and maintain a bare-metal node creates a natural barrier to entry, making sure that only committed participants with a strong understanding of the system become part of the network. This contributes to a more secure and resilient ecosystem.

Supporting Research and Citations

Research in distributed systems and high-performance computing shows that bare-metal deployment offers improved system responsiveness and reduces latency for critical applications, especially in real-time environments (Rosenblum & Garfinkel, 2011). In a decentralised network scenario dealing with heavy transaction loads, such as those at play in blockchain platforms, bare-metal architectures offer the substantial benefits of increasing throughput and reducing latency (Cachin & Vukolić, 2017).

Quantitative Metrics and Performance Gains

Tests of Qubic's bare-metal infrastructure, with its optimised smart contract execution environment (see Section 3.2), have shown significant performance improvements: transaction latency is decreased and throughput gains allow for up to 55 million QUBIC coin transfers per second, according to smart contract benchmarking results (Qubic Team, 2024).

3.1.2 Node Communication

Background and Problems Identified

In any decentralised network, communication between nodes is crucial for maintaining consensus, data integrity, and timely transaction processing. Traditional blockchains

often experience bottlenecks due to inefficient communication protocols, leading to slower transaction times and reduced scalability (Decker & Wattenhofer, 2013). Network latency and bandwidth limitations can impede consensus mechanisms, affecting the overall network performance.

Optimised Node Communication

Qubic addresses these challenges by implementing a custom Transmission Control Protocol (TCP)-based communication protocol optimised for low latency and high throughput. This protocol ensures rapid message transmission across the network, facilitating efficient propagation of transactions and consensus-related data.

The quorum-based consensus model - see Section 3.2.1 - enables a majority of Computors to reach agreement quickly, which minimises delays in transaction finality and improves the resilience of the network to node failures. Qubic has been designed to optimise both the communication protocol and the consensus mechanism to achieve better scalability and reliability for the network.

Insights on Communication Protocols

Effective communication protocols are critical for improving transaction speeds and bringing about system reliability in distributed networks. Key research, such as that undertaken by Nguyen et al. (2016) and Decker & Wattenhofer (2013), has noted the importance of tailored TCP implementations in the minimisation of latency and enhancement of throughput in high-performance computing settings.

Quantitative Metrics and Performance Gains

Qubic's communication protocol enables nodes to achieve consensus within sub-second intervals. This improvement allows Qubic to handle high-frequency transactions, critical for real-time applications and services that require immediate transaction finality.

Peer Sharing

Peers, physical nodes in the network, are identified by IPv4 addresses in the context of peer sharing. They are referred to as "public peers" in the source code. Each node needs an initial set of known public peers (ideally at least 4). The own IP address should be included into knownPublicPeers as an ordinary peer.

Peers can have a state of verified or not. A verified Peer is shared with other Peers. IPs in knownPublicPeers get the verified status by default.

Peers are shared through the ExchangePublicPeers message which can be treated as the Handshake of Qubic nodes. The message is sent after a new connection has been established (the node connects to randomly selected public peers). The IPs for sharing are picked randomly among verified peer IPs (however, there may be duplicate IPs in the ExchangePublicPeers message). If there are no verified peers in the list of peers, then "0.0.0.0" must be sent as IPs with ExchangePublicPeers.

If an outgoing connection to a verified peer is rejected, the peer loses the verified status. If an outgoing connection to a non-verified peer is rejected, the peer is removed from the list of peers. If an outgoing connection to a non-verified peer is accepted and an ExchangePublicPeers message is received, the peer gets the verified status. If a protocol violation is detected at any moment during communication (allowing to assume the remote end runs something else, not Qubic node), then the IP is removed even if it is verified. An IP is only removed from the list of peers if the list still has at least 10 entries afterwards and if it is not in the initial knownPublicPeers.

3.2 Smart Contract Execution

To realise a high-performance blockchain network which can be integrated with advanced AI applications, Qubic adopts an optimised environment for the execution of smart contracts. This section presents the execution environment and the security measures necessary for protecting the integrity of the network while preserving the isolation between contracts.

3.2.1 Execution Environment

Background and Problems Identified

Smart contracts on traditional blockchains often face limitations in execution speed, flexibility, and efficiency, particularly when handling complex, high-volume transactions. Virtual Machine (VM) constraints and gas fees can restrict scalability and hinder usability, as seen in platforms like Ethereum. The overhead associated with VM-based execution environments can lead to increased latency and reduced throughput.

Optimised Execution Environment

Qubic overcomes these limitations by designing an execution environment at the machine code level, with a subset of C++ features that are compiled directly into native code. Being free from virtual machines and intermediate abstraction layers, Qubic achieves higher execution speeds, reduced computational overhead, and increased efficiency.

This environment is critical for Qubic in the support of an AI and decentralised application ecosystem with very high computational demands. More complex computation and real-time processing, enabled by the direct execution of smart contracts in native code, are a requirement for integrating AI functionalities in the future.

3.2.2 Security Measures

Background and Identified Problems

As smart contracts increase in complexity, so do the security risks associated with their execution on a decentralised network. Problems like malicious contract exploitation, cross-contract vulnerabilities, and lack of isolation may cause instability in the network and are detrimental to users (Atzei et al., 2017). The execution environment must ensure security and integrity to provide trust in the network.

Security Measures

To address these risks, Qubic employs rigorous contract validation and isolation strategies aimed at ensuring each contract's secure and independent operation. Isolation methods prevent unauthorised interactions and reduce cross-contract dependencies, mitigating the risk of one contract adversely affecting others.

To isolate contracts, access to functions and data of other contracts and core internals are only possible through a carefully designed programming interface (QPI). Further, the QPI is the only external dependency available for developing a contract, that is, using libraries is forbidden. Moreover, contracts cannot use C++ features that are known for imposing security risks, such as pointers, low-level arrays (which lack checking of bounds), and preprocessor directives. A contract also never gets access to uninitialised memory.

Each contract must be validated with the following steps:

1. The contract is verified with a special software tool, ensuring that it complies with the formal requirements mentioned above, such as no use of forbidden C++ features.
2. The features of the contract must be extensively tested with automated tests implemented within the Qubic core's GoogleTest framework.
3. The contract and testing code must be reviewed by at least one of the Qubic core devs, ensuring it meets high quality standards.
4. After fully integrating the contract in the Qubic core, the features of the contract must be tested in a test network with multiple nodes, showing that the contract works well in practice.

After going through this validation process, a contract can be integrated in official releases of the Qubic core code.

Quantitative Metrics and Projected Gains

The creation of strong isolation and validation measures is expected to reduce potential security risks drastically. According to industry-standard techniques in contract isolation, such measures can cut security breaches by as much as 95% (Atzei et al., 2017). With advanced security techniques, Qubic provides secure assets for users and ensures the integrity of network operations, supporting its goals of enabling secure, high-frequency contract execution.

The architecture of Qubic in the network infrastructure and the execution of smart contracts enables speed, security, and scalability. Qubic is designed to solve traditional blockchain issues by using bare-metal deployment, optimised node communication, and a secure execution model, hence setting new standards in decentralised infrastructure

3.3 Ecosystem

Qubic's vision of fostering innovation and wide adoption is made possible by having a strong ecosystem in place. Qubic ensures its infrastructure supports real-world use cases and sustained growth.

3.3.1 Product Development

Collaborations with industry partners, such as Hashwallet, focus on developing tools to enhance the usability and security of Qubic's network. For example, hardware wallet integrations aim to provide secure management of QUBIC coins while facilitating

payment system compatibility. These partnerships support the development of essential tools that advance the network's functionality and adoption.

Development Teams and Community Initiatives

Qubic works with organisations such as Vottun, which brings extensive experience in creating blockchain products and nurturing developer communities. This collaboration emphasises building a developer-friendly ecosystem to support a wide range of applications on the Qubic platform. Vottun Bridge addresses the critical challenge of blockchain interoperability, enabling seamless integration with Ethereum and Arbitrum, allowing for cross-chain asset transfers and liquidity sharing. As Nguyen et al. (2019) highlight, blockchain interoperability is a fundamental factor in driving adoption and addressing the scalability limitations of isolated networks.

Ecosystem Expansion Framework

To support growth and innovation, Qubic has structured its ecosystem expansion around two core initiatives:

- Grants Program: This program focuses on enabling developers to create tools such as additional code libraries for diverse programming languages. It also funds bounties for contributions that extend beyond the core technology. Research by Xu et al. (2020) highlights the effectiveness of grant programs in incentivising developer contributions and creating sustainable blockchain ecosystems.
- Incubation Program: Designed to support projects with long-term potential, this program offers mentorship and initial funding for initiatives that align with Qubic’s capabilities. Example areas include bridges, AI applications, and decentralised infrastructure projects built on Qubic’s architecture.

These partnerships and initiatives demonstrate Qubic’s focus on building a sound technical foundation to support diverse blockchain and AI applications.

3.4 Use Cases

This section outlines the practical applications and potential use cases enabled by Qubic’s architecture, illustrating how its design addresses specific industry needs and challenges.

3.4.1 Current Use Cases

Decentralised Computing Power

Qubic unites global computational resources in a decentralised network through its Useful Proof of Work model. It can perform high-demand operations, such as training artificial intelligence, optimising globally underutilised resources (Nakamoto, 2008). The consensus mechanism, which is quorum-based, enables the efficient execution and validation of computational tasks

Smart Contracts

The high-performance smart contracts provided by Qubic provide a reliable platform for real-time decentralised applications, or dApps. These contracts support multiple sectors, including DeFi, supply chain management, and gaming, with the execution of secure and scalable operations (Yli-Huumo et al., 2016).

Micropayments

The QUBIC coin enables feeless micropayments, allowing for high-frequency transactions in areas such as content monetisation and IoT communications. This functionality is critical for applications requiring seamless, zero-cost transactions.

AI Training and Validation

Through its Useful Proof of Work model, Qubic channels computational resources into training Artificial Neural Networks (ANNs). This decentralised approach supports advancements in AI, contributing to innovation in machine learning and artificial intelligence.

Decentralised Exchange

QX, Qubic's decentralised exchange, supports secure and transparent trading of digital assets without intermediaries. Leveraging sub-second finality, QX offers structured fees for execution, trading services, and storage, making it suitable for high-frequency trading (HFT) and strengthening network utility. The development of Vottun Bridge improves QX's interoperability by enabling cross-chain transactions between Qubic, Ethereum, and Arbitrum.

4

CONSENSUS MECHANISM

One of the most crucial elements in the Qubic network is the consensus mechanism, through which Computors agree on the state of the blockchain and enable the processing of transactions in a secure and efficient manner. Qubic uses a Quorum-based consensus mechanism, with Byzantine Fault Tolerance (BFT), as outlined in Section 3.2. The next section will give a detailed protocol description and a security

What is Byzantine Fault Tolerance (BFT)?

Byzantine Fault Tolerance (BFT) is a security model that allows a network to function even when some nodes act maliciously. Qubic uses BFT in its quorum-based consensus to enhance security and reliability, even if up to one-third of nodes fail or act maliciously. For an in-depth explanation, see (Lamport et al., 1982).

4.1 Detailed Protocol Description

Unlike Nakamoto's (2008) Proof of Work model, which relies on computational work to secure the network, Qubic's consensus relies on quorum selection and voting among Computors to finalise ticks and transactions. The approach obviates the need for miners to solve cryptographic puzzles and instead guarantees security and resilience, via Computors, through distributed voting and fault tolerance.

Although Qubic relies on Useful Proof of Work (UPoW) to harness miners' computational power for useful AI tasks, the consensus mechanism is technically independent. UPoW incentivises contributing computational power for the network's AI-related aims, and quorum-based consensus works on reaching an agreement regarding the state of the blockchain. This separation between the two allows Qubic to achieve high efficiency in consensus without forcing the computational overhead that often characterises PoW-based consensus models.

This section presents a step-by-step walkthrough of Qubic's consensus algorithm, including mathematical models and proofs that demonstrate its effectiveness and robustness.

4.1.1 Overview of the Quorum-Based Consensus Algorithm

Qubic's consensus mechanism is designed to achieve agreement among a distributed set of Computors while tolerating Byzantine faults. The algorithm operates in discrete time periods called epochs, during which transactions are proposed, validated, and committed to the blockchain. During an epoch, there is a continuous sequence of consensus rounds called ticks, where Computors independently validate and execute transactions and reach agreement on results.

Key Components:

- Computors: Entities associated with nodes responsible for validating transactions, executing smart contracts, and participating in consensus.
- Computor Index: Each computor has its specific index per Epoch. The indices are from 0 to 675.
- Tick: Set of transactions to be executed and agreed on in one round of the consensus algorithm, with digests of the states of smart contracts, spectrum, and universe as well as temporal information, which uniquely identifies the tick in the sequence of ticks.
  Spectrum and universe contain all information about who owns how many QUBIC coins and other assets at this point of time.

  Note: In the source code of Qubic a Tick is one vote of a specific computor.
- Tick leader: The tick leader is the Computor that is responsible for a certain tick.

  The tick leader can be identified by this formula, which computes it's Computor index:

$$\textit{<COMPUTORINDEX>} = \textit{<TICKNUMBER>} \ \% \ 676$$

- Quorum: A subset of Computors required to reach consensus. In Qubic, a quorum consists of:

$$Q = 451 \textit{ Computors (out of a total of } N = 676\textit{)}$$

- Epochs: Epochs are broader time intervals (1 week) that consist of multiple ticks (consensus rounds). During each epoch, a sequence of consensus rounds is completed, and performance or rewards can be calculated based on the outcomes of these rounds.
- TickData: The TickData is the definition of a Tick, announcing the digests of the transactions to be included into the tick. The tick leader creates the TickData and will propagate the TickData in advance to the network.
- Arbitrator: A mechanism for dispute resolution and maintaining network integrity, as described in Section 3.2.3.

4.1.2 Qubic's Quorum Consensus Algorithm

The consensus process can be outlined as follows:

1. Each Computor is initialised with a distinct Computor index ranging from 0 to 675. The tick leader for a given tick *T* is the Computor with index *C*, given by T modulo 676, i.e.:

$$C = T \% 676$$

Example:

Tick: 15104383

ComputorIndex: 515

This resolves to:

$$C = 15104383 \% 676 = 515$$

2. The tick leader creates the "TickData".
   It packs the identifiers of the scheduled transactions, the contractFees, and marks everything with a timestamp, the tick number, and epoch.

   Each transaction is identified by its digests, which is the KangarooTwelve hash of the transaction.

   The complete packet is signed by the tick leader and broadcasted to the network. The time of broadcasting is defined by TICK_TRANSACTIONS_PUBLICATION_OFFSET. This parameter steers how many ticks in advance the tick leader sends out the TickData.

3. All other Computors in the network will receive the TickData and verify the signature. It is accepted only if the TickData is signed by the known tick leader Computor.
   If the TickData does not arrive in time to a certain Computor, this specific Computor will use its own version, which will be "empty" (no TickData).

4. To process the tick, the Computors need to have the full data of all the transactions, whose digests have been packed into the TickData by the tick leader.

The Computor checks if all transactions are already stored locally and if one or more are missing, it will request other Computors to send those transactions.

The Computor can only continue if all transactions are available locally.

5. When the TickData is verified, all transactions are available and verified, then the Computor will cast its vote on the tick.

6. Every Computor individually receives the votes from other Computors. A tick vote contains the tick number, epoch, the Computor index, timestamp and cryptographical state of the sending Computor.

   Ideally, each Computor sees 676 votes (including its own). But the received votes are grouped by their content continuously to apply the quorum rules. If at least 451 votes align in one group, it is called an aligned state and the Computors agree to proceed.

   According to Byzantine Fault Tolerance (BFT) principles, as outlined by Lamport et al. (1982) and later adapted by Castro and Liskov (1999), the network can only continue to tick when at least 451 Computors have the same view, achieving a two-thirds majority to maintain network stability.

   If more than 225 Computors cast votes for an empty tick, that means no other group will exceed 451 votes. Consequently, the network will decide to skip that tick, discarding the planned transactions. (226+ rule).

7. If at least 451 Computors have agreed (Consensus) on the content of the tick with their votes, the tick is processed, executing the transactions and proceeding to the next tick.

   If 226 or more Computors voted for an empty tick, the Computors proceed to the next tick without executing transactions.

Faulty State

The faulty state is used to mark Computors that carry out suspicious actions. If Computor A detects two different versions of TickData (or TickVote) from Computor B, it will mark Computor B as faulty.

The Arbitrator will use this information to potentially replace faulty Computors.

How transactions are sent and propagated across the network

Qubic transactions are not limited to simple coin transfers; they also facilitate the execution of smart contracts or communication between Computors. As with other blockchains, a Qubic transaction consists of several essential fields: source and destination addresses, and the amount to be transferred. Beyond these standard fields, Qubic transactions have a "Tick" field, designating the desired tick number for transaction inclusion, and an "InputType" field, specifying the procedure number of the destination smart contract. The "InputData" field provides the input data to be supplied to the specified smart contract procedure.

A transaction broadcast to a node will be propagated to six additional nodes by default, as configured by the DISSEMINATION_MULTIPLIER parameter.

Manual intervention from operators

Although the vote simply means YES/NO for the next tick, it is possible that votes can be split into more than 2 groups because it also contains node states digests, which can mismatch if operators run custom code or due to bugs from incompatible hardware. In the unlikely event of such a division, where no consensus can be reached, manual intervention from Computors may be necessary to ensure the progression of ticks.

4.2 Security Analysis

This subsection examines the consensus mechanism's resistance to various attack vectors and how it ensures network integrity.

4.2.1 Resistance to Byzantine Faults

Byzantine Fault Tolerance:

- The consensus algorithm is designed to tolerate up to:

$$f \leq \frac{N-1}{3}$$

faulty or malicious Computors within a network of *N* Computors.

Implications:

- Safety: The network ensures that no two honest Computors accept different ticks for the same epoch.
- Liveness: The network continues to make progress despite the presence of faulty Computors.

4.2.2 Ensuring Network Integrity

Cryptographic Security

- Transactions are signed by the sender.
- Consensus messages are signed by the responsible Computor.
- The state of a Computor (spectrum, universe, SC state) is hashed (with KangarooTwelve) every tick to ensure alignment and consistency across the network.

Consensus Resilience

- Quorum Size and Thresholds: Carefully chosen quorum sizes and consensus thresholds balance fault tolerance with performance.
- Diversity of Computors: Encouraging a wide distribution of Computors reduces the risk of centralization and increases security.

Governance Safeguards

- Arbitrator Mechanism: The Arbitrator oversees network operations and can intervene in exceptional circumstances, such as detecting widespread malicious activity.
- Supermajority Override: The Arbitrator can be overridden by a supermajority of Computors (451 out of 676), ensuring that control remains decentralised.

5

ECONOMIC MODELS

5.1. Emission Schedule

Having examined Qubic's infrastructure and consensus mechanisms, we now turn to its economic model, which sustains the network and incentivises ongoing participation.

5.1.1. Initial Coin Supply

At the very start of the Qubic network, a fixed total supply of QUBIC coins was set to ensure both stability of the network and sustainable long-term growth. Initially, a maximum supply of 1,000 trillion QUBIC coins was envisioned; however, this figure has subsequently been decreased by 80%, resulting in a revised cap of 200 trillion QUBIC coins. This intentional reduction is in line with Qubic's objectives to enhance scarcity and mitigate inflation.

- Total Supply: The maximum supply of QUBIC coins is 200 trillion, changed from the original supply limit to increase scarcity and reduce inflation.

The total supply information is detached from the ongoing emissions and burn mechanisms, which adjust the circulating supply post-launch.

5.1.2 Emission Phases

The emission of QUBIC coins is precisely scheduled to maintain network participation while keeping the level of inflation under control. Qubic’s emission model integrates scheduled emissions, burns, and halvings supported by the ‘Supply Watcher’ - a smart contract controlling burn rates in real time.

Qubic’s economics model incorporates both emission schedules and deflationary mechanisms to maintain long-term stability, drawing on concepts of controlled issuance and reward distribution as explored by Narayanan et al. (2016). This approach ensures that network participants are appropriately incentivised while avoiding inflationary pressures (Beiko, 2021).

Emission Phases:

1. Bootstrapping Phase (Years 1–2):

    - To encourage early adoption, emissions are set at a high rate of 1 trillion QUBIC per week

    - Outcome: This stage promotes early participation.

2. Stabilisation Phase (Years 3–5):

    - Starting from epoch 123, in year 3, part of the emissions is burned every week, starting at a 15% burn rate.

    - Halvings: There are scheduled halvings approximately every 52 epochs, approved by Quorum to guarantee community consensus. During these halvings, the proportion of QUBIC burned increases, effectively reducing the net supply in circulation without lowering the base emission rate of 1 trillion QUBIC per week.

    - Supply Watcher Adjustments: The Supply Watcher adjusts the burn rate to keep things stable. For example, 15% of weekly emissions were burned during Epoch 123, taking around 149 billion QUBIC out of circulation.

    - Outcome: Decreasing effective supply by means of burns through an initial 15% reduction in 2024, followed by annual halvings of the effective emission rate.

    - It is important to note that Qubic's emission schedule is not fixed. It is designed to be dynamic, and to evolve with the ecosystem. The Supply Watcher allows emissions and burn rates to be influenced by real-time factors such as smart contract activity and network conditions. This maintains flexibility in response to economic and market variability.

3. Sustainability Phase (Year 6 and Beyond):

    - Minimal Emission Focusing on Burns: Emissions reach a minimum value when the burn rate slowly surpasses the emission rate, causing a net reduction in the total supply.

    - Prolonged Scarcity and Value Maintenance: This stage highlights the scarcity of coins, while the Supply Watcher consistently monitors and adjusts burn rates to mitigate the risks of excessive deflation. As the burn rate takes precedence, the overall supply diminishes, thereby increasing QUBIC's scarcity.

Figure 3 shows the emission phases and reduction schedule, illustrating Qubic's effective weekly emissions after burns. From this visual representation, one can note a systematic decrease in Qubic's emission schedule over time, ensuring scarcity and sustainable economics.

Notes on flexibility: Since the Supply Watcher dynamically adjusts burn rates, the emission and burn figures here are estimates, not fixed values. The Supply Watcher aids stability by addressing potential concerns from miners and Computors over possible fluctuating rewards. This makes sure supply reduction is balanced, taking network conditions into account.

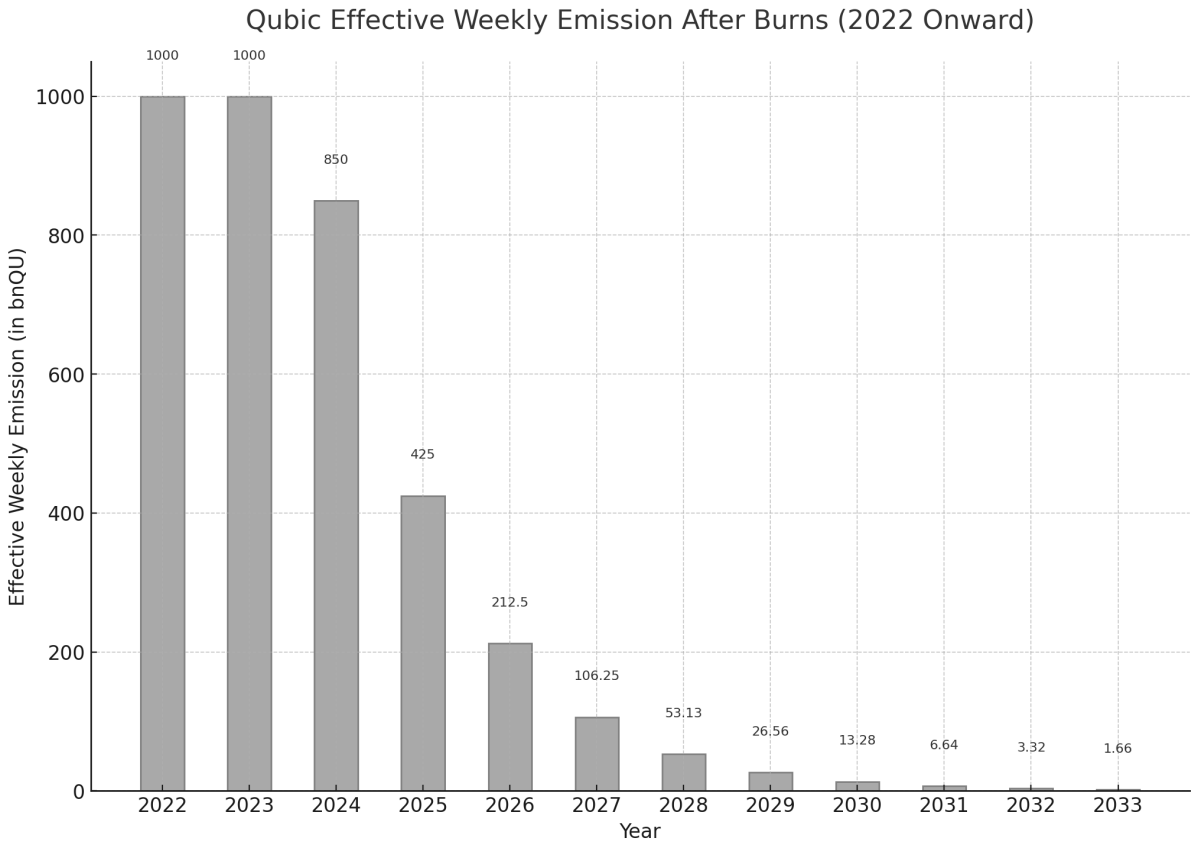

Figure 3: Qubic's effective weekly emission schedule after burns, illustrating a controlled reduction over years (starting from 2024) to support sustainable economics

Comparative Note: Qubic's emission model is inspired by the Bitcoin halving strategy (*figure 4*), but it adds flexibility thanks to the Supply Watcher and burn mechanisms. While Bitcoin decreases emissions through strict halving every four years, Qubic allows for an adaptation of the burn rate depending on the current state of the network. In this way, controlled scarcity can be achieved, and ongoing participation in the network can be incentivised without the sharp supply shocks associated with fixed halvings.

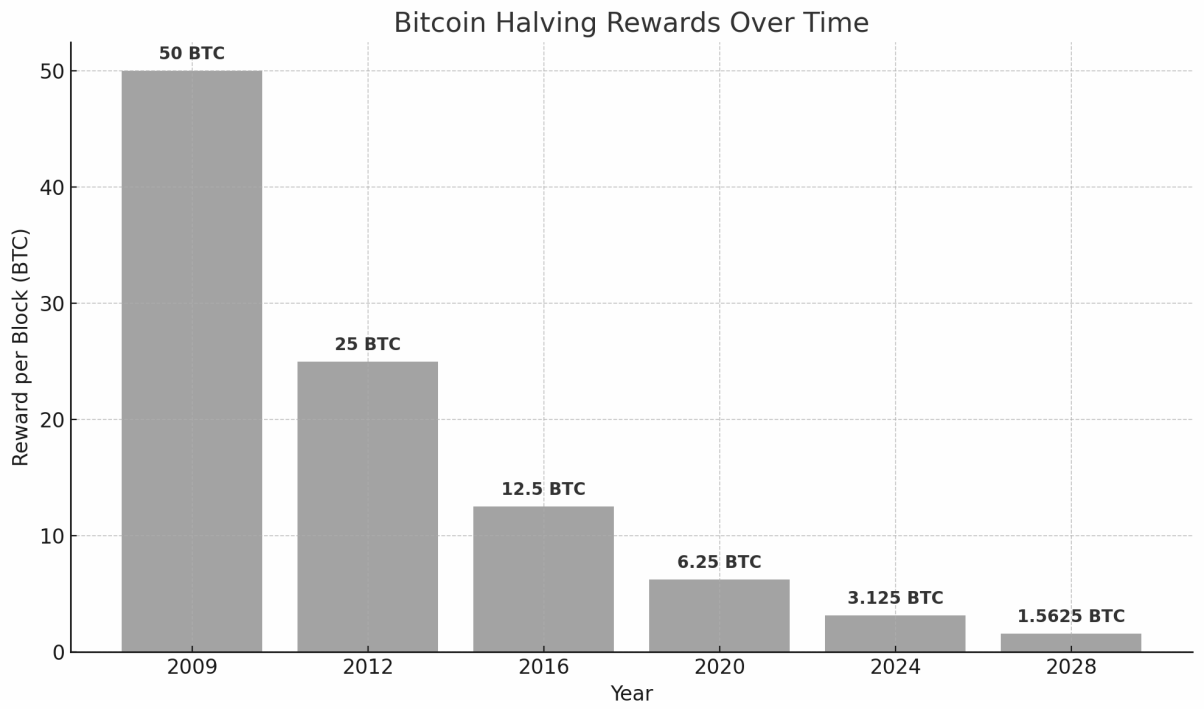

Figure 4: Bitcoin halving rewards over time

Source: *Independent Reserve, "Bitcoin Halving Explained" (*Independent Reserve, n.d.)

Mathematical Model of Emission

- Halving Schedule:

$$E(t) = \frac{E_0}{2^{\lfloor \frac{t}{n} \rfloor}}$$

This formula represents a halving schedule, where:

- E_0 is the initial emission rate.
- t is the time since launch of halvings.
- n is the interval (in years or epochs) after which the emission rate halves.
- $\frac{t}{n}$ represents the number of halvings that have occurred by time t, and $\lfloor x \rfloor$ is the floor function of x (rounding down to the nearest integer).

5.1.3 Reward Distribution

Qubic's reward distribution mechanism is a dynamic process. From the base reward which computors receive, they can define specific donations which can support certain purposes.

The revenue distribution process after each epoch is visualised below:

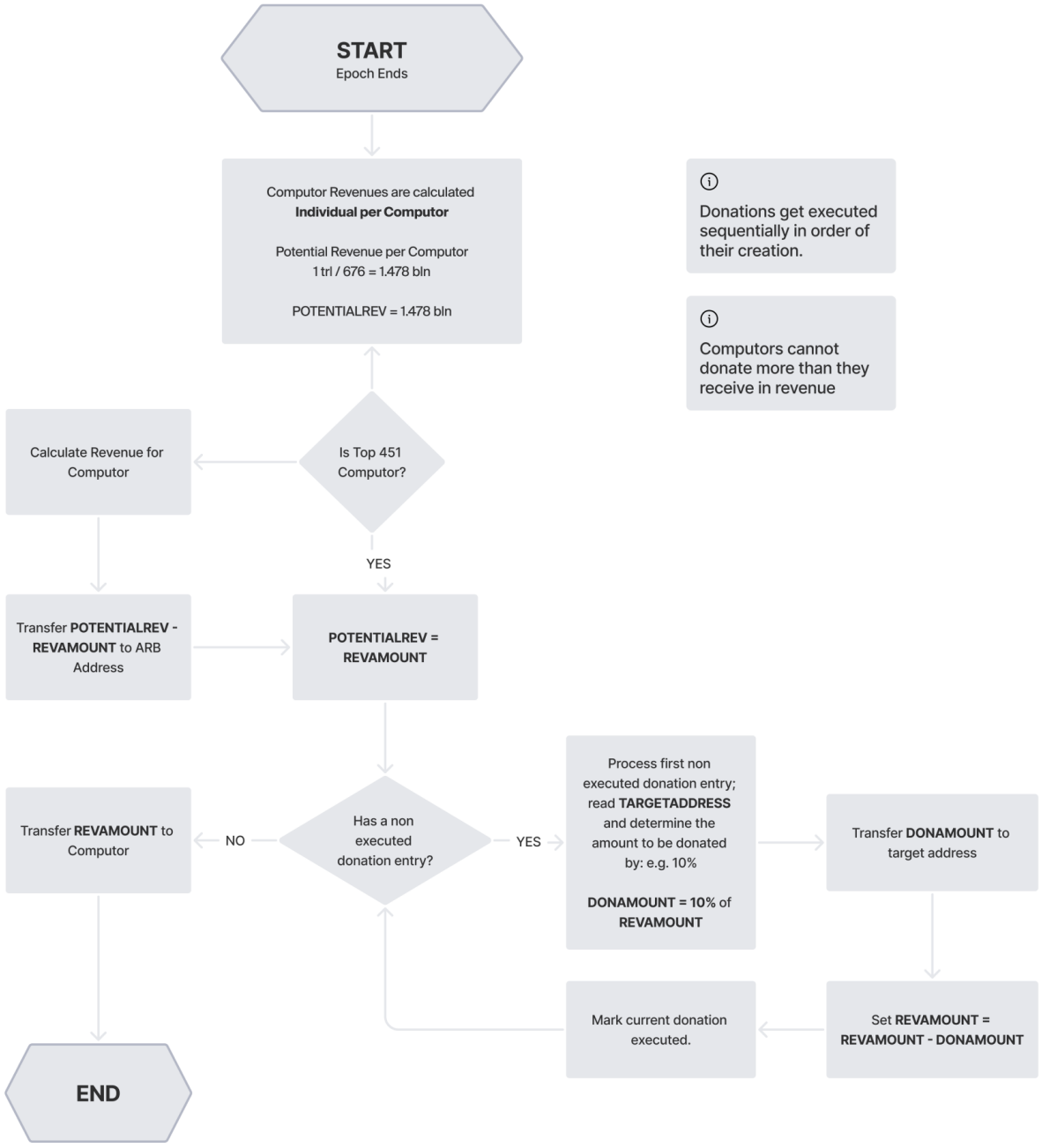

Figure 5: Overview of Revenue Calculation and Distribution Process

This diagram illustrates the overall framework for how revenues are calculated and distributed across the network. It begins with the definition of parameters, such as base rewards for Computors, and outlines the process of revenue allocation after each epoch.

Explanation of the Revenue Distribution Process:

1. Calculate Revenues for Computors: Each Computor's revenue is calculated based on its contributions and performance metrics.
2. Top 451 Computor Check: Only the top 451 Computors qualify for the reward in full, prioritising high performers to incentivise network stability.
3. Potential Revenue Adjustment: If a Computors performance is below the top 451 Computors, its revenue is decreased while the remaining part goes to Arbitrator.
4. Donation Execution: Registered donations, like the Supply Watcher Burn (15%) and CCF SC (8%), are deducted sequentially from Computor revenue and allocated to the respective addresses.
5. Final Revenue Transfer: After all deductions, the remaining revenue is transferred to the Computor, completing the distribution process.

This model ensures that revenues are allocated fairly while supporting Qubic's broader economic goals and donation commitments.

Figure 6 provides a step-by-step breakdown of the Computor revenue calculation process:

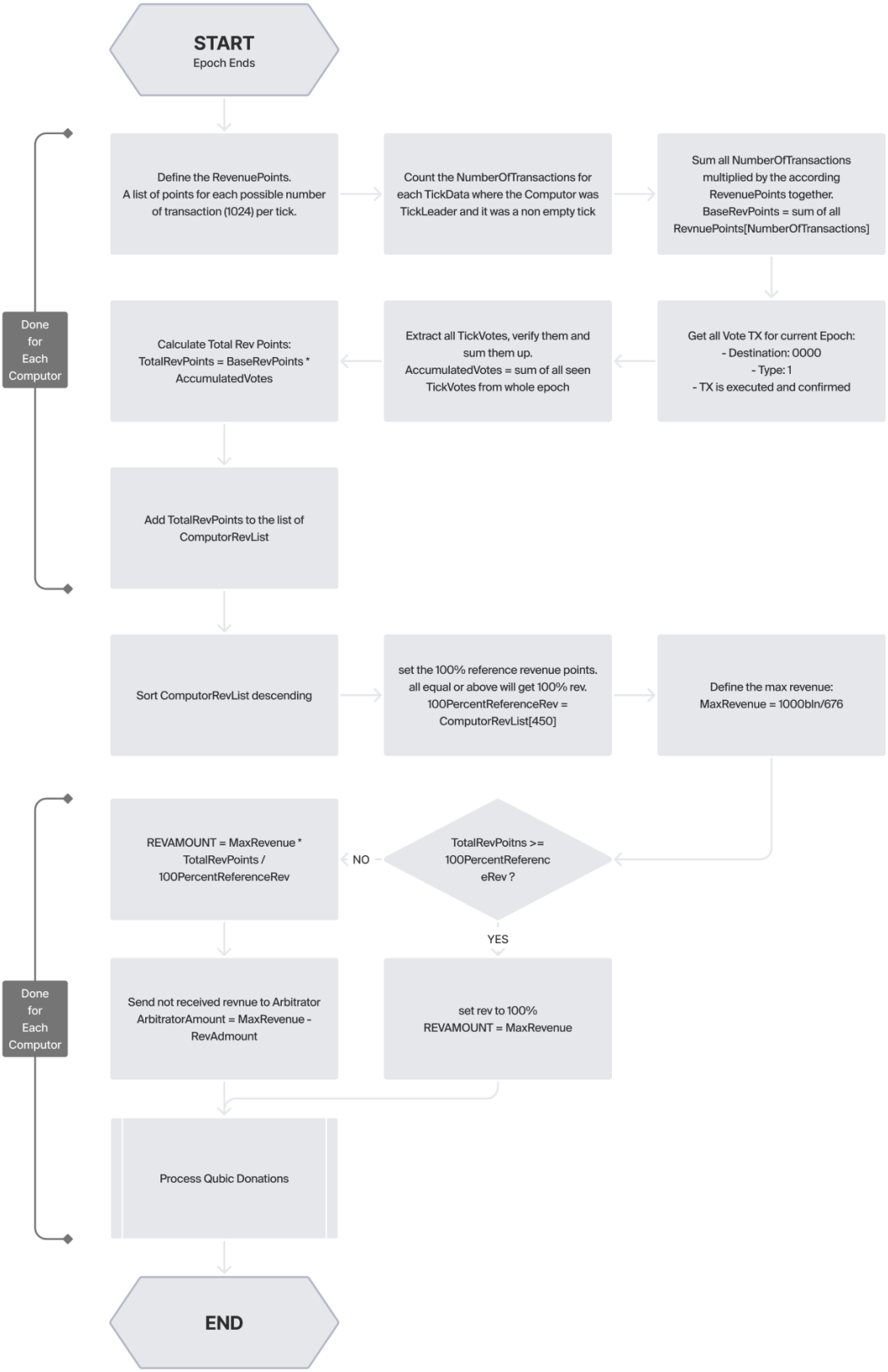

Figure 6: Detailed Flowchart for Computor Revenue Calculation

- The flow begins with the end of the event epoch, after which individual Computor revenues are calculated.
- Top 451 Computor Check: The diagram shows a conditional step determining if a Computor is within the top 451. If not, revenue adjustments are made accordingly.
- Sequential Donation Execution: Donations are processed in the order of their creation to ensure that the Computor cannot donate more than the revenue received.
- Revenue Transfer: Finally, after donation processing, the revenue is transferred to the Computor.

This detailed process reinforces the fairness and structure of Qubic's reward distribution, thus aligning it with economic goals and following network rules.

5.2 Deflationary Mechanisms

Deflationary mechanisms are implemented to control the circulating supply of QUBIC coins, contributing to economic sustainability.

5.2.1 Coin Burning

Coin burning involves permanently removing coins from circulation by sending them to an unspendable address. Qubic employs scheduled and event-driven coin burns.

Qubic's emission model includes a burn mechanism made possible by the Supply Watcher feature. Unlike fixed annual burn events or burns based on transactions, the Supply Watcher adjusts the burn rate in real time. This way, the burn rates are adjusted based on current conditions on the network, therefore not causing extreme deflation and ensuring stability.

- Weekly Burn Rate: The current weekly burn rate is approximately 150 billion QUBIC. This number can change based on the economic conditions of the network, which are monitored by the Supply Watcher.
- Role of Supply Watcher: The Supply Watcher adjusts the percentage of QUBIC burned in a week, aiming for a balance between emission reduction and overall network stability. This system gives the flexibility needed in managing effective emissions, allowing Qubic to adapt to changes in demand and participation.

5.2.2 Smart Contract Operations

- Contract Execution Fees:
    - Fees associated with executing smart contracts can include a burn component, further reducing supply.

5.2.3 Impact on Coin Supply

The combined effect of these deflationary mechanisms reduces the circulating supply over time, potentially increasing the scarcity of QUBIC coins. The total supply at any time is given by:

$$S_{circulating(t)} = S_{circulating(t-1)} + E(t) - B(t)$$

where:

- $E(t)$ = emission at time *t.*
- $B(t)$ = total coins burned at time *t.*

5.3 Economic Incentives

This section examines how the economic model of Qubic aligns incentives for various stakeholders in terms of participation and security within the network. Drawing on ideas presented in Section 3.1.3, we consider how rewards and economic structures support long-term growth and stability.

5.3.1 Alignment of Incentives

Qubic's economic model is designed to ensure that participants are rewarded for behaviours that contribute positively to the network.

Computors:

- Incentive to Maintain High Performance: Computors are incentivised to operate reliably and efficiently to receive rewards.
- Incentive to Mine or Engage Miners: Current Computors and potential future Computors are in competition to stay or become Computors, requiring enough mined solutions to qualify for the next epoch.

- Contribution to Network Security: By validating transactions and participating in consensus, Computors strengthen the network's integrity.

Miners:

- Incentive to Provide Computational Power: While not rewarded by Qubic directly, miners are rewarded via agreements with their associated Computors, based on their contribution to UPoW tasks, encouraging them to optimise hardware and algorithms.
- Support for AGI Development: Miners' computational efforts contribute to AI training within Aigarth, aligning individual incentives with the network's broader goals.

Coin Holders:

- Supply Scarcity and Network Engagement: The coin’s design incorporates deflationary mechanisms to manage supply, while network expansion seeks to create an active ecosystem. These elements are structured to support long-term sustainability.
- Participation in Governance: Holders of coins may be able to participate in governance decisions whose interests are aligned with network success.

5.3.2 Sustainability of Rewards

Qubic's approach to reward sustainability is supported by Beiko (2021) in the analysis of emission models, where he highlights the importance of reaching an equilibrium between rewards and network stability. Qubic uses a controlled emission schedule coupled with deflationary policies like coin burns to create a stable and incentivised economic environment for its stakeholders, which is aligned with standard best practices within blockchain economics. According to Beiko (2021), supply reduction needs to align with the incentives of participants for the long-term health and engagement of the network.

Qubic’s economic model ensures that rewards are sustainable over the long term:

- Controlled Emission: The emission schedule gradually reduces coin issuance, preventing excessive inflation.

- Deflationary Offsets: Coin burns counteract inflationary pressures, balancing the supply-demand dynamics.

- Economic Equilibrium: The balance between emissions and burns aims to achieve an equilibrium that supports network operations.

5.3.3 Network Growth and Stability

By incentivizing key behaviours, the economic model promotes network growth and stability.

Encouraging Participation:

- Diverse Ecosystem: A broad base of Computors and miners enhances decentralisation and resilience.

Enhancing Security:

- Incentivised Compliance: Rewards motivate participants to adhere to protocol rules.
- Resistance to Attacks: Economic disincentives for malicious behaviour reduce the likelihood of attacks.

5.3.4 Long-Term Economic Viability

The economic model is designed to ensure the long-term viability of the Qubic network.

- Adaptability: Mechanisms are in place to adjust economic parameters based on network conditions, allowing for flexibility in response to changes.
- Alignment with Network Goals: The economic incentives are closely tied to the network's objectives, such as supporting AGI development through Aigarth.
- Community Engagement: By aligning the interests of participants, the model creates a strong community invested in the network's success.

6

SECURITY CONSIDERATIONS

Security is paramount in the design and operation of the Qubic network. This section delves into the cryptographic foundations that underpin the network's security and examines potential attack vectors along with the strategies employed to mitigate them. By leveraging robust cryptographic algorithms and implementing comprehensive security protocols, Qubic aims to provide a secure environment for decentralised transactions and computations, including those related to AGI development through Aigarth.

6.1 Cryptographic Foundations

The security of the Qubic network relies on well-established cryptographic algorithms and protocols. This subsection details the cryptographic primitives and mechanisms employed to ensure data integrity, authenticity, confidentiality, and non-repudiation.

6.1.1 Cryptographic Hash Functions

Algorithm Used: KangarooTwelve

Purpose: KangarooTwelve is used for hashing operations within the network, including tick votes, tick data, transactions, and merkle trees of spectrum, universe, and smart contract states. It is a variant of the Keccak algorithm family (which SHA-3 is based on) but optimised for speed and scalability. KangarooTwelve's scalability and speed make it ideal for high-throughput environments, as highlighted by Bertoni et al. (2018). Its collision resistance and efficiency support Qubic's need for real-time consensus while ensuring data integrity across the network.

Properties:

- Collision Resistance: It is computationally infeasible to find two different inputs that produce the same hash output.
- Preimage Resistance: Given a hash output, it is computationally infeasible to find an input that produces that hash.
- Second Preimage Resistance: Given an input and its hash, it is infeasible to find a different input with the same hash.

Role in Qubic:

- Tick Hashing: Ensures the integrity of ticks by linking each tick to the previous one through the hashes of a predefined set of keys, using KangarooTwelve for efficient computation.
- Ensuring consistency of Computor states: By computing hashes of the spectrum, universe, and smart contract states and including them in the consensus protocol, Computors ensure the agreement of their state in each tick.

- Identifying Transactions: A hash of each transaction (also called digest) is computed using KangarooTwelve for identifying the transaction.
- Merkle Trees: Utilised for computing hashes of large data structures, such as spectrum and universe, efficiently and securely.

6.1.2 Digital Signatures

Algorithm Used: FourQ (adapted)

FourQ is an elliptic curve developed by Microsoft Research. It is designed for key agreement schemes (elliptic-curve Diffie–Hellman) and digital signatures (Schnorr) and offers about 128 bits of security (Costello & Longa, 2015).

- Purpose: Sign/Verify is employed to authenticate transactions and messages within the network, ensuring that only authorised parties can initiate actions.
- Properties:
    - Authenticity: Verifies the identity of the sender.
    - Non-Repudiation: Prevents the sender from denying the authenticity of their signature.
    - Integrity: Ensures that the message has not been altered.
- Role in Qubic:
    - Transaction Signing: Users sign transactions with their private keys, and Computors verify signatures using the corresponding public keys.
    - Consensus Messages: Computors sign their votes and proposals during the consensus process to maintain accountability and traceability.

6.1.3 Key Management

Public and Private Keys:

- Generation: Keys are generated using secure random number generators to ensure unpredictability.

- Storage: Private keys must be securely stored by users. Qubic encourages the use of hardware wallets or secure enclaves for key storage.

6.1.4 Secure Communication Protocols

Message Signing:

- Purpose: Ensuring authenticity and integrity
- Implementation: Messages sent in Qubic are signed by their sender. This allows the receiver to verify the authenticity and integrity of the Message.

6.2 Attack Vectors and Mitigations

This subsection identifies potential vulnerabilities within the Qubic network and outlines the strategies employed to mitigate them. By proactively addressing these threats, Qubic enhances its resilience against malicious actors and network disruptions.

In mitigating potential Sybil and 51% attacks, Qubic's model incorporates Byzantine Fault Tolerance principles, taking inspiration from studies in cyber attack defence taxonomy as reviewed by Simmons et al. (2009).

6.2.1 Sybil Attacks

- Description: An adversary creates multiple identities (Sybil nodes) to gain disproportionate influence.
- Mitigation:
    - Useful Proof of Work (UPoW): UPoW channels computational power towards useful tasks such as AI training, rendering an attack cost-prohibitive for any attackers amassing the computational resources necessary to perform a successful Sybil attack.
    - Proper Signing: In Qubic only the 676 computors have the right to vote. Without having the corresponding secret keys, a sybil attack is therefore not possible.

6.2.2. Forking Attacks

- Description: Malicious Computors create alternative chains to confuse or split the network.
- Mitigation:
    - Strong Finality: Once a tick is accepted by the Quorum, it is considered final, and subsequent ticks build upon it.
    - Chain Selection Rule: Honest Computors follow the chain with the highest cumulative support from quorum votes.

6.2.3 Collusion Attacks

- Description: A group of malicious Computors colludes to manipulate consensus decisions.
- Mitigation:
    - Fault Tolerance Threshold: The algorithm tolerates collusion as long as the number of colluding Computors is less than 226.
    - Randomised Quorum Selection: The unpredictable selection of Computors for each quorum decreases the possibility of sustained collusion.

6.2.4 Replay Attacks

- Description: Attackers resend valid transactions to disrupt the network.
- Mitigation:
    - Duplication ignore: Already known transactions are ignored by the computors.

6.2.5 51% Attacks

Threat Description:

- An attacker gains control of more than 50% of the network's computational resources or voting power, allowing them to manipulate the blockchain by reversing transactions or preventing new transactions from being confirmed.

Mitigation Strategies:

- Byzantine Fault Tolerance: The consensus mechanism tolerates up to $\frac{1}{3}$ faulty Computors, making it infeasible for an attacker to succeed without controlling a significant portion of the network.
- Decentralisation of Computors: Encouraging widespread participation reduces the risk of centralization.
- In Qubic, to take over the network, one needs >= 451 votes, which is approx ⅔ of the network.
- Economic Disincentives: The cost of acquiring sufficient resources to perform a 51% attack outweighs potential gains.

Reference: (Eyal & Sirer, 2014)

6.2.6 Eclipse Attacks

Threat Description:

- An attacker isolates a node or a group of nodes by controlling all their incoming and outgoing connections, enabling the attacker to manipulate the victim's view of the network.

Mitigation Strategies:

- Diverse Peer Selection: Nodes maintain connections with a diverse set of peers, reducing the chance of all connections being controlled by an attacker.
- Connection Limits: Limiting the number of connections from a single IP address or subnet.
- Separating In- and Outgoing Connections: Nodes cannot be blocked from outgoing connections.

- Periodic Peer Refreshing: Regularly random updating peer connections to prevent long-term isolation.

6.2.7 Smart Contract Vulnerabilities

Threat Description:

- Flaws in smart contract code can lead to unintended behaviour, security breaches, or exploitation by attackers.

Mitigation Strategies:

- Code Auditing: Mandatory audits of smart contracts by trusted third parties before deployment.
- Restricted Language Features: Preventing the use of complex or risky language features in smart contracts.

6.2.8 Quantum Computing Threats

Threat Description:

- The advent of quantum computing could potentially break traditional cryptographic algorithms, compromising the security of the network.

Mitigation Strategies:

- Quantum-Resistant Cryptography:
    - Research and Development: Monitoring advancements in quantum computing and developing quantum-resistant cryptographic schemes.
    - Algorithm Agility: Designing the protocol to allow for the integration of new cryptographic algorithms as they become available.
- Post-Quantum Algorithms: Exploring algorithms like lattice-based cryptography (e.g., NTRU) or hash-based signatures (e.g., XMSS).

Reference: (Bernstein et al., 2017)

6.2.9 Malware and Node Compromise

Threat Description:

- Malware infections or unauthorised access can compromise nodes, leading to data breaches or participation in malicious activities.

Mitigation Strategies:

- Secure Software Practices: Implementing code security best practices and regular security assessments.
- Isolation Techniques: Qubic runs in bare-metal without the need of an underlying operating system.
- Regular Updates and Patching: Keeping software and dependencies up to date to mitigate known vulnerabilities.

7

CONCLUSION

7.1 Summary of Contributions

This whitepaper has outlined Qubic's architecture and its approach to overcoming challenges within both the blockchain and artificial intelligence (AI) domains. By using a Layer 1 blockchain, Qubic integrates economic mechanisms, such as Useful Proof of Work (UPoW) and Byzantine Fault Tolerant (BFT) quorum-based consensus, which align network security with productive AI computations. Through Aigarth, Qubic enables scalable AGI development, setting it apart from conventional blockchain solutions by creating a more resource-efficient and ethically decentralised platform. The design also addresses economic sustainability through an economic model that includes controlled emission schedules and deflationary measures, balancing reward distribution and facilitating long-term participation.

Qubic's infrastructure demonstrates performance improvements, such as sub-second transaction finality and bare-metal deployment, which reduce latency and increase computational capacity. These capabilities enable the network to support high-demand, real-time applications while remaining energy-efficient. Further, Qubic's governance model promotes decentralisation and resilience by distributing decision-making authority among network participants and ensuring fault tolerance in adverse conditions.

8

REFERENCES

8.1. Bibliography

1. Atzei, N., Bartoletti, M., & Cimoli, T. (2017). *A survey of attacks on Ethereum smart contracts (SoK)*. *Proceedings of the 6th International Conference on Principles of Security and Trust (POST 2017)*, Lecture Notes in Computer Science, 10204, 164–186. Springer. https://doi.org/10.1007/978-3-662-54455-6_8

2. Beiko, T. (2021). *Ethereum EIP-1559: Transaction Fee Market*. Ethereum Improvement Proposal. https://eips.ethereum.org/EIPS/eip-1559

3. Belchior, R., Vasconcelos, A., Guerreiro, S., & Correia, M. (2021). *A Survey on Blockchain Interoperability: Past, Present, and Future Trends.* ACM Computing Surveys, 54(8), Article 168. https://dl.acm.org/doi/10.1145/3471140

4. Bernstein, D. J., et al. (2017). *Post-Quantum Cryptography: The State of the Art.* Annual International Cryptology Conference (CRYPTO). https://eprint.iacr.org/2017/314

5. Bertoni, G., Daemen, J., Hoffert, M., & Van Assche, G. (2018). KangarooTwelve: Fast Hashing Based on Keccak-p. Retrieved from https://keccak.team/files/KangarooTwelve.pdf

6. BitCompliance S.L. (2024). *Legal Note on the Legal Nature of the Qubic Token.* Retrieved from the legal document issued on July 19, 2024.

7. Cachin, C., & Vukolić, M. (2017). *Blockchain Consensus Protocols in the Wild*. arXiv preprint arXiv:1707.01873. https://arxiv.org/abs/1707.01873

8. Castro, M., & Liskov, B. (1999). *Practical Byzantine Fault Tolerance*. https://pmg.csail.mit.edu/papers/osdi99.pdf

9. Costello, C., & Longa, P. (2015). *FourQ: Four-dimensional decompositions on a Q-curve over the Mersenne prime*. Presented at ASIACRYPT 2015. Microsoft Research. Retrieved from https://www.microsoft.com/en-us/research/project/fourqlib/

10. Decker, C., & Wattenhofer, R. (2013). *Information propagation in the Bitcoin network.* IEEE P2P 2013 Proceedings, 1–10. https://ieeexplore.ieee.org/document/6688704

11. Eyal, I., & Sirer, E. G. (2014). *Majority is Not Enough: Bitcoin Mining is Vulnerable.* Financial Cryptography and Data Security. https://arxiv.org/abs/1311.0243

12. Gabuthy, Y. (2023). *Blockchain-Based Dispute Resolution: Insights and Challenges.* Games, 14(3), 34. https://doi.org/10.3390/g14030034

13. Independent Reserve. (n.d.). *Bitcoin halving explained*. Independent Reserve. Retrieved [Retrieved November 3, 2024], from https://www.independentreserve.com/blog/knowledge-base/bitcoin-halving-explained

14. Lamport, L., Shostak, R., & Pease, M. (1982). *The Byzantine Generals Problem*. https://lamport.azurewebsites.net/pubs/byz.pdf

15. Nakamoto, S. (2008). *Bitcoin: A Peer-to-Peer Electronic Cash System*. https://bitcoin.org/bitcoin.pdf.

16. Narayanan, A., Bonneau, J., Felten, E., Miller, A., & Goldfeder, S. (2016). *Bitcoin and Cryptocurrency Technologies: A Comprehensive Introduction.* Princeton University Press. https://press.princeton.edu/books/hardcover/9780691171692/bitcoin-and-cryptocurrency-technologies

17. Nguyen, T. A. N., Gangadhar, S., & Sterbenz, J. P. G. (2016). *Performance Evaluation of TCP Congestion Control Algorithms in Data Center Networks.* Proceedings of the 11th International Conference on Future Internet Technologies (CFI '16), 21–28. https://doi.org/10.1145/2935663.2935669

18. Nguyen, T. T. K., Truong, H. T. T., & Tuong, N. H. (2019). *A Survey on Applications of Game Theory in Blockchain*. Retrieved from https://arxiv.org/pdf/1902.10865.pdf.

19. Qubic Team (2024). *Qubic Achieves Over 55 Million Transfers Per Second for Smart Contract Executions.* Blog post. https://qubic.org/blog-detail/qubic-achieves-over-55-million-transfers-per-second-for-smart-contract-executions

20. Rosenblum, M., & Garfinkel, T. (2005). *Virtual Machine Monitors: Current Technology and Future Trends*. IEEE Internet Computing, 38(5), 39–47. https://ieeexplore.ieee.org/document/1430630

21. Shostack, A. (2014). *Threat Modelling: Designing for Security*. Wiley Publishing. https://www.wiley.com/en-us/Threat+Modeling%3A+Designing+for+Security-p-9781118809990

22. Szabo, N. (1997). *Formalizing and Securing Relationships on Public Networks.* https://nakamotoinstitute.org/formalizing-securing-relationships

23. Xu, J., Lu, Q., Gao, F., & Zhang, H. (2020). *Incentivizing Blockchain Ecosystem Development: A Game-Theoretical Approach*. Journal of Systems Science and Complexity, 33(4), 918–933. https://link.springer.com/article/10.1007/s11424-020-9189-2

24. Yli-Huumo, J., Ko, D., Choi, S., Park, S., & Smolander, K. (2016). *Where is current research on blockchain technology?—A systematic review*. PLoS ONE, 11(10), e0163477. https://journals.plos.org/plosone/article?id=10.1371/journal.pone.0163477

25. Zolfagharinejad, M., Alegre-Ibarra, U., Chen, T., et al. (2024). *Brain-inspired computing systems: a systematic literature review*. European Physical Journal B, 97, 70. https://doi.org/10.1140/epjb/s10051-024-00703-6

8.2. Further Reading

To support deeper exploration of the core technologies and approaches outlined in this whitepaper, the following resources offer insights into blockchain fundamentals, consensus mechanisms, cryptography, economic models, and advanced artificial intelligence (AI) integrations.

Blockchain Architecture and Consensus Mechanisms

- Nakamoto, S. (2008). *Bitcoin: A Peer-to-Peer Electronic Cash System.*
  The seminal paper that introduced the concept of decentralised digital currency, laying the groundwork for blockchain consensus models such as Proof of Work.

- Lamport, L., Shostak, R., & Pease, M. (1982). *The Byzantine Generals Problem.* ACM Transactions on Programming Languages and Systems, 4(3), 382–401
  A classic work that introduces Byzantine Fault Tolerance (BFT), essential for understanding consensus in distributed networks and its implementation in Qubic's consensus model.

- Narayanan, A., Bonneau, J., Felten, E., Miller, A., & Goldfeder, S. (2016). *Bitcoin and Cryptocurrency Technologies: A Comprehensive Introduction.* Princeton University Press.
  An in-depth introduction to blockchain technology and the cryptographic principles underlying it, including discussions on consensus and transaction validation.

- Castro, M., & Liskov, B. (1999). *Practical Byzantine Fault Tolerance.* Proceedings of the Third Symposium on Operating Systems Design and Implementation.
  Overview of the Byzantine Fault Tolerance (BFT) model, critical for understanding consensus in decentralised networks.

Cryptographic Foundations and Security Protocols

- NIST (2015). *SHA-3 Standard: Permutation-Based Hash and Extendable-Output Functions.* NIST FIPS 202.
  A detailed specification of SHA-3, which is essential for blockchain security protocols including hashing functions for block headers and Merkle trees.

- Bertoni, G., Daemen, J., Hoffert, M., & Van Assche, G. (2018). *KangarooTwelve: Fast Hashing* Based on Keccak-p. Submission to NIST for SHA-3 Derived Functions.
  A practical hashing function built on Keccak-p, KangarooTwelve provides fast, secure hashing suited for high-throughput applications, supporting Qubic's cryptographic security needs.

- Micali, S., Rabin, M. O., & Vadhan, S. P. (1999). *Verifiable Random Functions.* Proceedings of the 40th Annual Symposium on Foundations of Computer Science.
  Introduction to Verifiable Random Functions (VRFs), a crucial component for fair and unpredictable quorum selection.

Smart Contracts and Programmable Money

- Szabo, N. (1997). *Formalizing and Securing Relationships on Public Networks.* First Monday, 2(9).
  A foundational work on smart contracts, detailing the concept of programmable money and self-executing agreements, which are central to decentralised applications.

- Luu, L., Chu, D. H., Olickel, H., Saxena, P., & Hobor, A. (2016). *Making Smart Contracts Smarter.* ACM SIGSAC Conference on Computer and Communications Security.
  A critical exploration of smart contract vulnerabilities, including formal verification methods to prevent common security breaches.

Tokenomics and Economic Models in Blockchain

- Kroll, J. A., Davey, I. C., & Felten, E. W. (2013). *The Economics of Bitcoin Mining, or Bitcoin in the Presence of Adversaries.* WEIS.
  Explores the economic dynamics in mining and incentive structures, relevant to understanding Qubic's emission model and reward mechanisms.

- Saleh, F. (2021). *Blockchain Without Waste: Proof-of-Stake.* The Review of Financial Studies, 34(3), 1156–1190.
  Analyzes Proof-of-Stake (PoS) and its efficiency relative to Proof of Work (PoW), with concepts applicable to Qubic's sustainable economic model.

Advanced Artificial Intelligence Integration and AGI Development

- Silver, D., Huang, A., Maddison, C. J., Guez, A., Sifre, L., & Van Den Driessche, G. (2016). *Mastering the Game of Go with Deep Neural Networks and Tree Search.* Nature, 529(7587), 484-489.
  A landmark paper demonstrating practical applications of advanced AI in complex decision-making, relevant for understanding the role of AGI in Qubic.

- Gabriel, I. (2020). *Artificial Intelligence, Values, and Alignment. Ethics and Information Technology,* 22(1), 11-21. doi:10.1007/s10676-020-09578-0.
  Explores the ethical challenges of AI development, particularly focusing on the alignment of AI values with societal goals.

- Vivancos, D. (2023). *Artificiology.* Retrieved from *https://www.linkedin.com/pulse/artificiology-david-vivancos-5zmqf/*.
  A thought-provoking perspective on the intersection of AI, AGI, and human cognitive augmentation. Note: This article is non-peer-reviewed and reflects the author's personal insights.

Game Theory and Distributed Systems

- Yao, A. C. (1982). *Protocols for Secure Computations.* Proceedings of the 23rd Annual Symposium on Foundations of Computer Science (SFCS).
  Game-theoretic analysis of secure computations, underpinning the security and stability of decentralised systems.

- Eyal, I., & Sirer, E. G. (2014). *Majority is not Enough: Bitcoin Mining is Vulnerable.* International Conference on Financial Cryptography and Data Security.
  Analyses the 51% attack vulnerability and strategies to mitigate it, offering insights applicable to Qubic's BFT-enhanced consensus model.

9

APPENDICES

9.1 Glossary

Definitions of specialised terms and acronyms used throughout the document.

1. **AGI (Artificial General Intelligence)**: The ability of an AI system to understand, learn, and apply intelligence across a wide range of tasks, comparable to human cognitive abilities.

2. **Aigarth**: A project deeply coupled to Qubic that leverages the Useful Proof of Work (UPoW) model for training AGI models and other advanced AI applications.

3. **Bare-Metal Deployment**: Running applications directly on hardware without operating system or virtualisation, enhancing performance and security.

4. **Burn Mechanism**: A process by which coins are permanently removed from circulation, typically through network activities like smart contract execution fees, to help control inflation.

5. **Byzantine Fault Tolerance (BFT)**: A security model that enables network functionality even when a portion of nodes act maliciously.

6. **Computor**: A specialised node in the Qubic network responsible for validating transactions, securing the network, and participating in quorum consensus in exchange for QUBIC coins as rewards.

7. **Economics**: The economic structure and principles governing the issuance, distribution, and utility of coins within a network, in this case, the QUBIC coins within the Qubic network.

8. **Emission Model**: The structured schedule for releasing QUBIC coins into circulation, guiding how and when rewards are distributed to participants.

9. **Epoch**: A predefined time period (a week) in the Qubic network that structures the phases of reward distribution, mining qualification, and consensus activities.

10. **Efficiency Factor (E)**: A multiplier representing the proportion of successful solutions out of total attempts by a miner, reflecting hardware and algorithm efficiency.

11. **Hash Rate**: A measure of computational power, expressed in iterations per second (it/s), that indicates the number of potential solutions a miner's hardware can attempt.

12. **Miners**: Network participants who provide computational power to support tasks in the Useful Proof of Work (UPoW) model, receiving QUBIC coins as rewards for valid contributions from Computors.

13. **QUBIC coins**: The digital currency used within the Qubic network to reward Computors, facilitate transactions, and support network operations.

14. **Qubic Network**: A decentralised platform designed for secure, scalable, and efficient computation, supporting AGI development, economics, and consensus via the Useful Proof of Work (UPoW) model.

15. **Quorum**: A consensus model that requires a majority threshold, ensuring network integrity through Byzantine Fault Tolerance.

16. **Reward Allocation**: The process by which QUBIC coins are distributed to network participants in proportion to their contributions to the network.

17. **Solution Submission Rate (S_{rate})**: The rate at which valid computational solutions are submitted to the network.

18. **Spectrum**: Stores the number of QUBIC coins each entity owns at the current time / tick, including some information about incoming and outgoing transfers.

19. **Supply Watcher**: A Qubic network entity that monitors the total supply of QUBIC coins, triggering burn events to maintain economic stability.

20. **Tick**: Set of transactions to be executed and agreed on in the consensus algorithm, with digests of the whole states of smart contracts, spectrum, and

universe as well as temporal information, which uniquely identifies the tick in the sequence of ticks.

21. **Universe**: stores information about all assets (except QUBIC coins) existing in the Qubic blockchain at the current time / tick, including information on who owns and possesses them.

22. **Useful Proof of Work (UPoW)**: channels computational efforts toward AI and other valuable tasks instead of arbitrary problem-solving typical of traditional PoW models.

Acknowledgment of Contributors to the Whitepaper

This whitepaper represents a collaborative effort, with contributions from experts in blockchain technology, cryptography, artificial intelligence, and economic modelling. The following individuals played key roles in its research, drafting, and review:

Daniel Diez, iam333 , Zgirt, JOETOM, Dr Philipp Werner, dkat, frog-rabbit, mksala, pjdubs, Oreo, Eric Fung, David Vivancos, Dr Jose Sanchez, Come-from-Beyond, Foleycious, Talentnodes, Peter, MrUnhappyX, Dr Karin Lorez, CryptoDeighs, Crypdro

Insights and contributions from the Qubic community are also gratefully acknowledged.