Oako Network Litepaper

Oako Labs 2025

Abstract

This paper contains a brief description of Oako, a FHE-encrypted P2P network that supports distributed computations and storage: the structure, main features, and unique approaches used to make it work and scale in practical implementations. It is intended for all audiences and does not cover mathematics or cryptography in much detail; a full paper will be published before mainnet launch.

Introduction

Oako is a decentralized general purpose peer-to-peer network that allows anyone to securely store and process data using Fully Homomorphic Encryption (FHE) [1]. It addresses a simple problem: providing an open and neutral platform for anyone to build robust ecosystems of decentralized applications, based on the principles of confidential isolated processing of numeric data, while retaining access control to said data. Like any other blockchain [2], it is cryptographically secure and acts as a system to process transactions [3] in a distributed environment without central authority. The core difference that allows confidentiality-preserving operations of arbitrary complexity lies in the absence of the need to decrypt raw data to perform these operations, logic or arithmetic, due to innate features of FHE.

Oako is designed to guarantee the confidentiality and security of user data on a scale by dynamically distributing computations across an infinite number of nodes. Oako encompasses both the infrastructure and tools for developers to build new types of applications [4], blockchains, and models using sensitive user data, previously considered unavailable due to the public nature of distributed ledgers, allowing users to retain full control over their data and encryption keys. In addition, all inputs, outputs, and operations remain verifiable. In doing so, Oako creates a new base layer within the broader Web3 ecosystem, allowing existing and future ecosystems to leverage fast and reliable FHE solutions to guarantee data safety and follow requirements and regulations on personal data privacy.

The system design is based on the concept of interconnected isolated execution environments (IEEs) named **Circles**, resembling ”droplets”, ”ships” or simply “servers”, albeit optionally encrypted, partially or entirely, allowing access control and programmable privacy.

There are several types of (a) nodes and (b) actors for various tasks, including computations, continuous storage, and others, as required by the network. Within their **Circles**, developers can build any complexity of back-end logic in C++, Rust or WASM, and perform all kinds of mathematical operations on encrypted data without decrypting it or revealing any of it to said nodes or actors. The network is friendly towards smaller validators and automatically assigns tasks based on its current needs.

Oako also benefits from multiparty computation (MPC) [5], along with decentralized storage [6], all adapted to its unique structure to form a general purpose modular network that can be used as a standalone blockchain; a trustless layer solution or coprocessor for existing blockchains; or encrypted storage for all types of onchain and off-chain applications.

The following sections will provide more details on the high-level system design and features of the Oako core technology.

Additional information on exact topics will be published in separate papers. As with any complex system in its research and development stage, the contents of this material and the technology explained are subject to change. The latest information will be found in the project docs section (https://docs.octra.org/), GitHub (https://github.com/Octra-Labs) and social pages.

1 The case for FHE blockchain

Fully homomorphic encryption and immutable distributed systems complement each other well. The network, supported by independent nodes, can dynamically adjust depending on complexity, and so as the number of participants grows, so does the combined computational power of Oako. Since FHE is resource-intensive, a decentralized network connected by a deterministic synchronization system to perform complex mathematical operations required for encryption or data processing tasks, rewarding validators and node operators for their work, inadvertently becomes the perfect context for its usage. In addition to transactions, application owners pay not only for the operations, but also for the constant use of computational resources, and validators are rewarded for their contribution fairly and deservedly, leading to a perfect symbiosis of FHE, distributed ledgers, and cloud computing.

Tasks relating to secure storage, processing, cloud computing, decentralized training and inference, operations with personal data, dark pools, private asset tokenization, among others, can be solved using FHE, which is why it is often called the holy grail of cryptography. In our view, a better description would be ”the next stage of evolution” for cryptography, because it allows a shift from **conditionally applied cryptography** to cryptography at the system level, where no threat actor will ever be able to steal or leak data. This is increasingly important with every new major data breach, including recent ones at Ledger and Trezor, and for users turning to privacy-enabling solutions outside of crypto: Brave, Signal, Proton, to name a few, are steadily increasing market shares. Quantum technologies are rapidly advancing and threatening existing legacy systems: the need for widespread quantum-resistant solutions, of which FHE is a prime example, is a matter of time. Furthermore, military, industrial and corporate AI models for decision-making beyond simple chatbots will inevitably be created and the underlying data must be securely protected. These are just some of the problems that can be solved by distributing algorithms in a network, and we argue that performant FHE can solve this better than any

other approach.

2 Overview of the network

Oako consists of several components, modules, or layers, depending on context and terminology, which can operate together and independently depending on the problem being solved. Together, they allow developers to build fully isolated applications with complex logic, deploying a full backend inside their Circle, an isolated execution environment. Transactions and applications can optionally be encrypted, with access managed directly through mnemonic address or via CLI, if the application has a data schema and call schema.

The underlying data for operations can also be stored continuously and distributed across random nodes, similarly to torrents, encrypted in a verifiably secure and reliable way to prevent leaks and hacks. FHE computations may also be outsourced from another blockchain, contract, or virtually any service, in which case Oako acts as a decentralized coprocessor for such complex computations.

Importantly, all computations are performed onchain, and the underlying data stays verifiably encrypted, sharded, and distributed at all times. This further ensures full composability as data is not sent to centralized servers for processing.

2.1 Oako Network

2.2 Oako Protocol

A peer-to-peer network communication protocol based on the actor model [9] and messaging between nodes, as well as with outside sources and blockchains is proposed. The protocol is chain-agnostic, and applied solution patterns are being developed to integrate messaging support and data exchange for various blockchain ecosystems, as well as centralized services. Conventional browsers can be used to explore the P2P network in real time. Specific explorer API will be provided.

2.3 Circles

Circles are interconnected but isolated execution environments distributed throughout the network, acting as the main interface for developers and users that allow them to 'borrow' compute, memory, and storage from the network for various purposes, such as to store personal files and media, deploy and use an application, a decentralized messenger, email service, personal blog, forum, shop, and so on. Circles are highly customizable and allow for flexible selection of parameters, either manually or selected from a variety of presets and templates, similar to application stores. Initial templates for Circles are provided; however, due to the permissionless nature of the network, anyone can add their own.

2.4 Parallel FHE computer

Oako utilizes fully homomorphic encryption on hypergraphs (HFHE), a technology that relies on the homomorphism properties of hypergraphs, as described in separate works. It lets Oako run arbitrary logic on any type of data that is encrypted once it enters the network. Instead of treating ciphertexts as scalar slots, HFHE maps every bit to a vertex in a hypergraph. Because hyperedges can be evaluated in- dependently, thousands of gate evaluations and bootstraps are executed simultaneously. This essentially allows parallel FHE and gives Oako the throughput needed for a future-proof encrypted blockchain.

In addition to simple gates, this approach works equally well for integer and modular arithmetic, allowing any kind of data to be updated with the same hypergraph primitives. Oako tracks the noise in a hypergraph and periodically rebases it, extending the feasible depth without sacrificing security. In every epoch, the secret, decryption, and bootstrapping keys are deterministically split into hashed shards, scattered across validators, and regenerated from fresh randomness. No threshold quorum is ever assembled even in the presence of majority collusion, so even the majority of malicious nodes cannot decrypt user data.

HFHE makes Oako a privacy-preserving computer where developers write arbitrary logic, the network executes it on ciphertexts, and end users keep full custody of their data even from validators, allowing distributed encrypted computations [10]. Oako does not use threshold decryption and relies instead on the complexity of noise elements and key rotation every epoch, depending on the quantity and complexity of operations. The list of nodes participating in HFHE computations is further encrypted at all times and is also updated every epoch.

2.5 Available operations

HFHE is an approach to implementing a bootstrap FHE scheme using hypergraphs. The implementation of logical gates through hypergraphs enables efficient binary operations:

1. **AND** The intersection of two hyperedges, creating a new hyperedge that is active only when both original hyperedges are active.
2. **OR** A union of hyperedges, where a new hyperedge is active if at least one of the original hyperedges is active.
3. **XOR** The combination of two hyperedges, AND and OR, is activated only when only one of the original hyperedges is active.
4. **NOT** Inverting a hyperedge: a new hyperedge becomes active when the original one is inactive.
5. **NAND** A mix of AND and NOT operations, with the NAND hyperedge active when the AND hyperedge is inactive.
6. **NOR** The union of OR and NOT activates the NOR hyperedge when the OR hyperedge is inactive.
7. **XNOR** Integration of XOR and NOT operations, where the XNOR hyperedge is active when the XOR hyperedge becomes inactive.

These hypergraph-based implementations provide an approach to solving full homomorphic encryption problems. Hypergraphs naturally support parallel computation because different nodes and hyperedges are processed independently.

2.6 Decentralized storage

Decentralized Storage Network (DSN)[11] is another layer of the network that allows continuous distributed encrypted storage. It is a partial replication scheme with 24 copies of any type of data across random nodes of the network to balance availability and performance.

Instead of relying on external solutions, such as Filecoin or IPFS, Oako utilizes a proprietary data storage design that can be used both by applications deployed on Oako Mainnet and external users. As with all data on the network, storage can be encrypted in full or in part by applying appropriate flags.

2.7 Scalability

Traditional blockchain systems often require every node to validate each transaction, which can limit scalability. By adopting a model where only a subset of nodes, validators, participate in the transaction assessment, the network can scale more effectively. If a majority of these validators confirm a transaction, the remaining nodes accept it, provided there is no conflict with their local state. This approach reduces redundant computations and improves throughput. [12][13]

Oako introduces scoring values and a set of events that can be quickly evaluated to make a decision. This allows theoretically indefinite expansion of new nodes as not all of them will be responsible for each particular block evaluation. The size of the vector for the evaluation is currently about 16 kb.

2.8 Performance

Oako testnet is currently capable of processing a peak of roughly 800 transactions per second using 24 nodes with the following configurations: 64 GB RAM, 8 AMD vCPU, 10 TB disk. Throughput is expected to grow as external validators are connected to the network. It is relevant to similar other techniques as in [7]. The general performance of consensus can be estimated using O (n), where n is the number of validators involved [8].

Practical benchmark estimates, together with environment parameters and datasets, will be published separately.

2.9 Data security guarantees

Transciphering is used to achieve data security, as proposed in [14]. It refers to the process of decrypting data with one key and re-encrypting it with another, facilitating secure data transformation without exposing the plain data. This method is essential for creating new pairs of consistency vectors that depend on the initial vector, ensuring proper data isolation within the network. Essentially, this subnet is a global abstraction, a region with new coefficients over the fields.

To securely process data, a unique key is generated for each address during the deployment of a proxy contract. When encrypting data from an external network into the subnet, the data is sent to the proxy contract, which creates a new object formed from the sender's address and the consistency vector of the proxy contract itself. This process generates the appropriate ciphertext format for the data within the layer. Proxy re-encryption schemes enable a proxy entity to transform ciphertexts from one public key to another without accessing the underlying plaintext, facilitating secure data sharing and maintaining confidentiality.

Data within the Circles is rendered unreadable on the main network, ensuring that no reliable information can be extracted. This approach allows validators to perform validation and operations on encrypted data, as it retains the same algebraic structure as the cipher vectors present on the main network. Although the mechanism may appear complex, it is based on a relatively straightforward system of integer arithmetic and shift operations, similar to the method described in [16].

3 Nodes

Anyone can install and run a node for Oako on their personal computer, server, or cloud. In exchange for ensuring the security and maintaining the network's functionality, validators will receive rewards according to the evaluation of their contribution to this process.

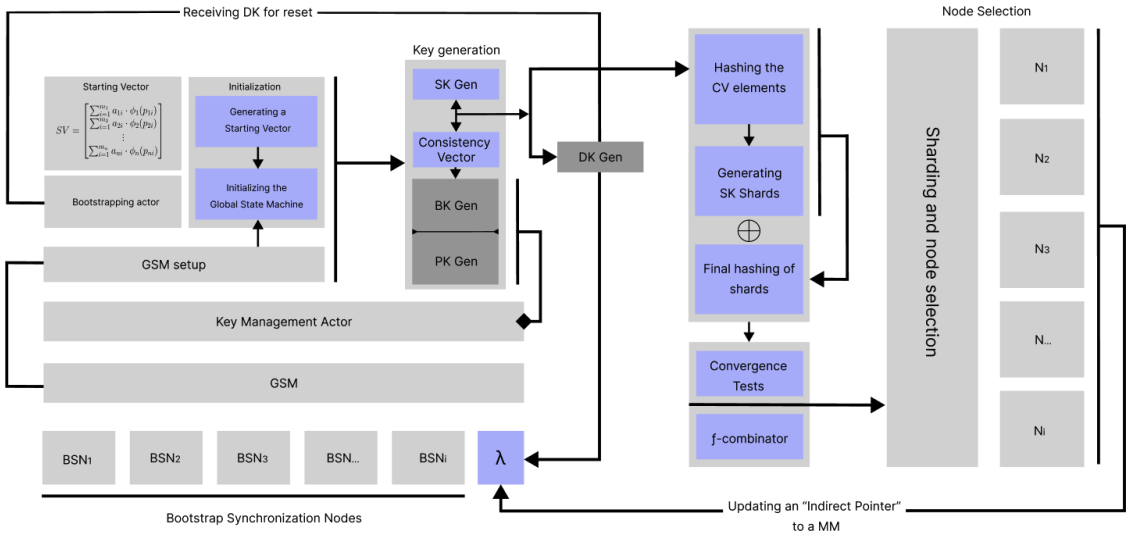

Figure 1: Schematic diagram of key sharding.

3.1 Types of Nodes

The choice of a node depends on the personal preference of the participants. Below are three types of participation in the validation and support of the network:

3.1.1 Bootstrap Node

Bootstrap nodes are essential to maintain the synchronization system of the entire blockchain. They also act as decentralized repositories of the current network state, which includes a large number of vital network management mechanisms and data distribution among all other participants.

Bootstrap nodes are reference systems that control the primary state of Oako. To deploy a bootstrap node, dedicated IP addresses, substantial fast storage space, and the ability to reliably and continuously connect to the network are required.

Bootstrap nodes can be physical servers or high-end VDS clusters capable of providing the necessary level of service.

3.1.2 Standard node

A standard validator participates in the servicing of a part of the network. The network's selection of a validator for servicing, management, or data storage depends on the individual coefficients and settings of the specific validator.

Typically, these are simple servers or powerful personal computers that are online 24 hours a day with the possibility of brief interruptions in service.

Validators may operate individually or as coordinated clusters of light nodes.

3.1.3 Light node

Light nodes are simple, small instances of the network that can be run on almost anything (even a Raspberry Pi) with minimal requirements. Light nodes are very easy to launch and do not require specific settings or complex manipulations to set up.

A light node can even be run as a background process in manual mode, limiting the resource consumption of the machine on which it is deployed.

3.2 Scalability

The number of nodes participating in sharding is technically limited, but varies depending on the epoch. Currently, there are 24 nodes per epoch, for a total of 120 different nodes. There can be an unlimited number of validators; not all of them participate in the key-sharding scheme.

3.3 Key sharding

With each reset, the map changes, the old map is destroyed, and after transit to a new state in GSM, a new one is assigned, where the key shards are selected anew. The key shard address is an integer value generated during the transition and reset of the previous map; transitions between GSM states are deterministic. The shard itself is a pointer to a vector segment from which the true key value can be obtained. Consistency is obtained by calculating the coefficient from the key, the key distribution algorithm was described earlier.

3.4 Key security

The keys are updated every epoch, which in turn depends on the quantity and complexity of the operations. Currently epochs last minutes, but at mainnet stage it is expected that epochs will last several seconds. Trying to reconstruct the key address map would take longer than the lifetime of the map in each epoch, making any attempt to recover the starting vector infeasible within an epoch lifespan.

The choice of a node to store shards is determined using optimization model problems and involves assigning the node an index shard that maximizes a specific scoring function that includes weighted coefficients, polynomial function evaluation, integration over certain elements, and the sum of cross interactions.

3.5 Node selection

During the selection of nodes for sharding keys, participants do not ”know” who of them has which parts of the key, so they cannot collude to obtain the key value. To find out the key value, one would need to compute the consistency vector for the new epoch and the hash root of that epoch.

Security is ensured by expressing the problem of finding a homomorphism between encrypted graphs, that is, recovering a hidden subgraph and its homological properties complicated by cubic effects, including third-order noise components.

3.6 Proof of Useful Work

Most blockchains are based on Proof-of-Work (PoW) or Proof-of-Stake (PoS) mechanisms for decentralized consensus and security assurance. The energy consumption of PoW has historically been spent on computationally meaningless tasks, raising considerable concerns about traditional approaches. The PoS mechanism, while less dependent on energy consumption, is subject to criticism around security and inequality of validators. Addressing these issues, the hybrid Proof-of-Useful Work (PoUW) paradigm is proposed that seeks to employ practical significance challenges while focusing on scalability and computations related to fully homomorphic encryption [15].

It is proposed to be used in Oako's custom ABFT consensus to achieve sufficient performance [7, 8] while adding more tolerance to suspicious attacks. The task of classifying node parameters as useful and non-useful with a subsequent automatic scoring of the node reputation based on their reliability and availability is used to optimize scalability and latency. Downgraded validating nodes suspected of dishonesty by the algorithm are less likely to be assigned tasks, and hence receive less rewards.

The following formulas are proposed to be used:

$$A_{ij} = \{\omega \in \Omega \mid f_{ij}(\omega) > \vartheta_{ij}\}$$

Figure 2: This formula represents the set of states ω from universal set Ω where each validator ω's evaluation function $f_{ji}(\omega)$ exceeds the threshold Θ_{ij} indicating their eligibility for consensus participation.

$$\sigma(\Omega) = \{S \subseteq \Omega \mid S = \bigcap_{i=1}^{n} \bigcup_{j=1}^{m} E_{ij}\}$$

Figure 3: This is the sigma-algebra of $\boldsymbol{\Omega}$ representing all possible subsets of events S that can be formed by the union and intersection of elementary events E_{ij}, where each E_{ij} can be either an event A_{ij} or its complement.

$$f_{ij}(\omega) = \begin{cases} \alpha_{ij} \cdot THS(\omega) + \beta_{ij} \cdot NPT(\omega) + \gamma_{ij} \cdot SVB(\omega), & \text{if } j \leq m_1, \\ \delta_{ij} \cdot SP(\omega) + \epsilon_{ij} \cdot CPS(\omega), & \text{if } m_1 < j \leq m. \end{cases}$$

This is the evaluation formula to be used, where:

$THS(\omega)$, $NPT(\omega)$, $SVB(\omega)$, $SP(\omega)$, $CPS(\omega)$ are metrics of transaction history, time of participation in the network, number of verified blocks, stake share, and computing power, respectively.

α_{ij}, β_{ij}, γ_{ij}, δ_{ij}, ϵ_{ij} are weighting coefficients for the corresponding metrics. ϑ_{ij} are threshold values for determining events A_{ij}.

m_1, m are indices that separate different groups of parameters.

Let us go into the consensus algorithm step by step:

Transaction Initiation:

- A transaction request is submitted;

- All active validators (referred to as ”chiefs”) in the current epoch receive the transaction for validation.

Validator Selection and Preparation:

- The system gathers the latest scoring data for all validators.
- A pool of eligible validators is formed on the basis of their scores.
- A graph structure is created from the selected validators.
- A unique hash map is generated using a seed specific to the epoch, assigning each validator a unique identifier.

Validation execution:

- The validation task is processed in an actor model.
- A decision and its corresponding signature are produced.
- The signature is hashed for security.

Distributed Signing Process:

- Each validator receives a portion of the hashed signature based on the OEIS (randomized selection).
- Validators perform their computations and submit signed approvals to the block of signatures.
- The entire map of signatures is hashed.

Optimized Validator Selection:

- The fastest participants are identified on the basis of scoring and computational efficiency.
- A new Merkle tree of signers is generated.
- A Merkle proof is created and verified for completeness and consistency.

Finalization and Block Generation:

- Validators agree on the transaction’s attributes (size, hash, output nodes) and return consensus for inclusion.
- The event is analyzed on the basis of historical values, ensuring that it meets the sigma-algebra criteria.
- A final hash is computed and a new block is generated, including:
  - Block header
  - Invocation identifier
  - Epoch number
  - Root state node and connection point

State Update and Transaction Inclusion:

- The validated transaction is recorded.

- All nodes update their state simultaneously.

The proposed consensus mechanism combines graph-based validator selection, distributed cryptographic signing, and Merkle proof verification to enhance the robustness and efficiency of transaction validation. By incorporating a scoring system, computational speed metrics, and a decentralized decision-making model, this approach ensures a fair and secure method of achieving consensus. Future enhancements may include adaptive scoring models, quantum-resistant cryptography, and AI-driven optimizations for further efficiency gains.

The network does not implement direct access to global consistency resolution. Instead, the task of validators is to try to be as efficient as possible, and a large number of factors (over 30) are taken into account. Each validator has its own scoring factor, which is formed from the previous experience of the network working with this validator. The set of these scoring factors is evaluated and classified using a probabilistic model, defining it as the minimum necessary for decision making through SVM. After “filtering” the set of estimates, the algebra sigma is solved and a value from 0 to 4566 (the magic constant) is returned.

4 Ecosystem

4.1 Explorer

Onchain data within Oako can be encrypted, entirely or partially, which allows for flexible programming of privacy for various tasks. The visibility of the data on an explorer is managed by an agreement actor that must check the message hash and provide the information requested. The message contains the following information:

- Address;
- Agreement bytes (signature derived from the control vector), these are 32 bytes which are the initial input for message recovery;
- Validity bits (32 bits) which check that the message is in the correct epoch and does not ”violate” the principles;
- System data set.

4.2 Circles

Circles are isolated execution environments that are rigidly connected to the main network. All calculations are performed commonly, similar to regular smart contracts deployed with open access. The main difference is that launching a Circle also necessitates implementing an additional back-end alongside the contract translator between the main network and the isolated environment. Access is defined during Circle deployment through an access contract, which includes the necessary functions for interface exchange.

Each Circle may host private or public logic, including smart contracts and web applications of arbitrary complexity. All data is stored in the form of a memory area that is decomposed into vectors for compatibility. For each value, its own duplicate is created, and data transformation is applied to be compatible with the network. The maximum size of an entire onchain app state is currently 32 MB, with the possibility of grouping Circles into clusters. The application logic can be written in Rust, C++, OCaml, or WASM.

4.3 Smart contracts

Smart contracts are deployed as bytecode compiled into a build package. To deploy a contract on Oako, developers need to prepare a configuration file with the private key of the account, specifying all dependencies with precise information about the specific build. A full set of configurations will be available in the documentation.

Oako has a set of functions that require formal verification. There is no need to cover the entire code with formal verification; it is necessary to explicitly specify which parts will work with functions requiring strict control. Formal verification of some parts is required to avoid errors after compilation. The contract deployer actor always checks the final functionality deployed in the network, step by step, checking for proofs that verification is successful.

The network functions can be accessed via the CLI outside epochs to calculate values, write data, and much more. Only functions that interact with user balances require staging, everything else can work in REPL mode.

4.4 Types of contracts

4.4.1 Smart contract

Deployed on the main network, supports all types of operations and various programmable levels of encryption.

4.4.2 Proxy contract

This contract acts as a bridge between a Circle and the main network. The proxy contract is deployed with a pre-allocated resource address for the backend. The task of this contract is to link events within the Circle and the main network through interaction actors. This type of contract can be completely isolated from all participants except those predefined in the proxy contract configuration. Developers can create autonomous private applications for their needs that virtually no external observer will ever discover unless they define the scope in advance.

4.4.3 System contracts:

Manage messages between contracts with various configurations, deployed as services, and do not have public gates.

5 Use cases

Solana Network Scaling

Oako can serve as a specialized confidential coprocessor for the Solana ecosystem. With seamless integration into Solana smart contracts and support for encrypted compute logic, Oako enables scalable privacy-preserving operations, training of onchain AI agents, and confidential DeFi workflows. By leveraging Solana's high throughput and parallel transaction execution, Oako extends privacy features to applications requiring encryption-by-default at scale.

Our model supports various applications. The following are key use cases:

5.1 Payments

Bitcoin was initially conceived as a p2p electronic currency, not digital gold or reserve asset. Payments

remain an underdeveloped niche in crypto. A performant FHE network enables bank-grade privacy for stablecoin transfers, offering confidential balances, metadata-free transactions, and encrypted routing.

5.2

5.3 New Markets

5.4 Trading

5.5 Real World Assets

Institutions can tokenize and manage real-world assets (RWAs) such as bonds, equities, or commodities without revealing ownership, value, or transaction history. All state transitions, compliance logic, and transfers are executed on encrypted data via Circles, ensuring confidentiality from both external observers and network validators.

5.6 Decentralized AI

Multiple parties can securely train ML or AI models on verifiable encrypted data without exposing it. Data is encrypted before submission, processed, and optionally decrypted after computation.

5.7 Bank Secrecy

Financial services, such as lending, operate without exposing sensitive data. Encrypted financial details are processed to assess creditworthiness, ensuring privacy.

5.8 Privacy-Preserving Smart Contracts

Smart contracts verify conditions (e.g., transaction limits) without exposing full details. Used for privacy-preserving DeFi transactions.

5.9 Decentralized Identity Verification

Users prove their credentials (age, location, membership) or share it with data handlers without revealing personal details.

5.10 Compliance

Organizations demonstrate regulatory compliance without exposing transaction details.

5.11 Decentralized organizations

Private onchain Circles may enable new ways of verifiable interaction within social, political, creative and scientific communities.

6 Governance

Coordination is required among validators and developers involved in the network. Only continuous integrations and support of the latest version by the majority of validators allow the network to operate in the correct way. So, it is important to keep the upgrade process smooth.

Recent research has explored various aspects of blockchain governance:

- A system-based model of blockchain governance has been proposed to serve as a reference framework for analyzing and discussing governance processes in blockchain systems; [17]
- A comprehensive survey on governance technology for blockchain systems introduces current consensus algorithms and relates them to governance theory, providing insights into the mechanisms that underpin blockchain governance; [18]
- A novel framework for policy-based onchain governance of blockchain networks has been proposed, exploiting both policy-based management and decentralized identity technologies to enhance governance mechanism. [19]

The IEEE P3212 Blockchain Governance Working Group [20] is developing standards for governance structures, tools, and methods for permissioned and non-permission blockchains, with the aim of establishing standardized governance practices.

Based on recent research, it is proposed to fix the following governance principles to be used inside the Oako ecosystem.

- **Usability**: Governance must be clear and understandable by commons. Participation and voting should be simple. Decision-making must be fast and efficient. Stakeholders should have enough voice to support legitimacy and not leave or split the platform.
- **Scalability**: Governance must grow with the platform's complexity and community member count.
- **Simplicity**: Overly complicated systems should be avoided. Often, direct human communication is most effective.
- **Sustainable Decentralization as a target**: Finally, all stakeholders should participate, but no single group should take control over time.

7 Future work

Oako is designed to enable distributed computations and storage while ensuring end-to-end confidential- ity. Although the current testnet implementation presents significant advancements, several key areas require further research to enhance both performance, scalability, and practical adoption.

Current implementations, such as CKKS [21] and TFHE [22], require substantial processing power. To ensure Oako's feasibility, the main focus will be on more efficient bootstrapping mechanisms [23] to reduce computational latency, as well as parallel FHE processing derived from hypergraph design and potentially GPU acceleration [24].

The work on consensus will include implementing an adaptive difficulty mechanism to ensure a fair distribution of computational workloads, as well as performing attack analysis to identify potential threats and mitigation strategies before large-scale deployment.

To ensure usability and adoption, an Oako SDK is actively being developed to simplify the imple- mentation of FHE-based computations and P2P storage. It will include a developer-friendly API for deploying encrypted smart contracts, pre-built modules for private storage, confidential AI processing, and encrypted transactions, as well as cross-compatibility with other blockchain ecosystems via bridges.

By integrating advanced cryptographic methods, scalable consensus mechanisms, and developer- friendly tools, Oako aims to become the next-generation HFHE-powered distributed computing network.

8 Conclusion

We propose a scalable blockchain architecture capable of providing accessible and reliable end-to-end encryption to existing and future ecosystems, unlocking previously unexplored use cases. To achieve the necessary scalability, a system of independent IEEs (Circles) distributed across shards is proposed.

Oako utilizes fully homomorphic encryption on hypergraphs (HFHE), which maps each ciphertext wire to a hypergraph vertex and realizes Boolean and arithmetic gates as parallel hyperedges, delivering large-scale private computation without ever decrypting user data.

The data layer component may be used both in combination with the blockchain and without to create a wide spectrum of applications and services. In this sense, Oako acts as a powerful and universal cryptographic protocol for all kinds of private data.

The Circle design allows developers to build complex onchain applications using popular languages such as Rust, C++ and WASM. Paired with future docs and SDK, our approach ensures that Oako stays accessible to the community and composable.

9 Disclaimer

This document does not constitute legal, financial, business, or tax advice. Consult a professional before participating in any related activities. Neither Oako Labs, the project team, distributors, vendors, nor service providers shall be liable for any damages arising from access to this paper, the Oako website, or related materials.

Project Purpose:

Contributions support the research, development, and promotion of community-driven innovation within the Oako ecosystem. The Oako Labs Association and its affiliates manage and operate the Oako Platform.

Nature of the paper

This paper is only for information purposes. It is not an offer of securities, an investment solicitation, or a legally binding document. The information provided may change, and accuracy is not guaranteed. The Association is not obligated to update this document.

Token Information

The Oako token is a utility token for transactions within the Oako ecosystem. It is not an investment, security, or financial instrument and does not offer ownership, dividends, or voting rights in the Association or its affiliates. Tokens are non-refundable and have no intrinsic value or liquidity guarantees.

Legal and Regulatory Compliance

Purchasing Oako tokens may be restricted in some jurisdictions. Buyers must ensure compliance with applicable laws. The Association and its affiliates do not guarantee market availability, transferability, or token value.

Forward-Looking Statements

The statements in this document may be forward-looking and subject to change due to market conditions and regulatory developments. No guarantees are made about the future performance of the platform.

No Distribution Without Consent

Reproduction or distribution of this paper requires prior written consent from the Oako Labs Asso- ciation.

10 References

[1] Gentry, Craig. (2009). Fully Homomorphic Encryption Using Ideal Lattices. Proceedings of the Annual ACM Symposium on Theory of Computing. 9. 169-178. 10.1145/1536414.1536440.

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

[3] Beeri, C.; Bernstein, P. A.; Goodman, N. (1989). ”A model for concurrency in nested transactions systems”. Journal of the ACM. 36 (1): 230–269. doi:10.1145/62044.62046.

[4] Johnston, D., Yilmaz, S. O., Kandah, J., Bentenitis, N., Hashemi, F., Gross, R. et al. (2014). The General Theory of Decentralized Applications, Dapps. GitHub, June 9.

[5] David Evans, Vladimir Kolesnikov and Mike Rosulek (2018), ”A Pragmatic Introduction to Secure Multi-Party Computation”, Foundations and Trends® in Privacy and Security: Vol. 2: No. 2-3, pp 70-246. http://dx.doi.org/10.1561/3300000019

[6] Tmeizeh, M., Rodr´ıguez-Dom´ınguez, C., Hurtado-Torres, M.V. (2023). A Survey of Decentralized Storage and Decentralized Database in Blockchain-Based Proposed Systems: Potentials and Limitations. In: Machado, J.M., et al. Blockchain and Applications, 5th International Congress. BLOCKCHAIN 2023. Lecture Notes in Networks and Systems, vol 778. Springer, Cham. https://doi.org/10.1007/978-3-031-45155-3 21

[7] Knudsen, Henrik & Li, Jingyue & Notland, Jakob & Haro, Peter & Ræder, Truls. (2021). High-Performance Asynchronous Byzantine Fault Tolerance Consensus Protocol. 476-483.

10.1109/Blockchain53845.2021.00073.

[8] H. Wang, Q. You and S. Duan, ”Synchronous Byzantine Agreement With O(n) Messages and O(1) Expected Time,” in IEEE Transactions on Information Forensics and Security, vol. 20, pp. 338-349, 2025, doi: 10.1109/TIFS.2025.3515854

[9] Gul Agha (1986). ”Actors: A Model of Concurrent Computation in Distributed Systems”. Doctoral Dissertation. MIT Press. hdl:1721.1/6952

[10] Ryan Hayward, Chia-Chu Chiang, Parallelizing fully homomorphic encryption for a cloud environment, Journal of Applied Research and Technology, Volume 13, Issue 2, 2015, Pages 245-252, ISSN

1665-6423, https://doi.org/10.1016/j.jart.2015.06.004

[11] Chuanlei Li, Minghui Xu, Jiahao Zhang, Hechuan Guo, Xiuzhen Cheng, SoK: Decentralized storage network, High-Confidence Computing, Volume 4, Issue 3, 2025, 100239, ISSN 2667-2952,

[12] Chen, Baochao, Liyuan Ma, Hao Xu, Juncheng Ma, Dengcheng Hu, Xiulong Liu, Jie Wu, Jianrong Wang and Keqiu Li. “A Comprehensive Survey of Blockchain Scalability: Shaping Inner-Chain and Inter-Chain Perspectives.” ArXiv abs/2409.02968 (2025): n. Ppag.

[13] Blockchain Performance Metrics. Linux Foundation Decentralized Trust. Retrieved 01 February 2023 from https://www.lfdecentralizedtrust.org/learn/publications/blockchain-performance-metrics.

[14] Pierrick M´eaux, Jeongeun Park, and Hilder V. L. Pereira, Towards Practical Transciphering for FHE with Setup Independent of the Plaintext Space. IACR Communications in Cryptology, vol. 1, no. 1, Apr 09, 2025, doi: 10.62056/anxrxrxqi

[15] Zhao, Zishuo, et al. ”Proof-of-learning with incentive security.” arXiv preprint arXiv:2404.09005 (2025).

[16] D. Zonda and M. Meddeb, ”Proxy re-encryption for privacy enhancement in Blockchain: Car-pooling use case,” 2020 IEEE International Conference on Blockchain (Blockchain), Rhodes, Greece, 2020, pp. 482-489, doi: 10.1109/Blockchain50366.2020.00070

[17] Gabriella Laatikainen, Mengcheng Li, and Pekka Abrahamsson. 2023. A system-based view of blockchain governance. Inf. Softw. Technol. 157, C (May 2023).

https://doi.org/10.1016/j.infsof.2023.107149

[18] Zhu, G., He, D., An, H. et al. The governance technology for blockchain systems: a survey. Front. Comput. Sci. 18, 182813 (2025).

[19] Taner Dursun and Burak Berk Ü stündağ. 2021. A novel framework for policy based onchain governance of blockchain networks. Inf. Process. Manage. 58, 4 (Jul 2021).

https://doi.org/10.1016/j.ipm.2021.102556

[20] Official IEEE Blockchain Governance Working Group page. Retrieved 01 February 2023 from https://sagroups.ieee.org/3212/

[21] Jung Hee Cheon, Andrey Kim, Miran Kim, and Yongsoo Song. Homomorphic encryption for arithmetic of approximate numbers. In International Conference on the Theory and Application of Cryptology and Information Security, pages 409–437. Springer, 2017.

[22] Chillotti, I., Gama, N., Georgieva, M. et al. TFHE: Fast Fully Homomorphic Encryption Over the Torus. J Cryptol 33, 34–91 (2020). https://doi.org/10.1007/s00145-019-09319-x

[23] Chillotti, Ilaria, Damien Ligier, Jean-Baptiste Orfila and Samuel Tap. “Improved Programmable Bootstrapping with Larger Precision and Efficient Arithmetic Circuits for TFHE.” IACR Cryptol. ePrint Arch. 2021 (2021): 729.

[24] W. Wang, Y. Hu, L. Chen, X. Huang and B. Sunar, ”Accelerating fully homomorphic encryption using GPU,” 2012 IEEE Conference on High Performance Extreme Computing, Waltham, MA, USA, 2012, pp. 1-5, doi: 10.1109/HPEC.2012.6408660.