Yellow Paper

Dr Michael Singh
Professor Michele Marchesi
Dr Giuseppe Destefanis
Dr Rumyana Neykova
Engr Anjum Nazir
Dr Julien Lange
Dr Lodovica Marchesi

Contents

1 Introduction **7**
- 1.1 Innovations . . . 9
- 1.2 Paper Outline . . . 10

2 Blockchain Technology & Cross-Border Trade **10**
- 2.1 No Suitable Blockchain Solution Available for Cross-Border Trade . . . 11

3 Components of a Blockchain Consensus Protocol **13**

4 Cobe's Blockchain Operation – Overview **14**
- 4.1 Proof of Stake (PoS) – A Quick Overview . . . 14
- 4.2 Operation of Cobe's CPoS Blockchain . . . 14
  - 4.2.1 Setup Phase . . . 15
  - 4.2.2 Running Phase . . . 16
  - 4.2.3 Reward Distribution Phase . . . 17
  - 4.2.4 Validator Penalization Phase . . . 17

5 Cobe's Proof of Turn (PoT) Consensus Protocol **19**
- 5.1 Working of the Cobe's Proof of Turn (PoT) Consensus Protocol . . . 20
  - 5.1.1 Cobe's Proof of Turn (PoT) Initial Block Schedule Message (IBSM) Generation . . . 20
  - 5.1.2 Cobe's Proof of Turn (PoT) Initial Block Schedule Message (IBSM) Propagation . . . 20
  - 5.1.3 Cobe's Proof of Turn (PoT) Schedule Generation . . . 21
  - 5.1.4 Block Generation . . . 21
  - 5.1.5 Block Propagation . . . 21
  - 5.1.6 Block Validation . . . 22
  - 5.1.7 Block Finalization . . . 23
  - 5.1.8 Incentive Mechanism . . . 23
- 5.2 Algorithm of Cobe's Proof of Turn (PoT) Consensus Protocol . . . 24
- 5.3 Worked Example of the Cobe's Proof of Turn (PoT) Consensus Protocol . . . 26
  - 5.3.1 Mathematical Interpretation . . . 27

6 Cobe's Permissionless CPoS Chain: Block Creation Share (BCS) Calculation **29**
- 6.1 Stake Age (σ_d) . . . 30
- 6.2 Online Age (O) . . . 30

7 Concurrency and Cobe's Parallel Chain Architecture **30**
- 7.1 Concurrent Block Creation . . . 31
  - 7.1.1 DApp-Based Concurrent Fork Chains . . . 31
  - 7.1.2 Load-Aware Concurrent Fork Chains . . . 33
- 7.2 Concurrent Transaction Execution and Verification (CTEV Algorithm) . . . 38

8 Cobe's Permissioned Concurrent Proof of Authority (CPoA) Blockchain **40**
- 8.1 Operation of Cobe's CPOA Blockchain . . . 40
- 8.2 Master Node Eligibility Criteria (E_c) . . . 40
- 8.3 Master Node Onboarding Process . . . 40
- 8.4 Master Nodes Reputation Score (ρ) Calculation . . . 42
  - 8.4.1 Master Nodes' Validation Share Score (VSS) & Validation Share (VS) Calculation . . . 43

8.5 Block Creation Schedule Generation . . . 43
8.6 Cobe's Confidentiality Enhancement Framework (CCEF) . . . 43
8.6.1 Homomorphic Encryption (HE) . . . 43
8.6.2 Zero Knowledge Proof (ZKP) . . . 44
8.6.3 Framework Operation . . . 44
8.6.4 Cobe Homomorphic Encryption Scheme . . . 46

9 Cobe's Dual Blockchain Interoperability Architecture (CDBIA) 47
9.1 Blockchain Interoperability Concepts . . . 48
9.2 Cobe's Dual Blockchain Interoperability Architecture (CDBIA) . . . 48
9.2.1 Components of Cobe's Dual Blockchain Interoperability Architecture (CDBIA) . . . 49
9.2.2 Operation of CDBIA . . . 49
9.3 Interoperability with External Blockchains . . . 51

10 Cobe Smart Language Stack 52
10.1 Overview of the Smart Language Landscape . . . 52
10.2 Overview of the Cobe's Language Stack . . . 53
10.3 Surface Language . . . 54
10.4 Intermediate Representation . . . 54
10.5 Low-level Language . . . 54

11 Cobe Virtual Machine (CVM) and Middleware 54
11.1 Overview of Blockchain Virtual Machines . . . 54
11.2 The Cobe Virtual Machine (CVM) Architecture . . . 55
11.3 Architecture of the Cobe Middleware . . . 56
11.3.1 Deployment Model of Smart Contracts . . . 56
11.3.2 Digitally Signed Contract . . . 57

12 Decentralized Apps (DApps) 57
12.1 DApp Elements . . . 58
12.2 Cobe DApp Architecture . . . 59
12.2.1 App System . . . 61
12.2.2 Terminals and Apps . . . 62
12.2.3 IoT Devices . . . 62
12.3 Frontend and Smart Contract Communication . . . 63
12.4 Cross Chain Decentralized App (CC-DApp) utilizing Cobe's Dual Blockchain Interoperability Architecture . . . 63

13 Oracles 65
13.1 How Oracles Work . . . 65
13.2 Types of Blockchain Oracles . . . 66
13.2.1 Inbound Oracles . . . 66
13.2.2 Outbound Oracles . . . 66

14 Blockchain Security 67
14.1 Brief Overview of Blockchain Attacks . . . 67
14.2 Proof of Turn (PoT) Protocol – Security Considerations . . . 68
14.2.1 Key Security Considerations . . . 68
14.3 Security of Smart Contracts . . . 69
14.4 Cobe's Blockchain: Threshold Signature Scheme (TSS) . . . 69

14.5 Cobe's Blockchain: Quantum-Resilience Strategy . . . 70
14.5.1 Transitioning to Quantum-Resistant Cryptography . . . 70
14.5.2 Cobe's Approach to Quantum-Secure Blockchain . . . 71
14.5.3 Security Against Quantum Threats . . . 71

15 Cobe Blockchain Economics **71**
15.1 Cobe Ecosystem Coin Stack . . . 71
15.2 Transaction Fees . . . 72
15.2.1 Transaction Fee for CPoS Chain . . . 72
15.2.2 Transaction fee for CPoA Chain . . . 72
15.2.3 Stabilized Elastic Fee Model for CPOA Chain . . . 73
15.3 Cobe's Inflation Economics . . . 73
15.3.1 CBE Inflation Schedule (CIS) . . . 74
15.3.2 CBE Coin Burning . . . 75
15.3.3 Staking Rewards (SR) . . . 75

16 Governance **77**
16.1 CPoS Blockchain Governance Structure . . . 78
16.1.1 Types of Stakeholders, Eligibility, and Voting Rights . . . 78
16.2 CPoA Blockchain Governance Structure . . . 80

17 Nucleus Platform **81**
17.1 Nucleus's Cross-Border Trade Platform . . . 81
17.1.1 Nucleus's Cross-border Trade Platform – Key Features . . . 81
17.1.2 Dispute Resolution . . . 82
17.1.3 Nucleus's Cross-border Trade Platform Technology Stack . . . 82
17.1.4 Cross-Border Transaction Architecture . . . 82
17.2 Nucleus's DeFi Platform . . . 85
17.2.1 Cryptocurrency Backed Trade Finance (DeFi) . . . 85
17.2.2 Centralized Trade Finance (CeFi) . . . 85
17.2.3 Nucleus DeFi Platform Technology Stack . . . 86
17.2.4 DeFi Borrowing . . . 87
17.2.5 Minting New Cobe Stable Coins . . . 87
17.2.6 Interest Rates . . . 88
17.2.7 Collateral Loan to Value (LTV) Ratio . . . 89
17.3 Nucleus's Product Authentication Platform . . . 90
17.3.1 Nucleus's Product Authentication Architecture . . . 91
17.3.2 Nucleus Product Authentication Platform Technology Stack . . . 92
17.4 Nucleus APIs . . . 92
17.5 Nucleus User Rating . . . 92
17.6 Nucleus Platform: User Adoption Growth . . . 93

18 Sonic: Cobe's Native Wallet **94**
18.1 Sonic Wallet – Technology Stack . . . 95

19 Cobe Labs **95**

20 The Cobe Foundation **95**

21 Cobe Ecosystem **98**

22 Disclaimer **99**

List of Figures

1 Cobe's dual blockchain architecture. . . . 9
2 Components of a consensus protocol. . . . 14
3 Cobe blockchain operation - one epoch. . . . 15
4 This process entails penalizing validators in the event of violations. . . . 19
5 Setup and schedule generation phases of Cobe's Proof of Turn (PoT) consensus protocol. . . . 22
6 Validators create a new initial block schedule message (IBSM). . . . 26
7 Initial block schedule message (IBSM) integrity check. . . . 26
8 Calculation of Synchronized Global Random Number (SGRN). . . . 27
9 Random Number Generation via CA30. . . . 27
10 Random selection of monitor nodes. . . . 31
11 Monitor nodes send `fork()` message on the network. . . . 32
12 DApps having their own separate fchain. . . . 32
13 DApp-based block schedule for each fchain. . . . 33
14 Synchronization of DApp-based fchains. . . . 33
15 Random selection of monitor nodes. . . . 34
16 Load-aware fork creation. . . . 35
17 Block schedule splitting process. . . . 35
18 Parent and fchains will share the same transaction pool. . . . 36
19 Creation of ftable. . . . 36
20 Transaction assignment process for fchains. . . . 37
21 Transaction assignment process for fchains. . . . 37
22 Fork fchain synchronization. . . . 38
23 Transaction block that contains transactions. . . . 38
24 Construction of separate occurrence net. . . . 39
25 Parallel execution of transactions. . . . 39
26 Cobe's CPoA blockchain operation - one epoch. . . . 41
27 Master node onboarding process. . . . 42
28 Transaction processing flow in CPoA. . . . 46
29 Cobe's Dual Blockchain Interoperability Architecture (CDBIA). . . . 50
30 Cross-chain coin transfer process. . . . 51
31 Cobe language stack. . . . 53
32 Basic design of the Cobe Virtual Machine. . . . 55
33 Mutable memory space inside a virtual machine. . . . 56
34 Cobe Virtual Machine – extended secure middleware design. . . . 57
35 The typical DApp elements. . . . 59
36 Cobe's proposed DApp architecture. . . . 60
37 Frontend and backend communication of a DApp. . . . 64
38 Cobe's Cross Chain-Decentralized App (CC-DApp); SC: smart contract. . . . 65
39 Data flow between smart contact, oracle, and attestation service. . . . 67
40 The hypothetical graph above shows how inflation rate will decrease over time. . . . 75
41 How a node's reputation score affects its staking reward. . . . 77
42 Graphical representation of Eq.15. . . . 79
43 Summarizes the voting process. . . . 80
44 Fiat cross-border transaction architecture. . . . 83
45 Crypto cross-border transaction architecture. . . . 84
46 Nucleus's CeFi trade finance. . . . 86
47 DeFi borrowing process. . . . 87

48 Seller milestone-based borrowing. 87
49 Example of Cobe's interest rate. 89
50 Nucleus's product authentication architecture 91
51 Cobe Labs. 96
52 The Cobe Foundation. 97
53 The Cobe ecosystem. 98

List of Tables

1 Kinds of blockchain and fees. 12
2 Cross-border trade application use cases. 12
3 Main blockchain solutions for cross-border trade. 13
4 Shows the random numbers generated by each validator and the timestamp. . . . 28
5 Arrival of an IBSM at validator 1. 28
6 Cobe ecosystem coin stack. 72
7 Summary of weights assigned to each type of transaction. 73
8 Inflation terminology. 74
9 Nucleus's cross-border transaction vs. Letter of Credit. 82
10 Nucleus's cross-border trade platform technology stack. 82
11 Nucleus DeFi platform technology stack. 86
12 Nucleus's product authentication platform vs. competitors. 91
13 Nucleus product authentication platform technology stack. 92
14 Sonic wallet – technology stack. 95

1 Introduction

Cobe's mission is to create the most comprehensive cross-border trade ecosystem to date, making global trade dramatically cheaper, faster, and more efficient.

By removing the barriers that make cross-border trade a real challenge, Cobe will enable millions of people who are currently excluded from the world economy to participate and find their place.

To effectively meet the demands of cross-border trade, Cobe aims to provide a dual blockchain solution, where both native permissioned and permissionless chains are built to work in synergy. Based on a Concurrent Proof of Stake (CPoS) consensus protocol, the permissionless blockchain is optimized for solutions that require a high degree of transparency, democracy, and participation (e.g., DeFi).

The permissioned blockchain, based on a Concurrent Proof of Authority (CPoA) consensus protocol, is optimized for solutions that require confidentiality, fixed transaction fees, and a high throughput (e.g., supply chain management, product authentication, information notarization).

No other blockchain project provides this type of holistic solution to building cross-border trade applications, where users have the flexibility to choose the best option for their specific needs when creating DApps.

To improve the scalability of the ecosystem and allow high transaction speeds, Cobe has developed a novel consensus protocol called 'Proof of Turn' (PoT). In this consensus protocol, the selection of the validator node of the next block is planned through a 'block schedule', which securely defines the sequence of the subsequent block creators. This removes the unnecessary delay that is encountered by legacy consensus protocols, which only select one block creator at a time at random, creating a significant delay in block throughput. PoT utilizes the computer science concept of Cellular Automata (CA) to generate its universally unique random schedules securely. Each validator in the schedule creates its block following the 'turn method'. Because of this, the protocol's consensus algorithm has been named 'Proof of Turn'. PoT also allows the enforcement of specific constraints to help prevent fraudulent transactions, increasing blockchain security.

To dramatically increase transaction speeds, Cobe has developed a novel set of cutting-edge concurrency protocols. These not only allow the parallel execution of transactions within a single block but also the parallel creation of entire blocks.

For concurrent block creation, Cobe has developed two approaches: 'Load Aware Concurrent Fork Chains' and 'DApp Based Concurrent Fork Chains'.

When the load on the network increases, the 'Load Aware' protocol enables the creation of parallel fork chains (fchains) to process the transactions more rapidly. Each fchain in the network operates in parallel with the main chain and uses the same consensus protocol. However, each fchain creates its own block schedule, made possible via Cobe's PoT consensus protocol. At the end of each round, all the fork chains created are integrated into the main chain.

The DApp Based Concurrent Fork Chains protocol enables forked chains to be created for different DApps operating on Cobe's network, thus reducing the load on the main chain. Furthermore, for each DApp, a distinct parallel fchain can be created, which integrates into the main chain after each round.

Alongside its block creation protocols, Cobe has also developed a novel Concurrent Transaction Execution and Verification (CTEV) protocol. This protocol utilizes cutting-edge techniques that enable transactions within a single block to be executed in parallel, significantly boosting the speed at which they are processed.

The introduction of the novel PoT consensus protocol and the cutting-edge use of concurrency make Cobe one of the most powerful blockchains to date.

Cobe is also launching its very own native cross-border trade platform, named Nucleus. This

consists of a suite of applications built on top of its blockchain.

Nucleus will target the three fundamental challenges that pose major obstacles to cross-border trade. They are:

- **Trust:** By providing a powerful alternative to Letters of Credit (LC), Nucleus will give millions of individuals currently unable to participate in global trade the ability to do so. Nucleus's smart escrow facility takes minutes to set up, as opposed to the weeks/months that a typical LC requires, all at around 1/10 of the cost.
- **Finance:** Nucleus will provide easy and low-cost access to Decentralized Finance (DeFi) for cross-border trade. With the ability to synergize lending with individual cross-border trade transactions, and its unique user rating system, Nucleus will be able to offer both buyers and sellers the lowest cost borrowing.
- **Product Authentication:** Nucleus will take the most comprehensive approach to authentication, including document validation, track & trace, and provenance.

For those looking to build cross-border trade DApps on Cobe's blockchain, the Nucleus platform provides significant advantages. These include:

- **Integration with Nucleus APIs:** DApp developers on Cobe's blockchain will be able to utilize its native platform Nucleus's smart escrow, DeFi, CeFi, and product authentication APIs to build more comprehensive applications. No other blockchain provides such a resource.
- **Instant Access to Nucleus Platform's User Base:** Developers launching DApps on Cobe's blockchain will have instant access to the large network of users on the Nucleus platform, enabling them to start scaling immediately. No other blockchain provides such direct access to a network of potential users of cross-border trade applications, giving Cobe a distinct adoption advantage in this space over alternative blockchains.

On the top of this, Cobe will look to form partnerships with key organizations involved in cross-border trade. This includes forming partnerships with global trade inspection companies, fair trade organizations, logistics providers, document notarization agencies, phytosanitary and food certification agencies, trade finance lenders, and many more. Cobe's core team will leverage these partnerships to create APIs, which its community can then utilize to create DApps per their requirements.

Figure 1 below shows Cobe's dual blockchain architecture and the Nucleus platform. This includes its permissioned and permissionless blockchains, interconnected via its Relay system, which will execute its Cross Communication Blockchain (CCB) protocol. The diagram also shows how DApp developers building applications on Cobe's blockchain will be able to utilize its Nucleus platform's cross-border trade APIs to build more comprehensive applications.

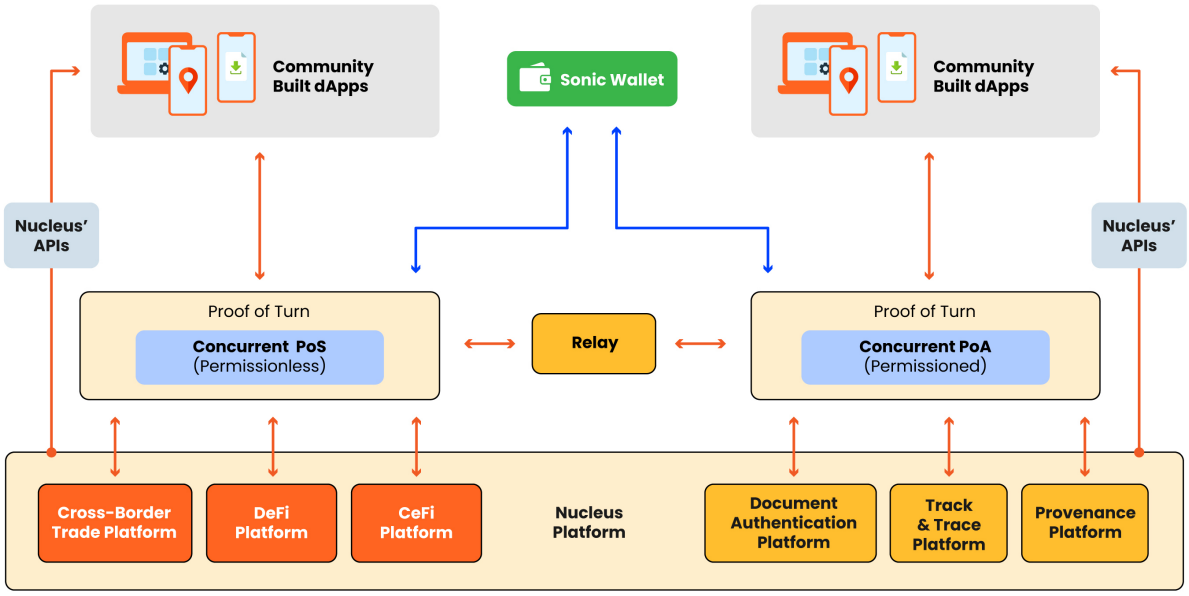

Figure 1: Cobe's dual blockchain architecture.

1.1 Innovations

Below is an outline of Cobe's key innovations:

- Dual-sided blockchain architecture, which includes both permissioned and permissionless chains working in synergy. This provides DApp developers with complete flexibility to choose between a chain with a high level of confidentiality and a chain with a high degree of transparency and decentralization. No other blockchain offers such a comprehensive solution.
- Fixed and variable transaction fee options. Some cross-border trade applications should be built on a blockchain with fixed transaction fees if they are to eliminate uncertainty, which Cobe's Concurrent Proof of Authority (CPoA) blockchain provides. On the other hand, other applications are better on a chain with a variable fee structure, which is offered by Cobe's Concurrent Proof of Stake (CPoS) chain. No other blockchain provides developers with such flexibility when creating cross-border trade applications.
- The development of a novel consensus protocol, 'Proof of Turn' (PoT), to increase transaction throughput.
- The development of three cutting edge concurrency protocols to further boost transaction speeds:
  - The Load Aware concurrency protocol for concurrent block creation, enabling parallel fork chains (fchains) to be created when the demand on the network increases.
  - The DApp Based Concurrent Fork Chains protocol, enabling fchains to be created for different DApps, which then integrate into the main chain after each round, reducing the load on the main chain.
  - The CTEV protocol, which enables transactions within a single block to be executed in parallel, boosting the speed at which they are processed.
- The Nucleus trade platform, built on Cobe's blockchain network, to make cross-border trade dramatically cheaper, faster, and more efficient.

- In addition to providing cross-border trade services to buyers and sellers directly involved in cross-border trade, Nucleus also offers the following unique advantages for DApp developers who build their applications on Cobe's blockchain:

  - Instant access to Nucleus's user base, enabling DApp developers to access a large network of potential users for their own cross-border trade applications, giving them the ability to start scaling immediately. No other blockchain provides such direct access to potential users.
  - Integrations with Nucleus APIs. DApp developers on Cobe's blockchain will be able to utilize and integrate Nucleus's smart escrow, DeFi, CeFi, and product authentication APIs to build more comprehensive applications. No other blockchain enables this.
  - Cobe will leverage partnerships with key organizations involved in cross-border trade to create APIs, which its community can then utilize to create DApps as per their requirements.
  - Cobe Stable Coin (CBS), which is pegged at a 1:1 ratio to the USD to avoid volatility issues.

1.2 Paper Outline

The goal of this paper is to describe Cobe's entire ecosystem, including its blockchain solution, Nucleus platform, Sonic wallet, and novel consensus protocols. The remainder of the paper is organized as follows. Section 2 describes the current state of blockchain technology in the context of cross-border trade. Section 3 presents an overview of the components of blockchain consensus protocols. Section 4 presents a complete overview of how the Cobe blockchain will operate, diving into the activity of block generation. Section 5 describes in detail the novel Proof of Turn (PoT) consensus protocol, developed by Cobe to increase transaction throughput. Section 6 introduces Cobe's permissionless blockchain, which utilizes Cobe's Proof of Stake (CPoS) consensus protocol. Section 7 describes Cobe's novel concurrency protocols. Section 8 introduces Cobe's permissioned blockchain, which utilizes its Concurrent Proof of Authority (CPoA) consensus protocol. Section 9 explains how the interoperability between the two blockchains, permissioned and permissionless, works; moreover, it shows the possibility of interoperating with external blockchains. Section 10 defines the Cobe Smart Contract (SC) language stack, based on Solidity and optimized for the Cobe Virtual Machine. Section 11 describes the architecture of Cobe's Virtual Machine and provides a high-level view of the deployment model of Smart Contract bytecode.

Section 12 describes how Cobe's blockchain network will operate. Section 13 briefly discusses Oracles, used to access information from an source external to Cobe's blockchain network. Section 14 explores blockchain security issues and explains how Cobe deals with them. Section 15 presents both the fixed and variable blockchain transaction fee models developed Cobe. Section 16 describes Cobe's native governance protocols. Section 17 describes Cobe's Nucleus platform. Section 18 describes Cobe's native wallet, called Sonic. Section 19 describes Cobe Labs, a platform where its research results will be regularly published. Section 20 covers the Cobe Foundation. And finally, Section 21 includes a diagram that summarizes the entire Cobe ecosystem.

2 Blockchain Technology & Cross-Border Trade

In this section we will provide an overview of the current state of blockchain technology in relation to cross-border trade.

2.1 No Suitable Blockchain Solution Available for Cross-Border Trade

When it comes to building cross-border trade applications, permissionless chains are better for generating transparency and lowering entry barriers to maximize participation, while permissioned chains retain the most confidentiality.

On top of this, some cross-border trade applications are best built on a blockchain with fixed transaction fees so that businesses can better predict their costs. Picture a product authentication application that's integrated into a business supply chain, processing a high volume of daily microtransactions. If the blockchain's transaction fees fluctuate by even a small amount, the business's day-to-day costs could change substantially. Naturally, most businesses would be uncomfortable with such uncertainty – making fixed transaction fees an essential prerequisite for adoption of such an application.

Table 1 highlights the preferable feature based on use cases, while Table 3 summarizes the main blockchains that are currently available to decentralized cross-border trade application developers.

As shown in Table 3, no blockchain besides Cobe offers the complete set of features that are essential for creating a comprehensive cross-border trade ecosystem, which are:

- **Total Flexibility Between Permissioned & Permissionless Blockchain Options:** to create an effective cross-border trade ecosystem, DApp developers need a facility that allows them to build parts of their application on a permissioned chain and others on a permissionless chain. For example, a product authentication DApp may require certain transactions to be permissioned to retain confidentiality, while other transactions need to be permissionless for maximum transparency.

- **Total Flexibility Between Fixed and Variable Transaction Fee Options:** despite it being an essential requirement for a blockchain solution that caters to the needs of all types of cross-border trade applications, this is unavailable at present.

- **Cross-Border Trade APIs:** these can be of huge help to developers looking to create effective cross-border trade applications. They include smart escrow, decentralized finance, product authentication, and provenance APIs. At present, no blockchain provides a robust set of cross-border trade APIs to its community.

Currently, there is no comprehensive decentralized cross-border trade ecosystem that effectively addresses the needs of both buyers and sellers when trading across borders.

Numerous blockchain technologies have been (or are being) developed in relation to smart contracts, securing payments, decentralizing finance, product authentication, and streamlining supply chains. However, these blockchain solutions and DApps are fragmented rather than being integrated into a cohesive ecosystem. By changing this, Cobe aims to dramatically improve the ability of businesses – even those situated in the most underdeveloped regions of the world – to conduct international trade with greater security and ease.

The Table 2 provides a few examples of different cross-border trade application use cases where a permissioned, permissionless, or hybrid blockchain is more likely to be the preferred option. Note that this list is not exhaustive, but it serves to provide the reader with a few practical examples.

Table 1: Kinds of blockchain and fees.

Feature	**Use Case**
Permissioned Chain	Required when high levels of confidentiality are a prerequisite. For example, government or supply chain applications where transaction details should only be disclosed to those authorized.
Permissionless Chain	Where low barriers to entry for higher participation and transparency are high priority.
Fixed Transaction Fees	Required when certainty in cost is a prerequisite. For example, product authentication applications that are integrated into a business's supply chain. Most businesses in this situation would prefer fixed fees over those that fluctuate on a day-to-day basis, leading to uncertainty.
Variable Transaction Fees	Preferred when retaining a high level of decentralization is a top priority of the network. Transaction fees in this instance are determined by demand on the network rather than being fixed manually.

Table 2: Cross-border trade application use cases.

Permissionless Blockchain Use Cases	**Permissioned Blockchain Use Cases**	**Hybrid Blockchain Use Cases**
Decentralized Trade Finance (Defi)	Supply Chain Applications	Cross-Border Transaction Applications
NGOs	Logistics Applications	Product Authentication Applications
Charities & Donations	Digital Identity and Document Verification Applications	Provenance Applications
Fair Trade Applications	Banks & Financial Institutions	Institutional Bodies
Open-Source Applications	Pharmaceutical and Healthcare Applications.	Oracles

Table 3: Main blockchain solutions for cross-border trade.

Blockchain	**Type**	**Transaction Cost**	**Primary Focus on Cross-Border Trade**	**Cross-Border Trade APIs Provided by Blockchain**
Ethereum	Permissionless	Variable	No	No
Cardano	Permissionless	Variable	No	No
Solana	Permissionless	Variable	No	No
Elrond	Permissionless	Variable	No	No
Algorand	Permissionless	Variable	No	No
Vechain	Permissioned	Variable	No	No
Stellar	Permissioned	Variable	No	No
Eosio	Permissioned	No Fee	No	No
R3 Corda	Permissioned	Fixed	No	No
Cobe	Permissioned, Permissionless & Hybrid Options	Fixed & Variable Options Available	Yes	Yes

3 Components of a Blockchain Consensus Protocol

In this section, we present a brief overview of the key components of blockchain consensus protocols. Blockchain networks are composed of nodes, which can be categorized into different types, such as (i) lightweight or simple nodes, (ii) full nodes, (iii) validator nodes, and (iv) archive nodes. Lightweight or simple nodes have limited capabilities, typically only sending and receiving transactions without hosting a full copy of the blockchain ledger. Full nodes play a crucial role in blockchain networks. They download and maintain a complete copy of the blockchain ledger. This enables them to independently verify and validate each block and transaction according to the network's consensus rules. In Proof of Stake (PoS) blockchain networks, validator nodes are responsible for creating new blocks and validating transactions. Validation is the process of adding valid transactions into a block and sharing it across the network. Archive nodes are a specialized type of full node. While full nodes can prune old data they no longer need, archive nodes retain the entire history of the blockchain ledger.

Since blockchain is a decentralized network, the process of selecting the node that will create the next block can be complex. Consensus enables the network to make this decision. The goal of a blockchain consensus protocol is to ensure that all participating nodes agree on a common transaction history, which is stored in the form of a distributed ledger. Yang Xiao et. al. [1] identified five different components of a consensus protocol, which are shown in Figure 2 and discussed below.

- **Block Proposal:** During this phase, validators create new blocks and sign them with their private keys.
- **Information Propagation:** This phase is responsible for the propagation of new blocks across the network.
- **Block Validation:** In this phase, newly created blocks are validated.

- **Block Finalization:** In this phase, validated blocks are permanently added to the blockchain.
- **Incentive Mechanism:** In this phase, the block creation reward is distributed among the validators.

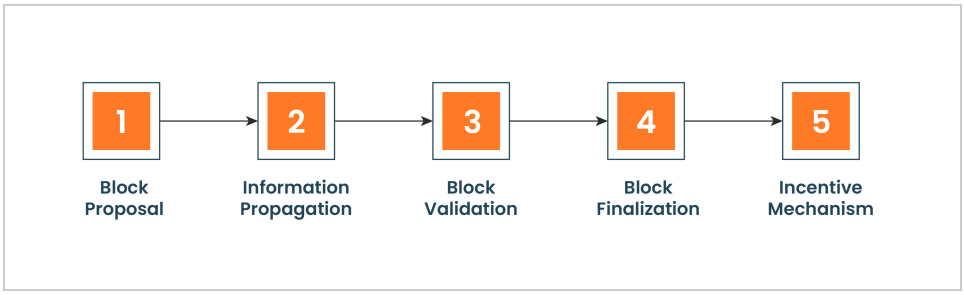

Figure 2: Components of a consensus protocol.

4 Cobe's Blockchain Operation – Overview

Cobe's permissionless blockchain is based on Concurrent Proof of Stake (CPoS) consensus protocol and Cobe's permissioned blockchain is based on Concurrent Proof of Authority (CPoA) consensus protocol.

4.1 Proof of Stake (PoS) – A Quick Overview

In Proof of Stake (PoS) consensus mechanism, validators lock up a certain amount of cryptocurrency (coins) as a stake to become eligible to validate new blocks. The selection process for validators is typically randomized but weighted by the size of the stake, giving those with a higher stake a better chance of being chosen. Once selected, the validator proposes a new block to be added to the blockchain. The proposed block is then checked for validity by other validators, ensuring that it adheres to the protocol rules and does not contain any fraudulent transactions.

The consensus process in PoS involves validators reaching agreement on the validity of the proposed block. Once consensus is achieved, the block is added to the blockchain, and the validator who proposed the block is rewarded with transaction fees and/or newly minted tokens. Validators face penalties, such as slashing, for misbehavior like proposing invalid blocks or being offline during their turn. Governance in PoS systems can be either on-chain or off-chain, allowing validators and stakeholders to vote on protocol changes and updates, ensuring that the network remains secure, efficient, and adaptable to new challenges and improvements.

4.2 Operation of Cobe's CPoS Blockchain

In blockchain terminology, an epoch refers to one complete time cycle (duration), which is used to align and execute different tasks or phases. In the Cobe blockchain, an epoch comprises four phases: (i) a setup phase, (ii) a running phase, (iii) a reward distribution phase, and (iv) a validator penalization phase. The setup phase and the validator penalization phase are executed only once per epoch. During the setup phase, tasks related to blockchain operations are performed, such as checking nodes' eligibility to become validator, validators selection, Block Creation Share (BCS) calculation for each validator. The running phase consists of several rounds, and in each round, two main tasks are performed: (i) generation of block creation schedule, and (ii) blocks generation. Each validator creates a block schedule using Cobe's Proof

of Turn (PoT) consensus algorithm. In the reward distribution phase, rewards are distributed among the validators that participated in that epoch. After the end of each epoch, validators may be subject to slashing for exhibiting bad behavior. This punitive measure ensures network integrity, efficiency, and security. This concept is illustrated in Figure 3. The details of the setup phase, running phase, reward distribution phase, and validator penalization phase are presented in Section 4.2.1, Section 4.2.2, Section 4.2.3, and Section 4.2.4 respectively.

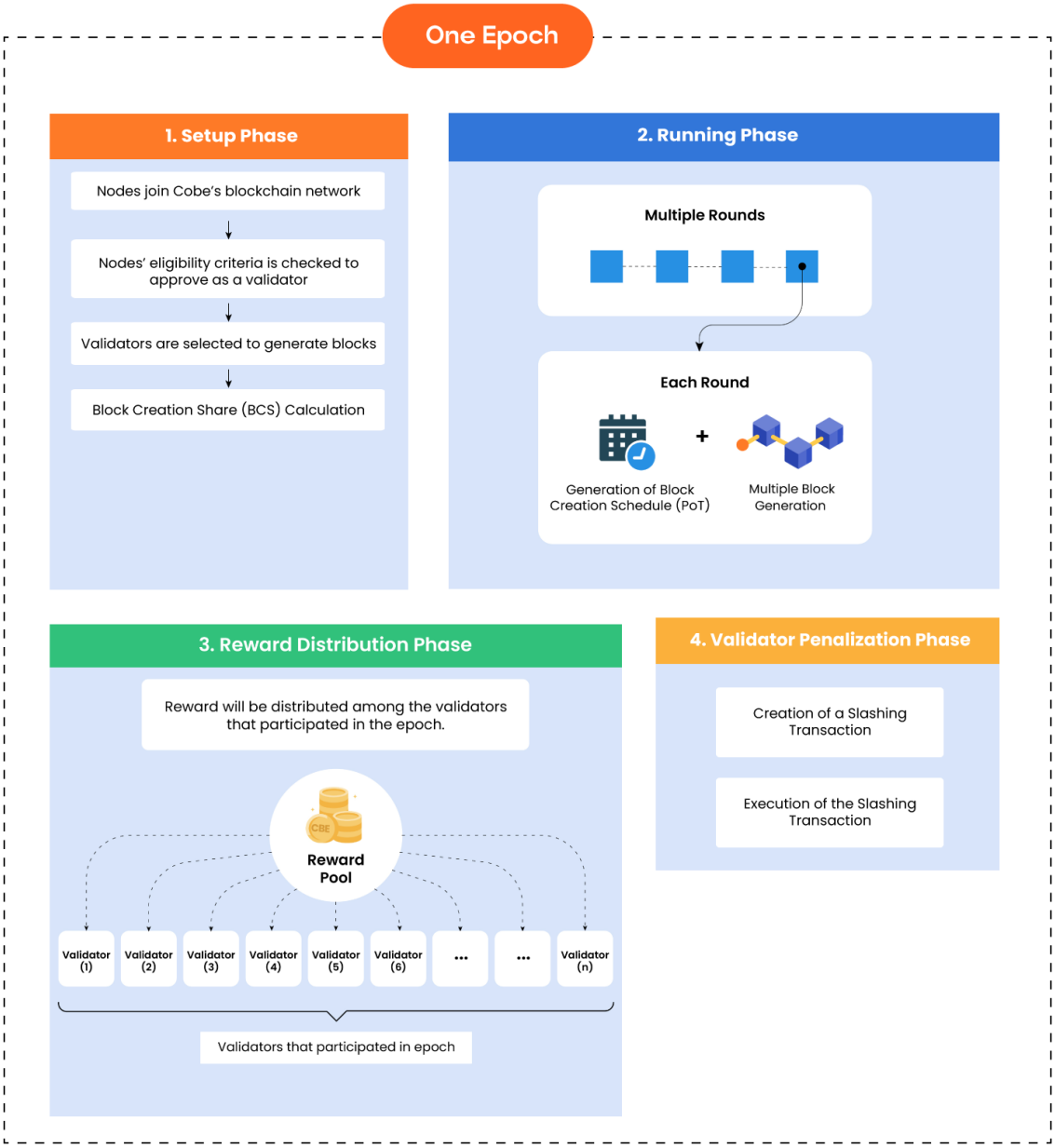

Figure 3: Cobe blockchain operation - one epoch.

4.2.1 Setup Phase

The following tasks are performed in the Setup phase.

1. A node joins Cobe's blockchain network and stakes its CBE (Cobe's native coin) in the Staking Pool.

2. If a node wants to become a validator in Cobe's Concurrent Proof of Stake (CPoS) blockchain, it must fulfill the eligibility criteria, which includes depositing a certain number of Cobe's native coins (CBE).

3. A predefined number of eligible validators are selected through elections performed among all selected validators.

4. During the election process, nodes can vote for other validators by linking their tokens to the validators.

5. After the election process, the Cobe Block Creation Share (BCS) will be calculated for each validator.

6. The BCS determines how many blocks each validator is to create in one complete epoch. The process of BCS calculation is discussed in detail in Section 6.

4.2.2 Running Phase

Cobe's running phase comprises several iterations or rounds, as shown in Figure 3. In each round, two tasks are performed: (i) generation of the block schedule and (ii) block creation.

Cobe's Block Schedule Generation:
In each round, validators will generate a synchronized random schedule for block generation. The process is summarized below.

1. Each validator creates a new Initial Block Schedule Message (IBSM) and shares it with the other validators in the network.

2. An IBSM contains necessary information such as an initial random number and the timestamp.

3. Each validator collects and verifies the integrity of all the IBSMs generated by other validators to formulate a block creation schedule for the next round.

4. If an IBSM is verified, then the receiving validator performs an XOR operation between (1) rnd_v, the random number of the received message, and (2) rnd_t, the random number updated after the most recent XOR operation. This process continues until the XOR operation has been completed for all the remaining IBSMs that are received from other validators in the network.

5. All the validators in the network then calculate the Synchronized Global Random Number (SGRN).

6. The SGRN is then fed into the Cellular Automaton Engine (CAE) to generate block creation schedule for that round. In this phase, all the validators in the network generate a copy of the globally synchronized block schedule. For details, please refer to Section 5.1.3.

Block Generation Phase:
After completing the block schedule generation phase, each validator proceeds to the block creation phase. In this phase, validators create or propose new blocks, which are then validated using Cobe's Proof of Turn (PoT) consensus protocol. Upon receiving a block from other nodes in the blockchain, a node is required to validate it. Block validation encompasses two key

aspects: (i) transaction validation, and (ii) the validation of the block itself. A block that receives validation from two-thirds of the nodes is considered 'final' and is subsequently added to the chain.

4.2.3 Reward Distribution Phase

The reward distribution mechanism in Cobe's Proof of Turn (PoT) consensus protocol is designed to encourage behaviors that enhance the network's health and security. In Cobe's Proof of Turn (PoT) consensus protocol, rewards are primarily earned through two key activities: (i) proposing new blocks and (ii) validating the blocks created by others. When a validator is selected to propose a new block in a given turn, they are rewarded with transaction fees and a specific block proposal reward.

In addition to block proposal, validators in the Cobe's Proof of Turn (PoT) consensus protocol play a critical role in validating blocks proposed by their peers. This validation process involves verifying the transactions contained in the block and ensuring adherence to the blockchain's rules and protocols. Validators who consistently participate in these activities and follow the rules of the network are rewarded, with the frequency and size of these rewards depending on the amount they have staked and the total number of active validators in the network.

4.2.4 Validator Penalization Phase

In blockchain technology, the concept of validator penalization, often referred to as 'slashing,' is a critical security mechanism. This process serves several purposes and offers various benefits, which are presented below.

- **Maintaining Network Integrity and Security:** Slashing is primarily used to maintain the integrity and security of the blockchain. If a validator acts maliciously or negligently (like validating fraudulent transactions or creating blocks on multiple chains), slashing penalizes them by imposing financial penalties (usually in the form of losing a portion of their staked tokens). This discourages dishonest behavior and helps to secure the network against attacks like double-spending.
- **Promoting Reliable and Consistent Participation:** Validators are expected to be online and actively participate in the consensus process. Slashing can be used to penalize validators who have excessive downtime or fail to participate, ensuring a more reliable and consistent network operation.
- **Enhancing Decentralization:** By penalizing misbehaving nodes, slashing helps to prevent any single validator or group of validators from gaining too much power or influence over the network. This supports the decentralized nature of blockchain technology.
- **Building Trust Among Users:** Knowing that there are strong deterrents against malicious activities and rules to ensure honest behavior builds trust among users, encouraging greater participation.
- **Network Efficiency:** Slashing can indirectly lead to higher network efficiency. Validators are motivated to improve their hardware and software infrastructure to avoid penalties, leading to a more robust and efficient blockchain network.

Validator Penalization Process

Cobe's validator penalization/slashing process consists of two main parts: (i) the creation of a slashing transaction, and (ii) the execution of the slashing transaction. These are discussed next.

Creation of a Slashing Transaction:

Cobe's blockchain will employ the following process for the creation of slashing transactions.

1. **Violation Detection:** Cobe's blockchain network protocol will continuously monitor the behavior of its validators. When a validator commits a slashable offense (like double signing, surround voting, etc.), this behavior is detected by either (i) the protocol or by (ii) other validators who are also monitoring the state of the blockchain.
   (a) Protocol-Level Detection. Like all blockchain protocols, Cobe's Proof of Turn (PoT) consensus protocol is also comprised of a set of rules. Cobe's node agents are equipped with a set of rules that can easily detect violations.
   (b) Detection by other validators. Apart from the protocol itself, other validators can also play a crucial role in monitoring and detecting violations. This includes utilizing techniques like Watchdogs and Whistleblowing. In the watchdog mechanism, validators monitor the actions of other validators as part of maintaining the network's integrity. In the whistleblowing process, the network includes a mechanism where validators can report others for violations. If a validator detects a violation, such as double voting or double signing, they can create proof and submit it in a slashing transaction. Validators are often incentivized to report violations. A portion of the slashed stake of the offending validator may be given as a reward to the whistleblower, encouraging validators to actively monitor for violations.
2. **Creation of the Slashing Transaction:** When a violation is detected, evidence of the offense is gathered and packaged into a "slashing transaction."
3. **Broadcasting to the Network:** The slashing transaction is then broadcast to the Cobe blockchain network, just like any other transaction. It enters the pool of pending transactions (mempool) where it awaits inclusion in a block.
4. **Incentivizing Inclusion in a Block:** Validators who are responsible for creating new blocks will be incentivized to include slashing transactions in the blocks they produce. There can be a whistleblower reward, a portion of the slashed stake, allocated to the validator that includes the slashing transaction in a block.
5. **Independent Verification by Validators:** Before including a slashing transaction in a block, the validator node that is responsible for creating that the concerned block will independently verify the evidence it contains. This is crucial to prevent false accusations or errors from leading to unjustified penalties.
6. **Consensus on the Slashing Transaction:** After a slashing transaction is included in a block and the block is finalized after Cobe's Proof of Turn (PoT) consensus process, the evidence and the slashing become part of the permanent record of the blockchain.

Execution of the Slashing Transaction:

Once the slashing transaction is finalized, the next phase, 'stake slashing' of the culprit validator, commences. The slashing process is presented below and in the Figure 4.

1. **Triggering the Slashing Logic:** Once the block containing the slashing transaction is finalized, the logic coded into the slashing transaction is automatically triggered, through a smart contract call.

2. **Calculation of the Penalty:** Calculating the penalty for the target validator is an important and complex task. This process involves reducing the stake of the offending validator by a specific amount. The exact penalty amount depends on the offense and other factors, such as the total amount of CBE staked on the network, validator ejection, etc. Cobe's penalty calculation method includes a dynamic penalty scaling parameter that adjusts based on the total amount of CBE staked in the network and the number of validators being slashed around the same time. This design aims to impose more severe penalties in the case of larger, coordinated attacks on the network.

3. **Updating Validator Status:** These changes in the validator's stake and status are recorded in the validator registry, a component of the Cobe protocol that tracks the status of all active and inactive validators.

4. **Notifications and Records:** The changes in the validator's status and balance are recorded on the blockchain, providing a transparent and immutable record of the slashing event.

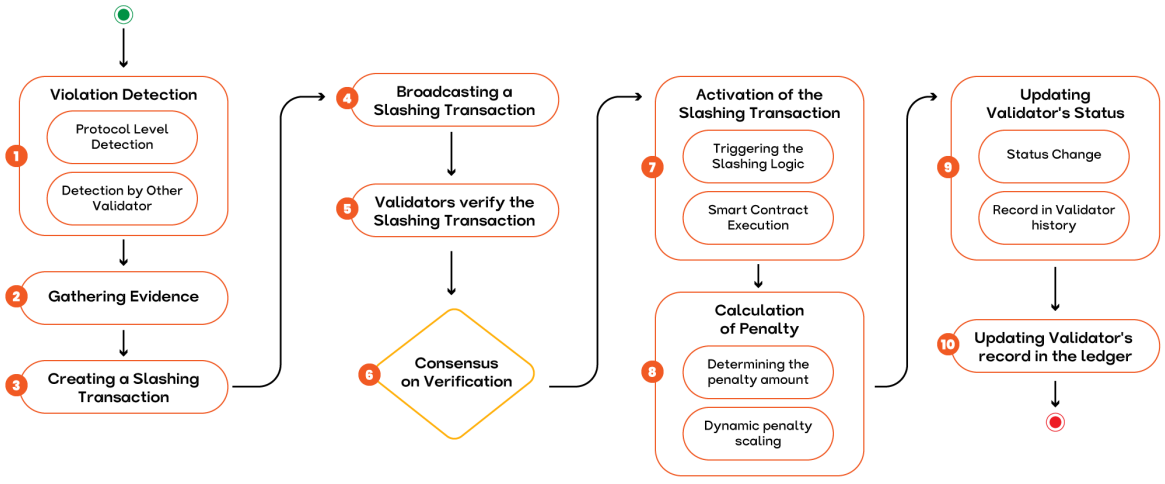

Figure 4: This process entails penalizing validators in the event of violations.

5 Cobe's Proof of Turn (PoT) Consensus Protocol

Cobe's Proof of Turn (PoT) consensus protocol boosts the block creation throughput by ensuring that validators don't need to wait for the selection of the next validator to create the new block. This is achieved by using 'Randomized Early Scheduling' (RES). By using the RES technique, all validators generate the same random schedule for block creation. Each validator is assigned a unique *turn* represented by `TurnID`. A turn refers to a discrete quantifiable timeslot, which is calculated by all the validators. A validator can generate blocks in its own turn only and not in other validators' turns. This is known as Proof of Turn (PoT).

Proof of Turn (PoT) is a block schedule generation methodology that creates a true random schedule of validators that will create blocks in a particular round (cycle). This approach exploits the functionality provided by the Trusted Platform Module (TPM) or virtual TPM

(vTPM) combined with a Cellular Automaton (CA) to generate a true random schedule, which is verified by all the validators before creating any block.

Proof of Turn (PoT) can be encoded in a one-dimensional array, which holds the node ID (of the validating node) as an element of the array, while the array index represents the Turn ID (`TurnID`) of the round. The `TurnID` is used to determine when a validator is allowed to generate the block. Below we present the improvements to the blockchain network due to the inclusion of the Proof of Turn protocol.

- Since each validator creates a 'Synchronized Random Schedule' at the beginning of each round, validators are not required to decide who will create the next block at each step. This reduces block creation time and thus increases block creation throughput.
- In Proof of Turn, validators are allowed to create blocks only in their own turns or timeslots. If a validator creates a block in some other validator's timeslot, the blockchain will not accept that block and the validator may get penalized. This ensures an additional level of security, which can also help to identify spoof or fake blocks.
- Proof of Turn (PoT) enforces additional measures such as (i) preventing one validator from generating two consecutive blocks, and (ii) limiting the number of blocks a validator can generate per round, among others. These measures not only help identify fraudulent transactions and blocks but also enhance overall security.

5.1 Working of the Cobe's Proof of Turn (PoT) Consensus Protocol

Cobe's Proof of Turn (PoT) consensus protocol comprises the phases described below.

1. Cobe's Proof of Turn (PoT) Initial Block Schedule Message (IBSM) Generation
2. Cobe's Proof of Turn (PoT) Initial Block Schedule Message (IBSM) Propagation
3. Cobe's Proof of Turn (PoT) Schedule Generation
4. Block Generation
5. Block Propagation
6. Block Validation
7. Block Finalization
8. Incentive Mechanism

5.1.1 Cobe's Proof of Turn (PoT) Initial Block Schedule Message (IBSM) Generation

In this phase, each validator prepares an Initial Block Schedule Message (IBSM). An IBSM contains a random number generated by the validator and the timestamp field. The random number is used to generate Block Creation Schedule (BCS) as discussed in Section 5.1.3.

5.1.2 Cobe's Proof of Turn (PoT) Initial Block Schedule Message (IBSM) Propagation

When a validator creates an IBSM, it must be propagated in the network so that it reaches every validator. In Cobe blockchain networks, validators use a multicasting strategy to share information with the other validators. Multicasting is a widely used communication technique

used to deliver data to a group of recipients only. If the IBSM is not received within a specified time duration, then it will not be considered and will be discarded. In Algorithm 2, T_{pct} represents the IBSM circulation timer. This timer is used to control how long a node will wait for the IBSMs from other validators to be received.

5.1.3 Cobe's Proof of Turn (PoT) Schedule Generation

In this section, we present the process of generating the Cobe Block Creation Schedule (BCS). The process starts with each validator verifying the integrity and authenticity of the received Initial Block Signature Messages (IBSMs) by checking their signatures. If a validator cannot verify an IBSM's signature, it discards the IBSM and sends a response message back to the source validator. Conversely, if the IBSM is validated, the receiving validator will perform an exclusive OR (XOR) operation between (i) the random number (rnd_v) of the received message and (ii) a locally generated random number (rnd_t). The XOR operation is performed to generate an initial random number that will be common for all the validators. The resultant random number vector is stored back in (rnd_t). This process continues till the XOR operation is completed with all the remaining IBSMs received from other validators. Given that the random number for each node is counted once only, all validators compute the same random number vector (∇`rnd`), i.e., the same SGRN.

This random number (∇`rnd`) will serve as an initial key to generate the random schedule. This random number will pass on as an input to the actual Schedule Generation Module (SGM). The SGM will generate a random number by using the Cellular Automaton (CA) 'Rule 30' [2]. The SGM will generate a 64 bit random number. If the newly generated random number is greater than the total number of validators, then the modulus can be taken to adjust it within the range. If the random number is the same as was generated in the previous iteration, then it is ignored, and a new random number is generated by using the same approach.

This process continues until the required schedule is generated. All the random numbers generated by the SGM module will be added to the schedule list `L[]`. The schedule list `L[]` is basically an array whose index refers to `TurnID`, and its value shows the validator's ID. Please refer to Algorithm 2 for more details. The SGM can have different control parameters to accept or reject the generated random number. As the random number refers to the validator's ID, it can ensure that validators cannot generate more blocks than the allowed limit, two consecutive blocks cannot be generated by the same validator, etc. By using this methodology, all validators will generate the same schedule for block creation for that round only. Cobe's Proof of Turn (PoT) consensus protocol setup and schedule generation phases are presented in Figure 5.

5.1.4 Block Generation

After successfully completing the block schedule generation phase, validators will enter the block creation phase. In this phase, validators will create the new blocks.

5.1.5 Block Propagation

The gossip protocol refers to a specific type of P2P (peer-to-peer) communication that takes place between nodes that are a part of a distributed network. Each node is paired with another node at random and sends data to it. This process is continuously repeated to spread data through the system quickly and effectively.

This protocol is widely used in distributed networks because of its scalability, fault tolerance, and robust nature. In the literature, different gossip protocols have been proposed, such as dissemination protocols, aggregation protocols, and anti-entropy protocols. Of all of these, Cobe

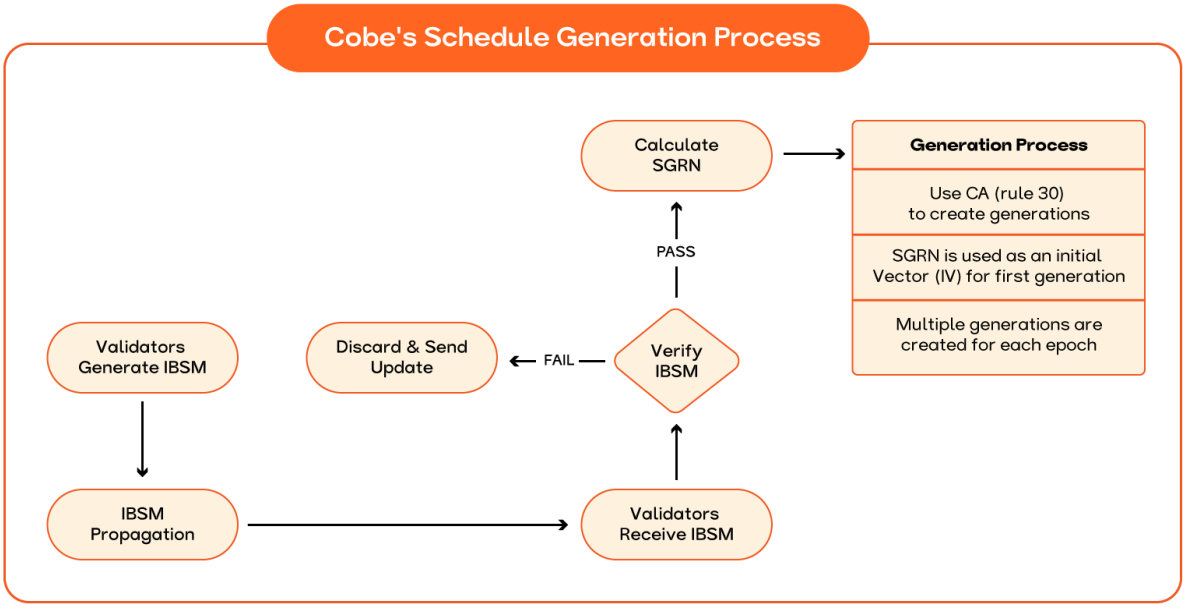

Figure 5: Setup and schedule generation phases of Cobe's Proof of Turn (PoT) consensus protocol.

will use a dissemination protocol in which a node sends data to multiple nodes simultaneously, like multicasting in legacy networks.

5.1.6 Block Validation

When a validator receives a block from other nodes in the blockchain, it must be validated. Block validation includes two main tasks:

1. Validation of the transactions that are included in the block.
2. Validation of the block itself – i.e., the block header.

When a validator receives a new transaction, it validates it. As every transaction must be signed by the owner of the transaction, the digital signature of the transaction is verified initially. After this, the transaction is considered valid if the sender in the transaction has an initial balance in their wallet equal to or greater than the amount being sent in the transaction (including any transaction fee, etc.) Rules and requirements for transaction verification may vary depending on the protocol in use.

If the transaction is not validated successfully, it will be discarded. However, a complete transaction log is maintained for audit purposes.

When a validator receives a block for validation, it will perform a series of steps that are presented in Algorithm 1 and listed below.

1. Validators will verify the digital signature of the received block. If the digital signature of the block is not verified successfully, then the block is discarded.
2. Validator will ensure that the timestamp of the received block must be greater than the timestamp contained in the IBSM received from that node.
3. Since each validator has a synchronized global schedule, the validator will check the timestamp of the received block along with the node ID from where the block was received within the Cobe block schedule generated earlier.

4. The Cobe block schedule generated earlier is essentially the list that contains the mapping between node ID and the timeslot (`TurnID`) assigned to it.

5. The Turn ID can be converted into a timestamp and vice versa. For example, if the duration of one timeslot is t seconds and the initial timestamp is t_i, then the timestamp of the 10_{th} slot or turn will be $t_{10} = t_i + 10 * t$.

6. Therefore, if the time of the received block lies within the range of $t_{10} \pm \delta$, where δ is an agreed acceptable variation, then the block will be accepted; otherwise it is rejected.

Algorithm 1: Cobe Block Validation Process

Input: Block B_i
Output: *boolean*

```
Function validateBlock(Block B_i):
    /* Verification of Digital Signature of the block B_i.  */
    call verifyDigitalSignature(B_i);
    if signature verification == false then
        return false; /* block will not be generated.  */
    end
    /* Verify Proof of Turn.  */
    /* It matches that block is generated at correct turn.  */
    if timestamp(B_i) < timestamp(B_{i-1}) then
        /* This condition is true or holds if the timestamp of a block B_i is
           less than the timestamp of the previous block.  */
        return false;
    end
    /* Below condition ensures that timestamp of B_i is in correct range and
       the hash of B_i is valid.  */
    if timestamp(B_i) < timestamp(B_{i+1}) then
        if validateTransactions(B_i) == true then
            return true;
        end
    end
    return false;
End Function
```

5.1.7 Block Finalization

In Cobe's blockchain network, block generation and block finalization are achieved via two independent operations. All deterministic finality algorithms require at least "2f + 1" non-faulty nodes, where f is the number of faulty or malicious nodes in the network - excluding the honest nodes. In other words, if a block is verified by two-thirds of the nodes in the blockchain, then it will be considered "final" and added to the chain. Cobe's block finalization function monitors the attestation votes cast to each block by different attesters and blocks are finalized only when two-third of the votes are cast to a block.

5.1.8 Incentive Mechanism

In Cobe's blockchain, validators have the opportunity to earn through different incentive mechanisms that includes (i) block creation reward, and (ii) transaction fees sharing.

1. The transaction fee of all transactions is accumulated into a *'Collection Pool'*.

2. At the end of each epoch, the transaction fee is distributed among all active validators.

5.2 Algorithm of Cobe's Proof of Turn (PoT) Consensus Protocol

Cobe's Proof of Turn (PoT) consensus protocol is presented in Algo. 2 and technical details of each step in the following sections.

Algorithm 2: Cobe's Proof of Turn (PoT) consensus protocol.

```
/* DECLARATION */
V={v_1, v_2, v_3, ..., v_n} /* Set of elected validators. */
Total no. of Validators = |V|
Schedule List = L[ ] /* Array to store list of validators IDs. */
Turn ID = TID /* A variable to store turn id no. */
Count[ ] /* Array to track the count of blocks to be created by the
   validators, array index represents node ID. */
MaxBlock[ ] /* Array that stores the max. blocks can be created by a
   validator, array index represents node ID. */
blocksRequired /* Total no. of blocks to be created in a round. */
blocksCreated /* Blocks created in a round. */
Φ = Σ_{i=1}^{v} φ_i /* Blocks creation share of each validator. */
IBSM Circulation Timer = T_pct
Random no. generated by V, either via TPM or via library = rnd_local
Random no. received from other validator = rnd_v
/* SETUP PHASE */
join(); /* New nodes join the Cobe Blockchain Network. */
staking(); /* New nodes deposit / update stake in Cobe stake-pool. */
selection(); /* Nodes are selected to produce new blocks. */
/* COBE IBSM GENERATION AND PROPAGATION */
foreach <round> do
    foreach <validator> do
        /* Validators flood their IBSMs to other validators by using
           multicasting. */
        m = generateIBSM() /* Each validator generates the IBSM. */
        floodIBSM(m); /* Validators flood the IBSM to each other. */
        /* All validators must wait by T_pct seconds, before entering block
           schedule generation phase. */
        wait(T_pct);
```

```
/* BLOCK SCHEDULE GENERATION PHASE */
/* In this phase all validators will generate a random schedule
rnd_t = rnd_local
foreach <validator v> do
    /* Each validator verifies the received IBSM, it checks and verifies
     message signature and hash to ensure authenticity and integrity.  */
    if verifyIBSM() == fail then
        discard(m);
        sendUpdated(v);
    else
        rnd_t = rnd_v ⊕ rnd_t
g = rnd_t
TID = 0;
blocksCreated = 0;
while true do
    /* Generate new random no.  by using Cellular Automata rule 30.  */
    g' = buildGeneration(g)
    g' = g' (mod |V|)
    if g == g' then
        continue
    else if Count[g] > MaxBlock[g] then
        continue;
    L[TID] = g';
    g = g';
    blocksCreated++;
    Count[g]++;
    TID++;
    if blocksRequired == blocksCreated then
        break; /* Schedule completed.  */
```

```
Input: Generation g
Output: New Generation g'
Function buildGeneration(Generation g):
    /* A new random number is generated via CA rule 30 [2].  */
    g' = CArule30(g)
    return g'
End Function
Function generateIBSM():
    rnd = generateRND()      /* Generated via TPM. */
    t_s = timestamp()
    return Signed(message);
```

5.3 Worked Example of the Cobe's Proof of Turn (PoT) Consensus Protocol

In this section, we present a diagrammatic illustration of the workings of Cobe's Proof of Turn (PoT) algorithm.

Step 1: Each validator creates a new IBSM and shares it with the other validators in the network (Figure 6).

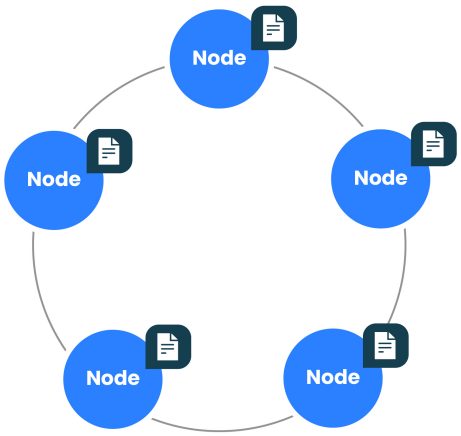

Figure 6: Validators create a new initial block schedule message (IBSM).

Step 2: Each IBSM contains the necessary information required by Cobe's Proof of Turn (PoT) consensus protocol to formulate the block schedule for the next round, including a random number and timestamp.

Step 3: Each validator verifies the integrity of the IBSMs received from all the other validators in the network (Figure 7).

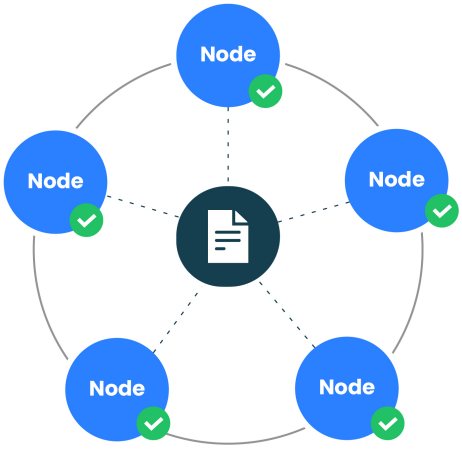

Figure 7: Initial block schedule message (IBSM) integrity check.

Step 4: If an IBSM is verified, then the receiving validator performs an XOR operation between (1) rnd_v the random number of the received message and (2) rnd_t the random number updated after the most recent XOR operation. This process continues until the XOR operation is completed for all the remaining IBSMs that are received from the other validators in the network.

Step 5: All the validators in the network then calculate the Synchronized Global Random Number (SGRN) (Figure 8).

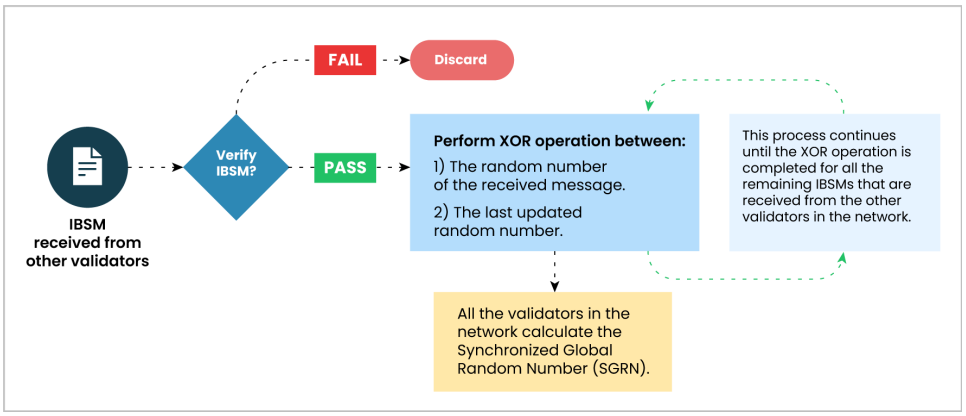

Figure 8: Calculation of Synchronized Global Random Number (SGRN).

Step 6: The SGRN is then fed into the Cellular Automaton Engine (CAE) to generate more random numbers. In this phase, all the validators in the network generate a copy of the globally synchronized block schedule (Figure 9).

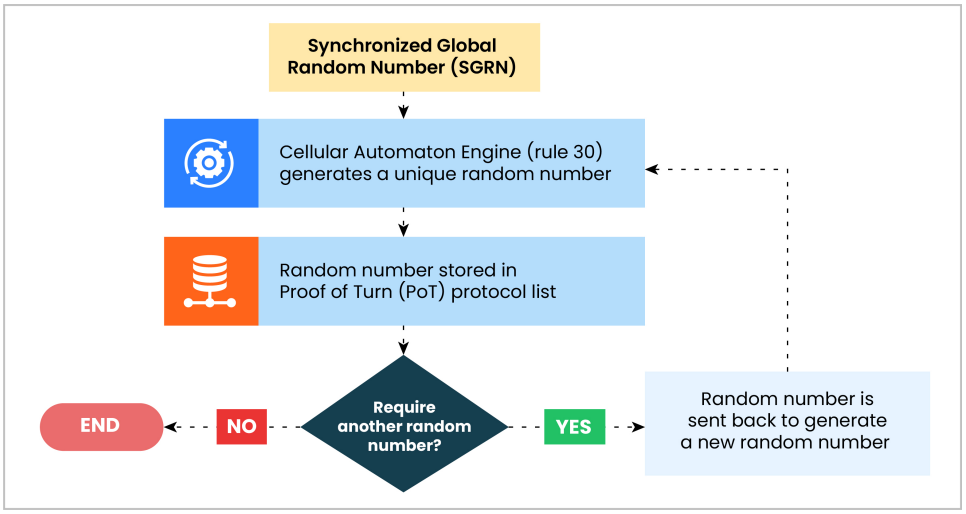

Figure 9: Random Number Generation via CA30.

5.3.1 Mathematical Interpretation

If "N" is the number of validators in the network, let's assume N is equal to 5 as shown in Figure 6. The random numbers and timestamps generated by all the five validators is shown in the Table 4.

Each validator verifies the digital signature, attached to IBSM, received from other validators to establish the integrity of the IBSM, as shown in Figure 7.

Because the blockchain is a decentralized network, IBSMs from different validators arrive at different times. In the following example, we show the operation of validator 1. All remaining validators in the network will execute the same process to compute the SGRN.

Suppose validator 1 receives an IBSM at the times t_1, t_2, t_3, t_4 from validators 3,2,5,4 respectively as shown in the table below:

Table 4: Shows the random numbers generated by each validator and the timestamp.

Validator ID	Random Number (rnd_{local})	Timestamp (t_s)
1	1267436890	1674088739
2	2267144535	1674088738
3	3634048938	1674088736
4	2399829310	1674088740
5	1664968184	1674088738

Table 5: Arrival of an IBSM at validator 1.

Time	Validator ID
t_1	3
t_2	2
t_3	5
t_4	4

Then validator 1 will perform the following steps to compute the SGRN:
Initially:

$$rnd_t = rnd_{local}$$

At time t_1:

$$rnd_t = rnd_t \oplus rnd_3$$

where,
rnd_3 is the random number received from validator 3.

$$rnd_t = 1267436890 \oplus 3634048938$$
$$rnd_t = 2467341040$$

At time t_2:

$$rnd_t = rnd_t \oplus rnd_2$$

where,
rnd_2 is the random number received from validator 2.

$$rnd_t = 2467341040 \oplus 2267144535$$
$$rnd_t = 338772903$$

At time t_3:

$$rnd_t = rnd_t \oplus rnd_5$$

where,
rnd_5 is the random number received from validator 5.

$$rnd_t = 338772903 \oplus 1664968184$$
$$rnd_t = 4029531912$$

At time t_4:

$$rnd_t = rnd_t \oplus rnd_4$$

where,
rnd_4 is the random number received from validator 4.

$$rnd_t = 4029531912 \oplus 2399829310$$
$$rnd_t = 2133310006$$

When the XOR process is completed for all validators, the rnd_t, computed in the last iteration, becomes the Synchronized Global Random Number (SGRN). This number is then fed into the Cellular Automaton Engine (CAE) to generate the block schedule.

$$SGRN = rnd_t$$
$$SGRN = 2133310006$$

6 Cobe's Permissionless CPoS Chain: Block Creation Share (BCS) Calculation

Cobe's blockchain uses a node's reputation score (ρ) and its stake size (σ_s) to calculate the block creation share of each validator. It uses a weighted sum approach to calculate the reputation score of the node. The reputation score of the node can be determined by using Eq. 1, while the validation share score (VSS) of the node can be calculated by using Eq. 2. The validation share score (VSS) is used to calculate the actual validation share of the node by using Eq. 3. The share of each validator is proportional to the score achieved by an elected validator. The total number of blocks to be created in an epoch is calculated via Eq. 4.

$$\rho = \omega_1 * \sigma_d + \omega_2 * O + \omega_3 * \beta_{missed} - \omega_4 * \beta_{bad} \tag{1}$$

where,
ρ = Reputation Score
σ_d = Stake Age
O = Online Age
β_{missed} = Missed blocks
β_{bad} = Bad blocks
ω_1, ω_2, ω_3, and ω_4 are weights assigned to each parameter

Note: For the first round, the values of β_{missed} and β_{bad} are 0.

$$VSS = \omega * \sigma_s + \rho \tag{2}$$

where,
VSS = Validation Share Score.
ρ = Reputation score.
ω = Weight.

$$VS = \left\lfloor \left(\frac{VSS}{\sum_{i=1}^{n} VSS} \right) \beta_{Max} \right\rfloor \tag{3}$$

where,
VS= Validation Share
n= Total number of elected validators

β_{Max} = Maximum blocks can be created in an epoch.

$$\beta_{Total} = \sum_{i=1}^{n} VS_i \tag{4}$$

where,
β_{Total} = Total No. of blocks to be created in an epoch
VS_i = Share of each validator
n = Total number of elected validators.

6.1 Stake Age (σ_d)

Stake age refers to the duration, measured in epochs, during which a validator has locked its coins (CBE) in the blockchain network. Stake age can be calculated using Eq. 5.

$$\sigma_d = CBE_{Locked} * \tau_n \tag{5}$$

where,
σ_d = Stake age
CBE_{Locked} = CBE staked or locked by a validator
τ_n = No. of epochs for which CBE staked or locked

6.2 Online Age (O)

Online age is another important factor that shows a node's reliability in the blockchain network. Online age can be measured from the time the node has been online during the last 'x' epochs. Online age can be calculated as follows:

$$O = \frac{\tau_o}{\tau_t} \tag{6}$$

where,
τ_o = No. of epochs a validator remains online
τ_t = Total epochs under consideration.

For example, if a validator remains online for 10 epochs in the previous 21 epochs, then its online age will be

$O = \frac{10}{21} * \approx 0.476$

7 Concurrency and Cobe's Parallel Chain Architecture

Scalability is one of the major challenges in blockchain networks. To enhance network scalability, the Cobe blockchain network utilizes two state-of-the-art techniques: (1) concurrent block creation by using fork chains, and (2) concurrent transaction execution and verification.

7.1 Concurrent Block Creation

Concurrent block creation is an innovative approach in blockchain technology designed to address scalability and efficiency issues. Traditionally, blocks in a blockchain are created sequentially, which can lead to bottlenecks, especially as the number of transactions increases. With concurrent block creation, multiple blocks are generated simultaneously by different nodes in the network. This parallel processing capability allows for a significant increase in the number of transactions that can be processed within a given time frame, thereby enhancing the throughput of the network. Cobe's has developed two approaches for concurrent block creation, known as (i) DApp-Based Concurrent Fork Chains, and (ii) Load-Aware Concurrent Fork Chains.

7.1.1 DApp-Based Concurrent Fork Chains

As the transaction load for a specific DApp increases, parallel independent chains, known as fchains, are created exclusively for that DApp. To accelerate transaction processing and block finalization, each DApp-based fchain operates a distinct instance of Cobe's Proof of Turn (PoT) consensus protocol. At the end of each round, all DApp-based fchains are seamlessly integrated into the main chain. The following section details the complete process of DApp-based concurrent fork chains.

- At the beginning of each round, a few validators are chosen randomly to monitor the transaction load on the network, which includes transaction pool, as shown in Figure 10.

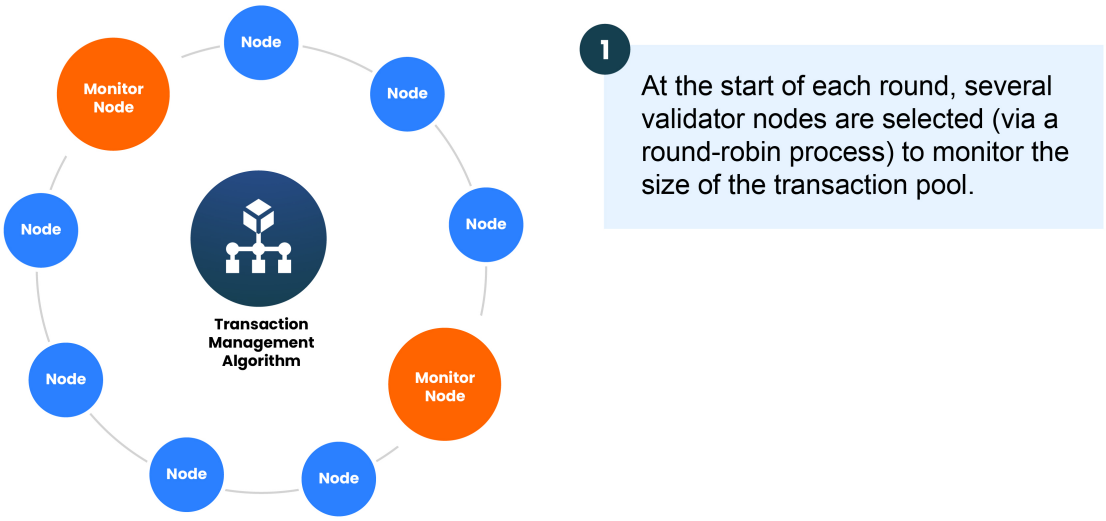

Figure 10: Random selection of monitor nodes.

- If the transaction load on a DApp increases, the monitor nodes will multicast a `fork()` message across the network. Subsequently, a fork-block will be appended to the parent chain, as illustrated in Figure 11.

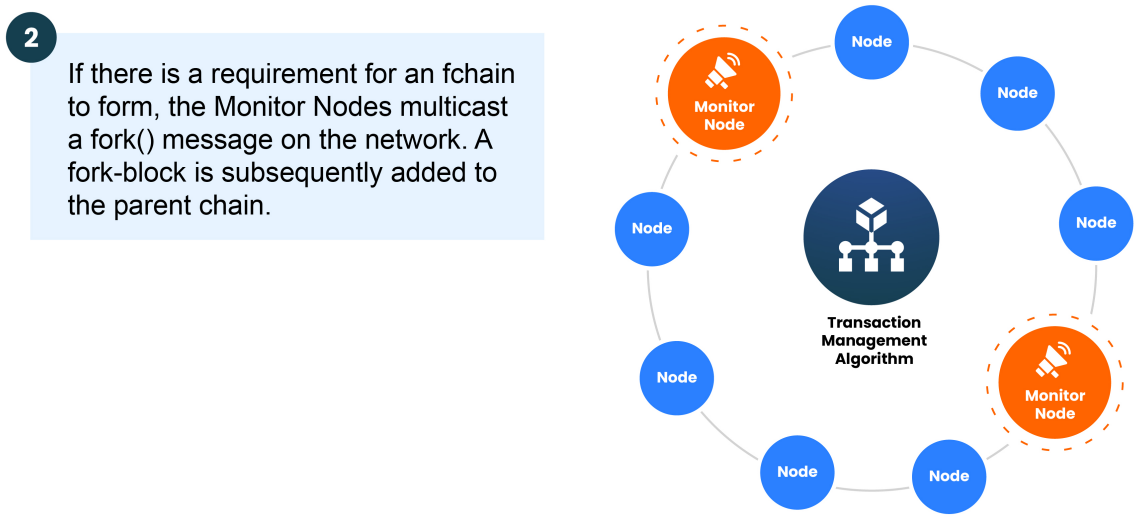

Figure 11: Monitor nodes send `fork()` message on the network.

- Afterwards, a unique subchain (fchain) will be created for that DApps, as shown in Figure 12.

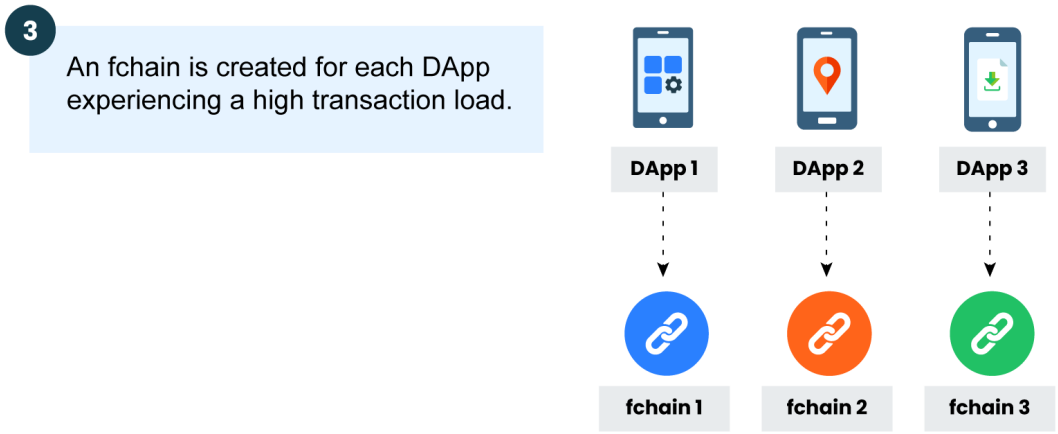

Figure 12: DApps having their own separate fchain.

- Each fchain will run a separate instance of Cobe's Proof of Turn (PoT) consensus protocol; thus each chain will have separate block schedule. This process is shown in Figure 13.

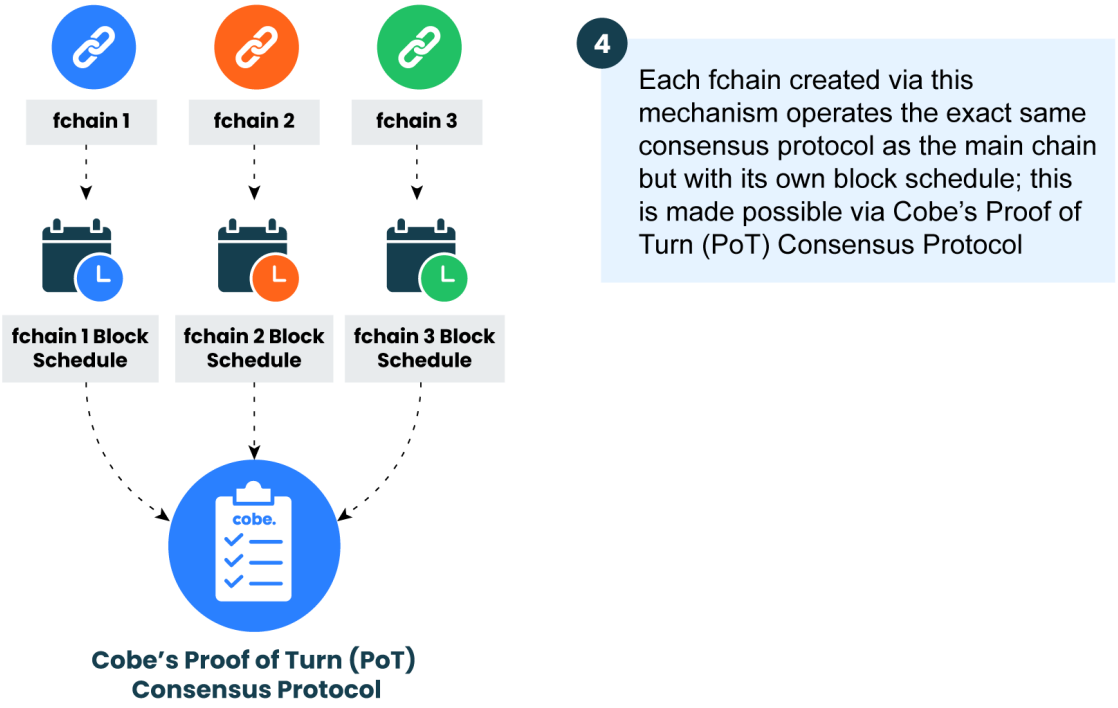

Figure 13: DApp-based block schedule for each fchain.

- At the end of each round, each DApp-based fchain will be synchronized to the main chain, as shown in Figure 14.

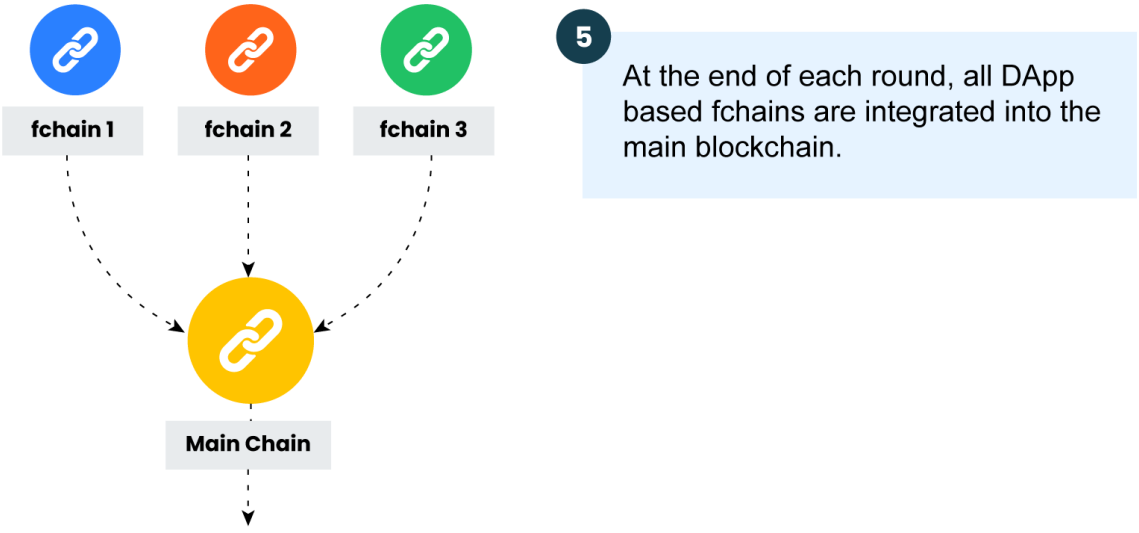

Figure 14: Synchronization of DApp-based fchains.

7.1.2 Load-Aware Concurrent Fork Chains

In contrast to DApp-based fchains, Load-aware fchains are created when the overall transaction load on the network is high. This approach allows the Cobe blockchain to dynamically adjust to varying loads, ensuring efficient processing across the entire network. While DApp-

based fchains are specifically designed to handle increased activity within individual DApps, Load-aware fchains are activated in response to network-wide demand, optimizing the system's performance and stability when facing high transaction volumes. This dual approach of using both DApp-based and Load-aware fchains represents a robust strategy for managing scalability in the blockchain network.

Working of Fork Chains:

1. As with DApp-based fchains, at the beginning of each round, a specific number of validators are chosen as monitor nodes through a round-robin algorithm. These nodes are responsible for monitoring the transaction processing system, which includes a transaction pool, as illustrated in Figure 15.

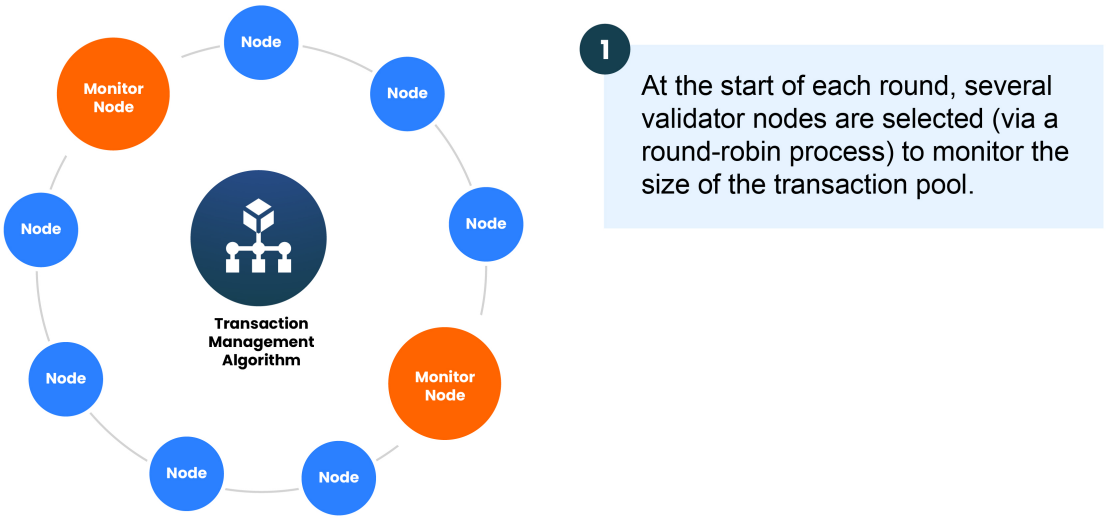

Figure 15: Random selection of monitor nodes.

2. Based on the load of the networ, if there is a need to generate fork chains, the monitor nodes will multicast a `fork()` message across the network. Subsequently, a `fork-block` will be added to the parent chain, as depicted in Figure 16.

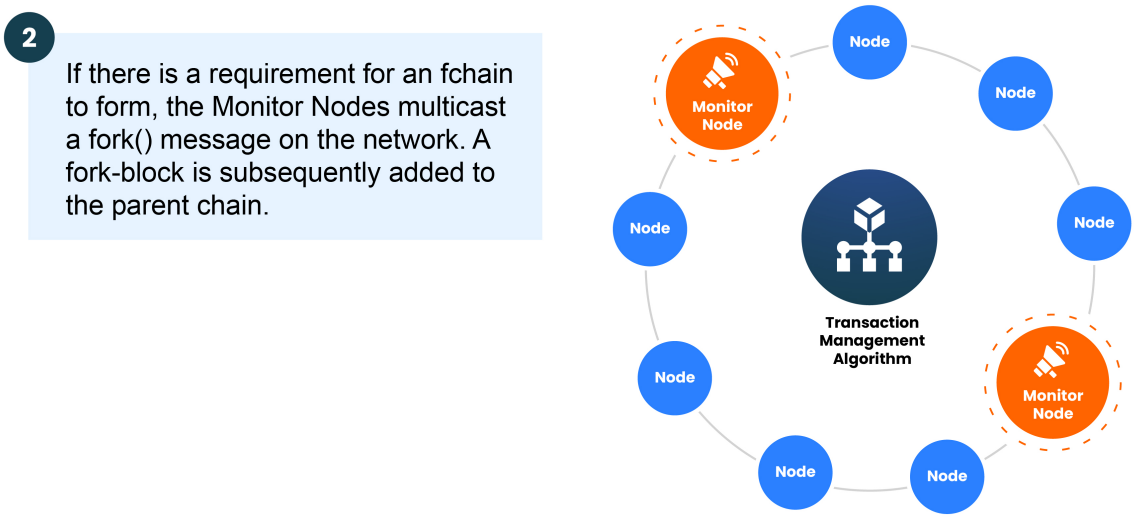

Figure 16: Load-aware fork creation.

3. The block schedule, generated at the start of the current round, will be split into two halves. The first half will be assigned to the parent chain, and the second half of the schedule will be assigned to the fchain, as shown in Figure 17. If more than one fchain is to be generated, then the block schedule will be divided an equivalent number of times. For instance, if N number of fork chains are to be generated, then block schedule will be split into 2^N parts.

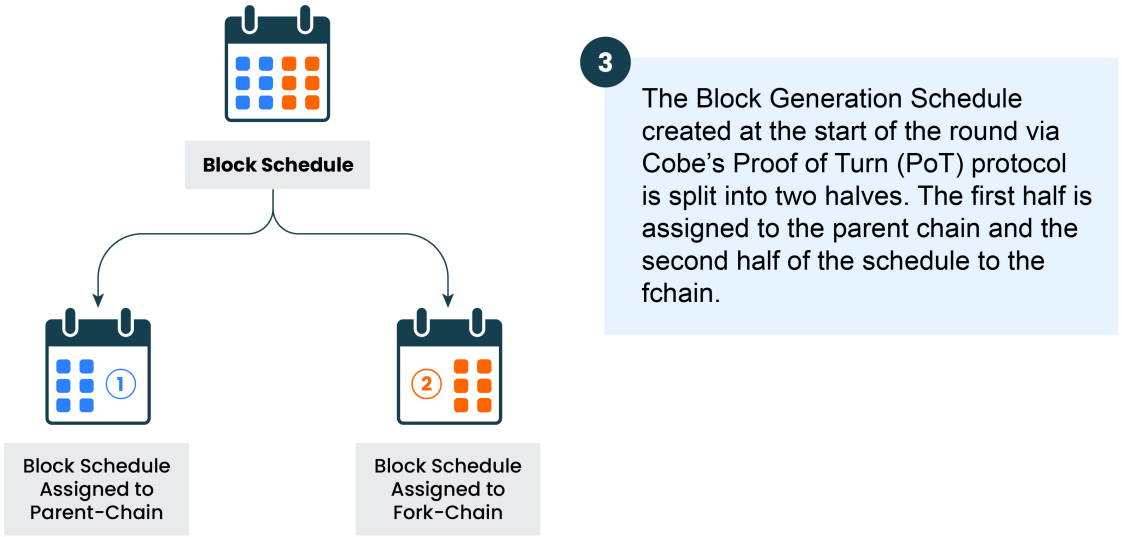

Figure 17: Block schedule splitting process.

4. All fchains and the parent chain will use the same transaction pool, as shown in Figure 18.

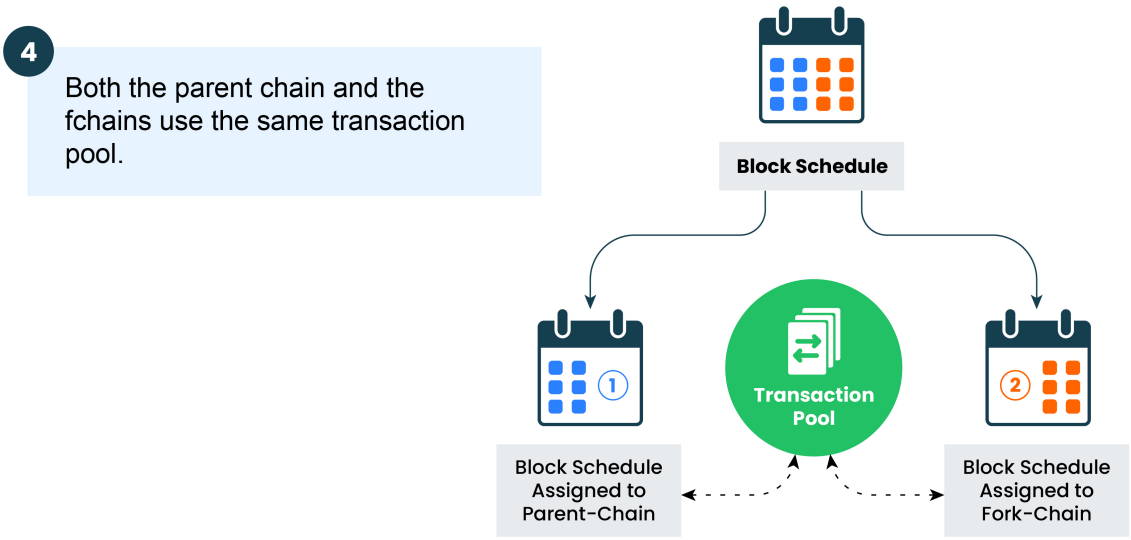

Figure 18: Parent and fchains will share the same transaction pool.

5. However a new table, known as an ftable will be created. This ftable will include the following fields: (i) address (either of sender or receiver), (ii) fchain ID. The fchain ID field is used to track which fork chain processed transactions related to which wallet, as shown in Figure 19.

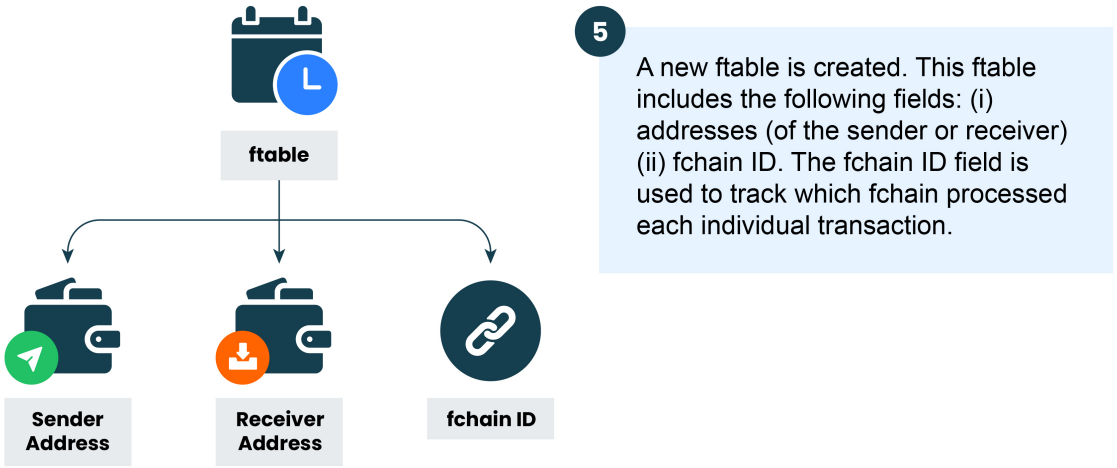

Figure 19: Creation of ftable.

6. When a new transaction is submitted to the transaction pool, the transaction processing system will look up the addresses of sender and receiver in the ftable. If the address is not found in the ftable, this means that it is a new independent entry. The transaction processing system will then assign this transaction to any one of the chains in a round-robin manner and update the ftable, as shown in Figure 20.

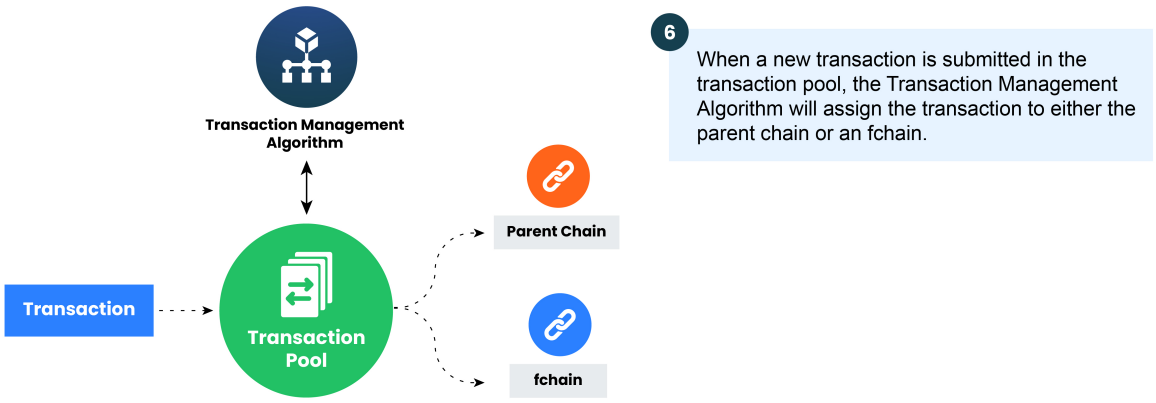

Figure 20: Transaction assignment process for fchains.

7. If a previous entry of the sender and receiver addresses is found in the ftable, the transaction procssing system will assign the transaction to the same chain. The whole process is represented in Figure 21.

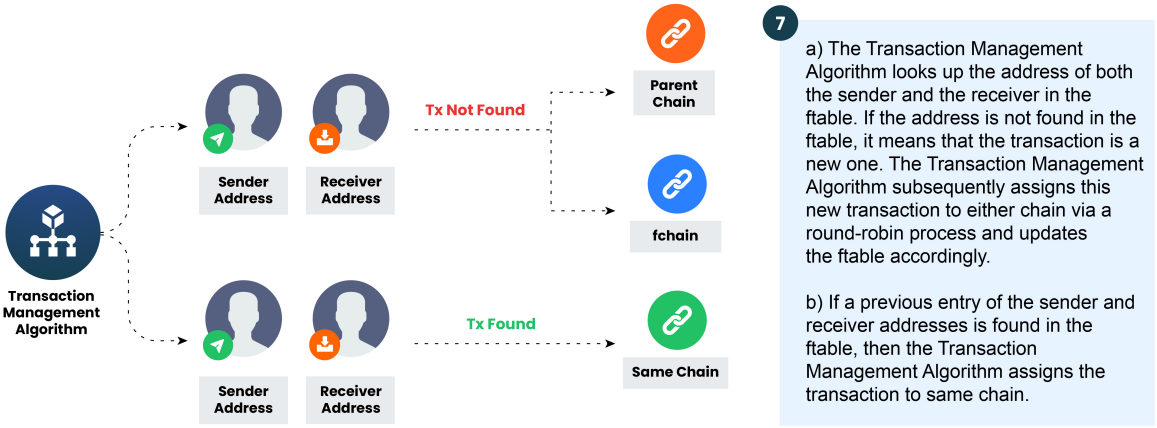

Figure 21: Transaction assignment process for fchains.

Synchronization of fchains:

The chain synchronization process is invoked when (i) a round is completed and (ii) the network load falls. Chain synchronization is achieved with the help of 'monitor' nodes, as presented below.

1. The monitor nodes are also responsible for the synchronization of fchains.
2. Monitor nodes will multicast a `join-message` across the network when two conditions are met: (i) a round has been completed, and (ii) the network load decreases.
3. Upon receiving a `join-message`, all fchains are either merged or linked together using a `join-block`.
4. The `join-block` serves as a jumbo block and contains references (pointers) to the last block of each fchain. These pointers are critical for maintaining the integrity of the

blockchain, as they ensure a coherent and verifiable link between the separate fchains and the main chain.

5. A `join-block` is created after each round to synchronize the state of fchains across the entire network. Depending on the network load, it is possible to increase or decrease the number of fchains after each `join-block`.

6. For more details, please refer to Figure 22.

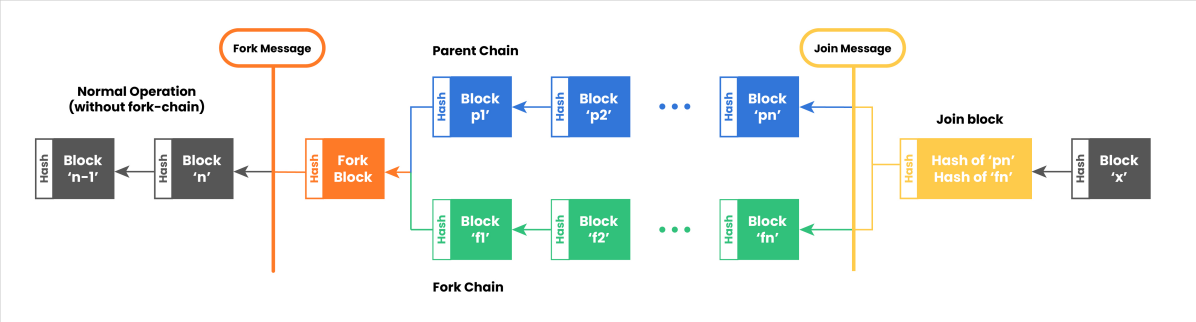

Figure 22: Fork fchain synchronization.

7.2 Concurrent Transaction Execution and Verification (CTEV Algorithm)

Blockchains organize transactions into blocks, where traditionally, each transaction within a block is executed sequentially, one after the other. Cobe's Concurrent Transaction Execution Protocol and Verification (CTEV) algorithm introduces an innovative approach, employing cutting-edge 'static transaction analysis' techniques. This enables the execution and verification of transactions concurrently within a block boosting throughput. Below, the steps of the CTEV algorithm are described in detail.

- The CTEV algorithm selects a set of transactions from the transaction pool and constructs an arbitrary linear sequence, as shown in Figure 23.

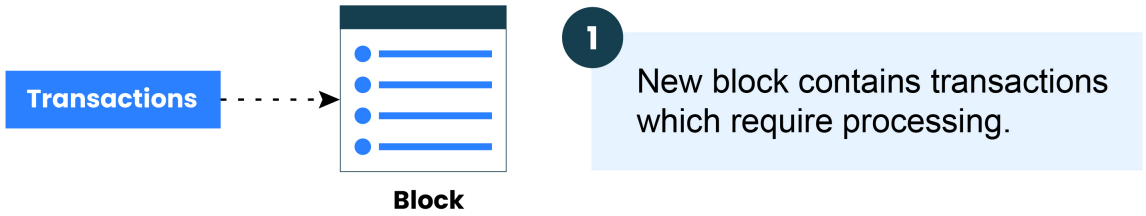

Figure 23: Transaction block that contains transactions.

- The CTEV algorithm constructs an occurrence net of transactions, following the algorithm in [3]. Occurrence nets are a specific type of acyclic Petri net. The occurrence net will consist of transaction sets having no dependency, as shown in Figure 24.

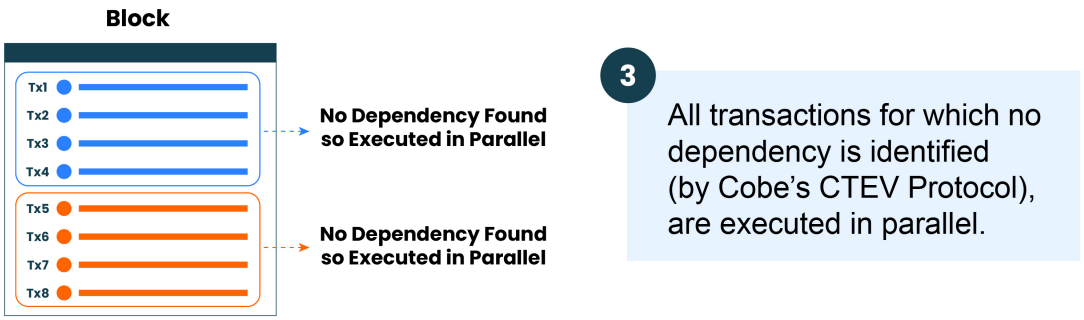

Figure 24: Construction of separate occurrence net.

- The CTEV algorithm executes transactions concurrently according to the occurrence net, exploiting the available parallelism on the node; see Figure 25.

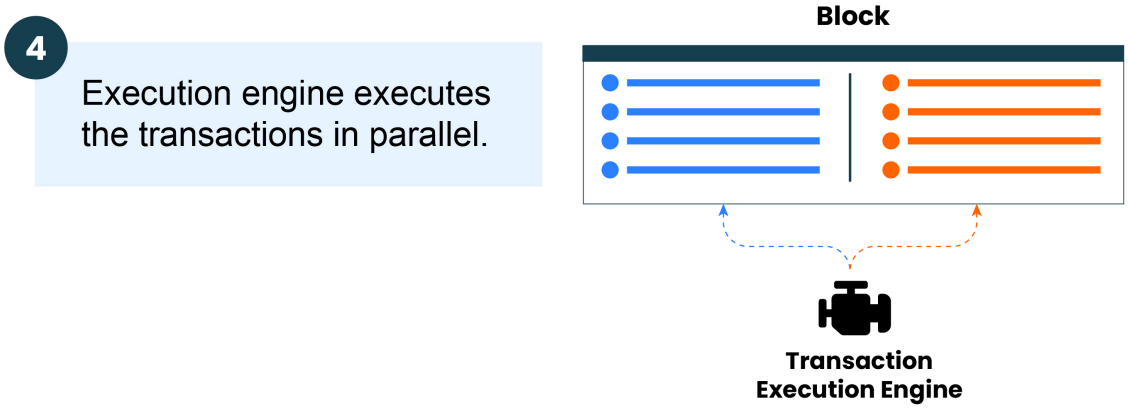

Figure 25: Parallel execution of transactions.

8 Cobe's Permissioned Concurrent Proof of Authority (CPoA) Blockchain

Proof of Authority (PoA) is a consensus protocol in which a group of validators are preselected to validate blocks. In a Proof of Authority (PoA) based blockchain network, a node needs to provide its identity information instead of stake in order to become a validator. This approach contrasts with mechanisms like Proof of Stake (PoS) based blockchain networks. In PoA, validators are typically chosen based on their reputation and reliability, ensuring a degree of trust in the network. As discussed in Section 1, Cobe's Concurrent Proof of Authority (CPoA) consensus protocol is optimized for solutions that require confidentiality, fixed transaction fees, and a high throughput (e.g., supply chain management, product authentication, information notarization). In CPoA blockchain network, validators are known as master nodes. Nodes are required to meet the eligibility criteria discussed in Section 8.2 to become master nodes.

8.1 Operation of Cobe's CPOA Blockchain

Cobe's CPoA blockchain operates in a manner similar to Cobe's CPoS chain. Each epoch is comprised of four phases: (i) a setup phase, (ii) a running phase, (iii) a reward distribution phase, and (iv) a validator penalization phase. During the setup phase, tasks essential to blockchain operations are performed, such as the onboarding of master nodes and the calculation of Block Creation Share (BCS). The running phase includes several rounds, in each round two main tasks are performed: (i) the generation of a block creation schedule, and (ii) block generation. In the reward distribution phase, rewards are distributed among the master nodes that participated in that epoch. Following the end of each epoch, master nodes may be subject to slashing for exhibiting bad behavior. The entire operation of the CPoA blockchain is depicted in Figure 26.

8.2 Master Node Eligibility Criteria (E_c)

The eligibility criteria are the most important requirement for a node to join Cobe's CPoA chain. A node must verify its identification (ID) as outlined in Section 8.3, Master Node Onboarding Process. After the ID verification process, the node's account balance is checked. A node must have a minimum of 350,000 CBE in its account. In Cobe's CPoA chain, a node's eligibility can be calculated using Eq. 7.

$$E_c = \Xi * CBE_{Min} \tag{7}$$

where,
Ξ = Node identification status (0 or 1). One means node has passed ID verification process.
CBE_{Min} = Minimum (CBE) account balance a master node must maintain.

The eligibility score of a node must be greater than 0 to become master node.

8.3 Master Node Onboarding Process

A node meeting the eligibility criteria can be become a master node. The master node onboarding process is presented below.

1. The initial step for a node is to submit an application to become a master node.
2. The node must complete an essential ID verification process. During this phase, documentary evidence is required to authenticate the node's true identity.

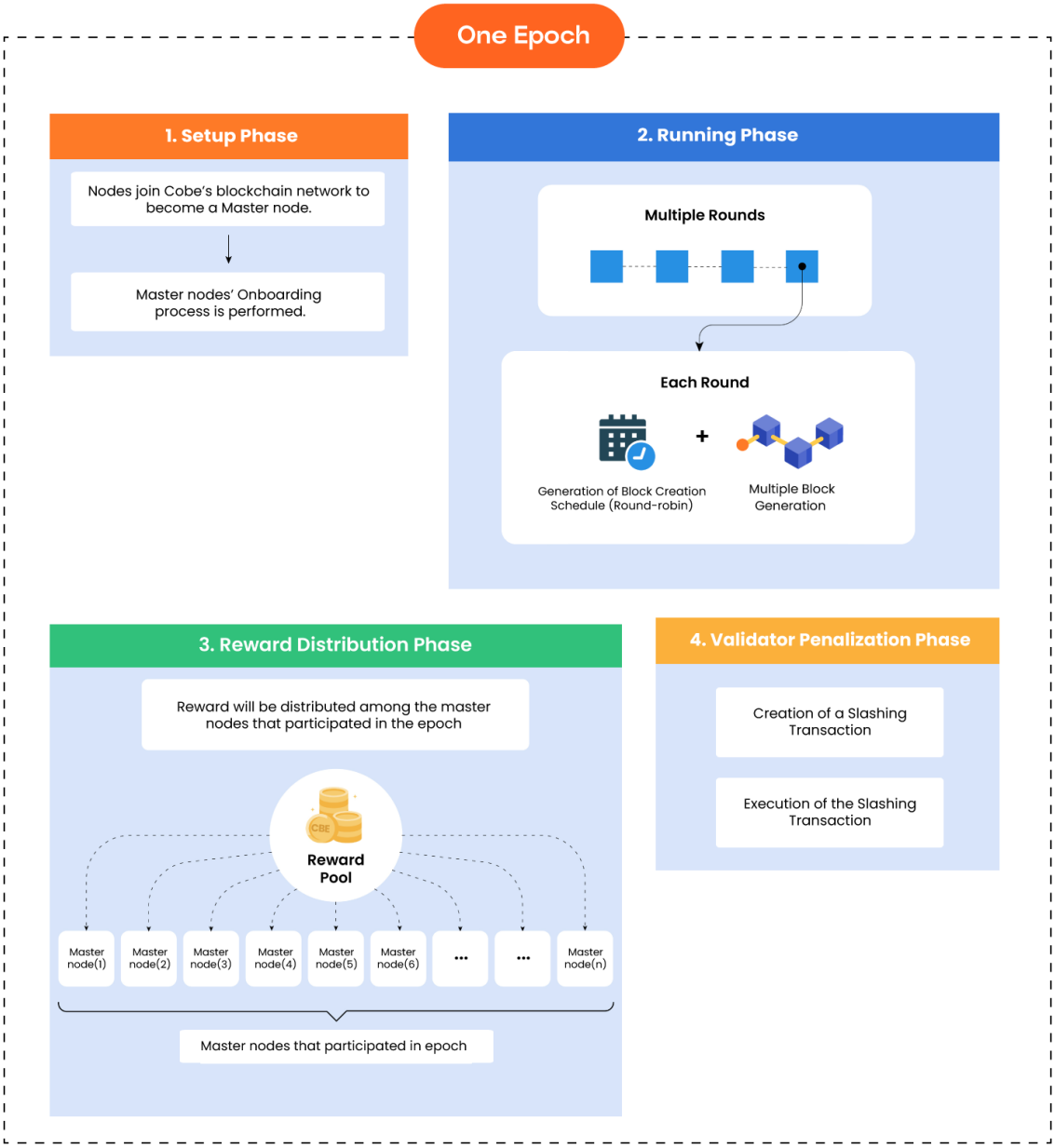

Figure 26: Cobe's CPoA blockchain operation - one epoch.

3. After successful verification, the node is required to deposit 350,000 CBE and accept the terms & conditions.

4. The reputation score of the master node is calculated using by Eq. 8 as presented in Section 8.4.

5. The Validation Share Score (VSS) of the master node is calculated by using Eq. 2, and the Validation Share (VS) is calculated using by Eq. 3.

6. Cobe's CPoA Master Node onboarding process is depicted in Figure 27.

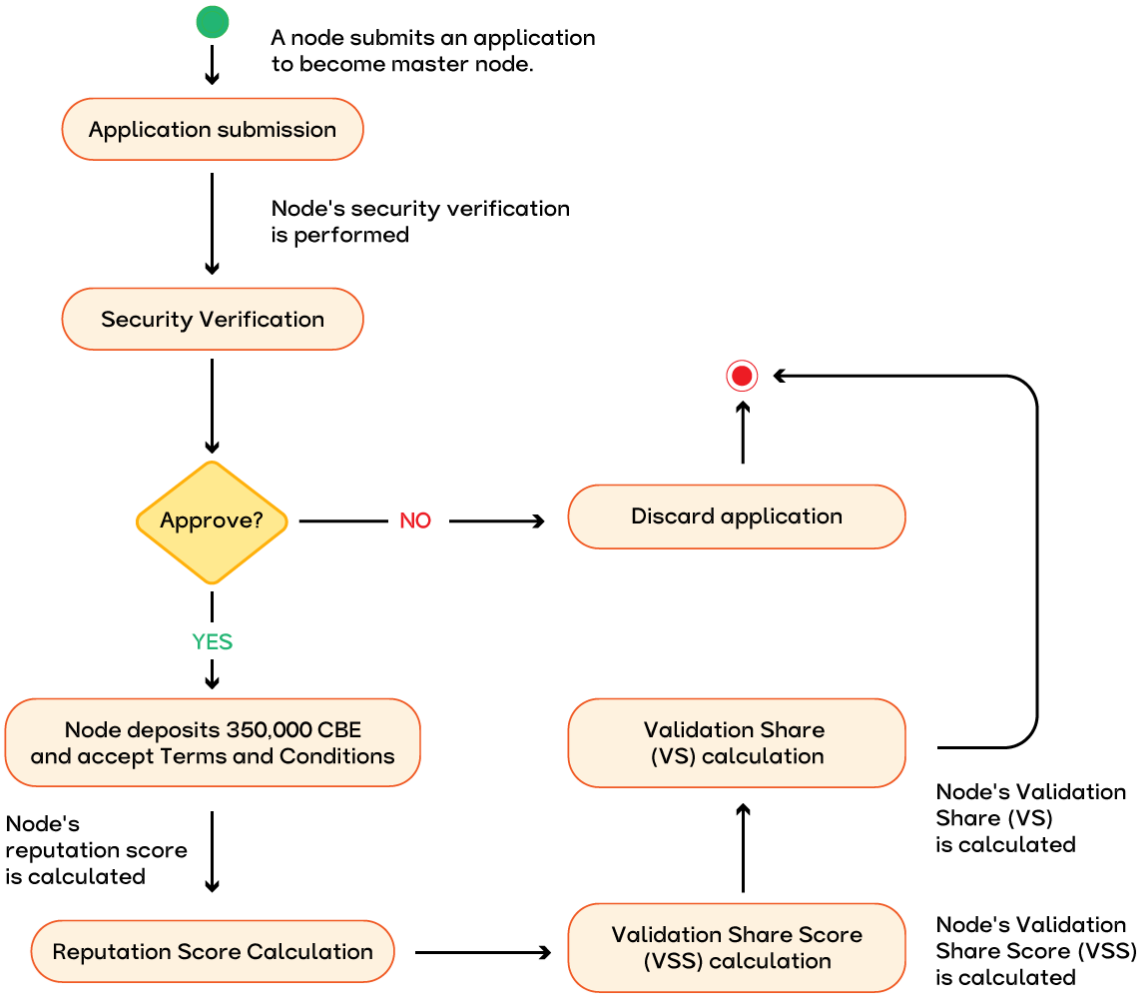

Figure 27: Master node onboarding process.

8.4 Master Nodes Reputation Score (ρ) Calculation

In the first round, all selected master nodes will create the same number of blocks in a round-robin fashion. But over a period of time, master nodes reputation score is built. The reputation score will be used to determine how many blocks a master node can validate in a round. A lower score will result in a reduced number of blocks to be validated by that master node, and consequently, a lower reward. If a node's reputation score falls below a certain minimum threshold, it may be completely removed from participating in the network. In Cobe's CPoA blockchain, the reputation score is calculated by using Eq. 8.

$$\rho = \omega_1 * O - \omega_2 * \beta_{missed} - \omega_3 * \beta_{bad} \tag{8}$$

where,
ρ = Reputation score.
O = Online age.
β_{missed} = Blocks missed by the master node
β_{bad} = Bad blocks created by the master node.
ω_1, ω_2, and ω_3 are weights assigned to each parameter.

8.4.1 Master Nodes' Validation Share Score (VSS) & Validation Share (VS) Calculation

In Cobe's CPoA blockchain, the Validation Share Score (VSS) and Validation Share (VS) can be calculated using Equations 2 and 3, respectively.

8.5 Block Creation Schedule Generation

In the running phase, Master nodes generate a block creation schedule that is pivotal for maintaining the integrity and efficiency of the blockchain network. In Cobe's CPoA blockchain, this process is thoroughly organized via the implementation of a round-robin scheduling algorithm. The round-robin algorithm ensures an equitable distribution of block creation opportunities according to the Master nodes' validation share. The round-robin process allows for the rotation of the block creation task among the available Master nodes, which not only guarantees fairness but also enhances the network's security and prevents any single node from dominating the block creation process. This systematic scheduling contributes to the overall stability and reliability of the CPoA blockchain network, ensuring that transactions are processed smoothly and without undue delay.

8.6 Cobe's Confidentiality Enhancement Framework (CCEF)

Confidentiality, integrity, and availability are the three pillars of any information system. In a blockchain, data integrity is ensured by using cryptographic hash functions; similarly, ledger availability is achieved through the distributed nature of the network. However, confidentiality is still a challenge for both permissioned and permissionless blockchains. Cobe introduces a novel confidentiality enhancement framework through which users can protect their ledger.

Cobe's CPoA chain uses a homomorphic encryption (HE) technique to ensure data confidentiality. Homomorphic encryption allows operations to be performed on encrypted data without the need for decryption. Therefore, validators do not need to view the details of the transactions; they can validate them in their encrypted state, which ensures the confidentiality of the data. In this section the working of the Cobe Confidentiality Enhancement Framework (CCEF) is presented. Cobe's CPoA blockchain has introduced two novel techniques to ensure data confidentiality and privacy.

8.6.1 Homomorphic Encryption (HE)

Homomorphic Encryption (HE) allows users to perform certain computations on encrypted data (ciphertext), obtaining the equivalent result as if the same computations were performed on the given plaintext. This feature enables users to perform any calculation without decrypting the ciphertext, as a result of which unauthorized information disclosure is prevented. Cristina R. et. al. in [4] discuss different types of homomorphic encryption: (i) Partially Homomorphic Encryption (PHE), (ii) Somewhat Homomorphic Encryption (SHE), and (iii) Fully Homomorphic Encryption (FHE).

Partially Homomorphic Encryption (PHE). A Partially Homomorphic Encryption (PHE) scheme allows users to perform only one computational operation (either addition or multiplication) an unlimited number of times on the ciphertext.

Let E be the encryption function and D the decryption function. Let $\oplus$ denote the homomorphic operation supported by the PHE scheme (either addition $+$ or multiplication $\times$). For plaintext messages m_1 and m_2, the PHE property is:

$$D(E(m_1) \oplus E(m_2)) = m_1 \oplus m_2$$

This states that decrypting the homomorphic operation of the ciphertexts gives the same result as the operation on the plaintexts.

Somewhat Homomorphic Encryption (SHE). Somewhat Homomorphic Encryption (SHE) allows limited set of operations, denoted as $\oplus$ for addition and $\otimes$ for multiplication, performed a limited number of times. For a given number n of operations, the SHE property is:

$$D((E(m_1) \oplus E(m_2)) \otimes \ldots \otimes E(m_n)) = m_1 \oplus m_2 \otimes \ldots \otimes m_n$$

Here, operations can be mixed but only applied a finite number of times before decryption is no longer reliable.

Fully Homomorphic Encryption (FHE). An FHE scheme supports an unlimited number of operations, both addition $+$ and multiplication $\times$. The FHE property, where $*$ represents either operation, is:

$$D(E(m_1) * E(m_2) * \cdots * E(m_n)) = m_1 * m_2 * \cdots * m_n$$

This expresses that any sequence of additions and multiplications applied to encrypted data will, after decryption, yield the same result as if applied to the plaintext. FHE scheme enables the construction of programs for any desirable functionality, which can be run on encrypted inputs to produce an encryption of the result. FHE eliminates the trade-off be- tween data usability and data privacy, no trusted third parties are required, and it is quantum safe. However, the challenge with FHE is its poor performance.

8.6.2 Zero Knowledge Proof (ZKP)

Zero Knowledge Proof (ZKP) is a cryptographic method by which one party (the prover) can prove to another party (the verifier) that a statement is true, without conveying any information apart from the fact that the statement is indeed true. This is especially useful in situations where privacy is a concern.

In the context of blockchain technology, ZKP allows for the verification of transactions without revealing the transaction's details. This helps in maintaining the privacy of the transaction's contents while still ensuring its validity. For example, Zcash [5] uses Zero-Knowledge Succinct Non-Interactive Argument of Knowledge (zk-SNARKs) [6] to preserves a user's payment privacy. Similarly, Cobe's CPoA (Certificate of Proof of Authority) chain employs zk-SNARKs to confirm that a payer has a sufficient balance for a transaction without revealing the balance or any other details of the payer's account. This use of ZKP within blockchains is a powerful tool for enhancing privacy and security within the ecosystem.

Cobe's CPoA chain uses zk-SNARKs to verify that the payer has sufficient balance in their account.

8.6.3 Framework Operation

Cobe CPoA chain uses FHE and ZKP together to ensure the confidentiality and privacy of user data. The fundamental operations of encrypted transactions are presented below.

1. **Transaction Creation:**
    - Alice wants to pay 'm' CBE to Bob.

- Alice's account balance is 'y'.

2. **Transaction Encryption & Proof:**
   - Alice encrypts 'm' using Bob's public homomorphic key pk_{Bob}, resulting in $E_{pk_{Bob}}(m)$.
   - Alice creates a zk-SNARK proof π that she knows a secret 'y' such that her balance is sufficient for the transaction without revealing 'y':

$$\pi = \text{ZK-SNARK}(y, \text{“Alice has at least 'm' in her account”})$$

3. **Transaction Submission:**
   - The transaction, now a tuple $(E_{pk_{Bob}}(m), \pi)$, is submitted to the transaction pool.

4. **Validation by the Validator:**
   - The validator checks the ZKP:

$$\text{Verify}(\pi) \stackrel{?}{=} \text{true}$$

   - If the proof is valid, the validator performs the homomorphic operation to adjust Alice's encrypted balance:

$$E_{pk_{Alice}}(y) \rightarrow E_{pk_{Alice}}(y - m)$$

   - The transaction is homomorphically processed to credit Bob's account:

$$E_{pk_{Bob}}(y_{Bob}) \rightarrow E_{pk_{Bob}}(y_{Bob} + m)$$

5. **Block Creation:**
   - The validated transaction is used to create a new block and is added to the ledger.

6. **Verification by Bob:**
   - Bob can decrypt his new balance and verify the transaction using his private key sk_{Bob}:

$$D_{sk_{Bob}}(E_{pk_{Bob}}(y_{Bob} + m)) = y_{Bob} + m$$

   - Bob can perform aggregate operations on his balance using homomorphic properties without decrypting.

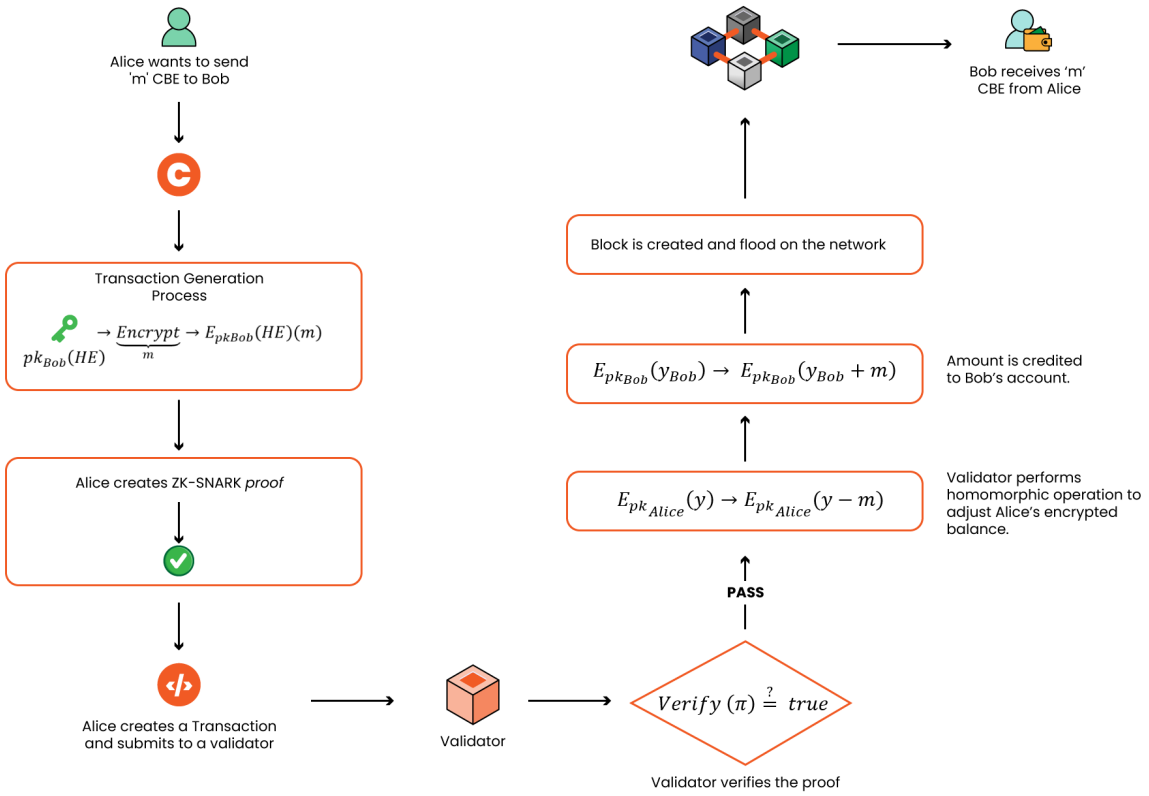

Figure 28: Transaction processing flow in CPoA.

8.6.4 Cobe Homomorphic Encryption Scheme

In this section, technical details related to the homomorphic encryption scheme used in Cobe's CPoA chain are presented.

Proc. `ParamGen()` is used to generate the necessary parameters required for encryption keys.

Procedure ParamGen()
Data: Security parameter λ, PT, K, B
Result: Params

where,
λ: denotes the desired security level of the scheme; for instance, 128-bit or 256-bit security
PT: denotes plaintext
K: represents the dimension of the vectors to be encrypted
B: represents auxiliary parameter that is used to control the complexity of the program.

Encryption parameters generated via Proc. `ParamGen()` will be used to generate encryption keys as shown in Proc. `KeyGen(Params)`.

Procedure KeyGen(Params)
Data: Key generation parameters *Params*
Result: SK, PK, EK

where,
SK: represents secrete key. SK will be used to decrypt the ciphertext
PK: represents public key. PK will be used to encrypt the PT
EK: represents evaluation key. EK will be used to perform homomorphic operations over the ciphertext.

Encryption of the given message will be performed by the encryption procedure (see Proc. `Encrypt(m,PK)`).

Procedure Encrypt(m,PK)
Data: Encryption parameters m, PK
Result: c

where,
m: represents the message to be encrypted
PK: represents the homomorphic public key of the recipient
c: represents ciphertext.

When a transaction is received by a validator, it may be required to perform various computations over the ciphertext. A generic evaluation method is represented in Proc. `Eval(c,EK,Params)`.

Procedure Eval(c,EK,Params)
Data: Parameters $c, EK, Params$
Result: $c^{'}$

where,
$c^{'}$: represents new ciphertext generated after performing some operation by the validator.

When the new ciphertext is received by the recipient, the recipient will decrypt it by using the decryption algorithm presented in Proc. `Decrypt(`$c^{'}$`,SK)`.

Procedure Decrypt($c^{'}$,SK)
Data: Parameters c, SK
Result: $m^{'}$

where,
$m^{'}$: is the new message generated after computing over the ciphertext. It will be equivalent to PT generated after performing same operation.

9 Cobe's Dual Blockchain Interoperability Architecture (CDBIA)

In this section, we present Cobe's Dual Blockchain Interoperability Architecture (CDBIA). Cobe's dual blockchain architecture is developed to provide interoperability between Cobe's Concurrent Proof of Stake (CPoS) and its Concurrent Proof of Authority (CPoA) blockchains. CDBIA will allow developers to build hybrid decentralized applications (DApps). DApps developed using CDBIA can leverage the features of the CPoS blockchain, which offers permissionless operations, a high level of transparency, variable transaction fees, and democratic processes. Additionally, they can benefit from the CPoA blockchain, known for its permissioned operations, enhanced confidentiality, fixed transaction fees, and superior throughput. Blockchain interop-

erability comprises of a set of rules, protocols, and services through which different blockchains can interoperate with each other.

9.1 Blockchain Interoperability Concepts

Blockchain interoperability can be considered as "A composition of distinguishable blockchain systems, each representing a unique distributed data ledger, where atomic transaction execution may span multiple heterogeneous blockchain systems, and where data recorded in one blockchain are reachable, verifiable, and referable by another possibly foreign transaction in a semantically compatible manner" [7]. In the context of blockchain interoperability, Besançon et. al. [8] present three categories or types of blockchain interoperability: (i) cross-blockchain interoperability (interoperability between different blockchains), (ii) interoperability between DApps using the same blockchain, and (iii) interoperability between blockchain and other technologies (such as integration with enterprise systems). The focus of this text is cross-blockchain interoperability, particularly between Cobe's CPoS and CPoA chains.

A Cross-Chain Communication Protocol (CCCP) is involved whenever a pair of blockchains interact to synchronize cross-chain transactions correctly. CCCP allows homogeneous blockchains to communicate. For instance, sidechains typically use a CCCP (e.g., Zendoo allows communication between Bitcoin-like blockchain systems) [10]. In contrast, a Cross-Blockchain Communication Protocol (CBCP) is used whenever a pair of blockchains interact to synchronize cross-blockchain transactions correctly. CBCPs allow heterogeneous blockchains to communicate. CCCPs generally utilize the interoperating blockchains' constructs and functionality (e.g., utilizing smart contracts to implement a relay), whereas CBCPs normally require blockchains to be adapted. Similarly, a Cross-Chain Transaction (CC-Tx) refers to transactions between different chains that belong to the same blockchain system (homogeneous blockchains), e.g., between two EVM-based blockchains. A Cross-Blockchain Transaction (CB-Tx) is a transaction between different blockchains (heterogeneous blockchains), e.g., between Hyperledger Fabric and Bitcoin. In the literature, the terms CC-Tx and CB-Tx are used interchangeably. Similarly, a Cross-Chain Decentralized Application (CC-DApp) is a DApp that uses cross-blockchain transactions to implement its business logic. We use the terms Cross-Chain Decentralized Application (CC-DApp) and Cross-Blockchain Decentralized Application (CB-DApp) interchangeably.

Furthermore, in the literature two additional related ideas are presented: Internet of Blockchains (IoB) and Blockchain of Blockchains (BoB). An Internet of Blockchains (IoB) is a system where homogeneous and heterogeneous decentralized networks communicate to facilitate cross-chain transactions, whereas a Blockchain of Blockchains (BoB) is a system in which a consensus protocol organizes blocks that contain a set of transactions belonging to CC-DApps that spreads across multiple blockchains. Internet of Blockchains (IoB) directly refers to the connection relationships among blockchains, whereas the term BoB refers to an architecture made possible by IoB.

Therefore, blockchain interoperabilty can be summarized as the ability of a source blockchain to change the state of a target blockchain (or vice versa), enabled by cross-chain or cross-blockchain trans- actions, spanning across a composition of homogeneous and heterogeneous blockchain systems.

9.2 Cobe's Dual Blockchain Interoperability Architecture (CDBIA)

The main responsibility of CDBIA is to enable Cobe's CPoS and CPoA blockchains to interoperate. A blockchain platform may comprise several layers [9], i.e., a data layer, network layer, consensus layer, and application layer. Each layer within the framework has a unique set of functionality, which is presented below.

- **Data Layer:** this layer describes the representation of data in the blockchain. It includes the structure and the format of the block header and specifications of transactions in the block.
- **Network Layer:** the network layer is responsible for the identity of different types of nodes in a decentralized blockchain network. A node can be of different types, e.g., full-node, light-node, validator, etc.
- **Consensus Layer:** the consensus layer deals with the blockchain consensus protocol – for example, Cobe's CPoS Proof of Turn (PoT) algorithm.
- **Application Layer:** the application layer comprises DApps and smart contracts.

9.2.1 Components of Cobe's Dual Blockchain Interoperability Architecture (CDBIA)

CDBIA introduces a few new roles that are required to enable interoperability between CPoS and CPoA blockchains. In this section, we present the description of the architecture and communication flow between the different components.

Relay Node (RN): relay nodes in CDBIA are responsible for offering relay services to their respective blockchain users, i.e., CC-DApps. These nodes are responsible for communication between CPoS and CPoA homogeneous blockchains. They are capable of using Cross Blockchain Communication Protocol (CBCP). In one blockchain instance, more than one node may be selected to provide relay services for high availability.

Decentralized Blockchain Registries (DBRs): DBR registries hold the information related to relay nodes – for example, their (i) endpoint addresses, (ii) valid signature IDs, (iii) health, etc. Whenever a relay node is setup, it must register itself to decentralized blockchain registries. The DBR will verify and confirm the authenticity and authority of the relay node. After a successful registration process, a relay node can communicate with the relay nodes of other blockchains under CDBIA.

9.2.2 Operation of CDBIA

In this section we describe the overall operation of Cobe's dual blockchain interoperability architecture. CDBIA operation is demonstrated in Figure 29. In the figure, two blockchains are presented, CPoS blockchain and CPoA blockchain. These blockchains use the CPoS and CPoA consensus mechanism, respectively.

Each blockchain selects one or more relay agents or relay nodes (RNs). For simplicity a single relay node is shown in the diagram. These RNs will register to decentralized blockchain registries (DBRs). After successful registration (authentication and verification), their status is updated to 'connected'. These RNs can communicate with other RNs of other blockchains. In this scenario, the relay node of a CPoS blockchain can communicate with the relay node of a CPoA blockchain.

When a CC-DApp running on a CPoS blockchain wants to perform a cross-blockchain transaction with a CPoA blockchain, it will submit its transactions to the CPoS blockchain RN. The RN of the CPoS blockchain, which is already connected with the DBR, will perform a lookup for the RN of CPoA blockchain. After getting the response from the DBR, the relay nodes will connect via cross-blockchain communication protocol (CBCP). CBCP is used to perform communication between two blockchains. In this scenario, the CPoS blockchain and the CPoA blockchain are two distinct blockchains; however, both blockchains operate under the same

administration. The RN of the CPoA blockchain will first confirm the authorization of the RN of the CPoS blockchain from the DBR. After verifying the authorization, their connection status will be marked as established. Once their connection status is established, they can communicate for the exchange of data/transactions.

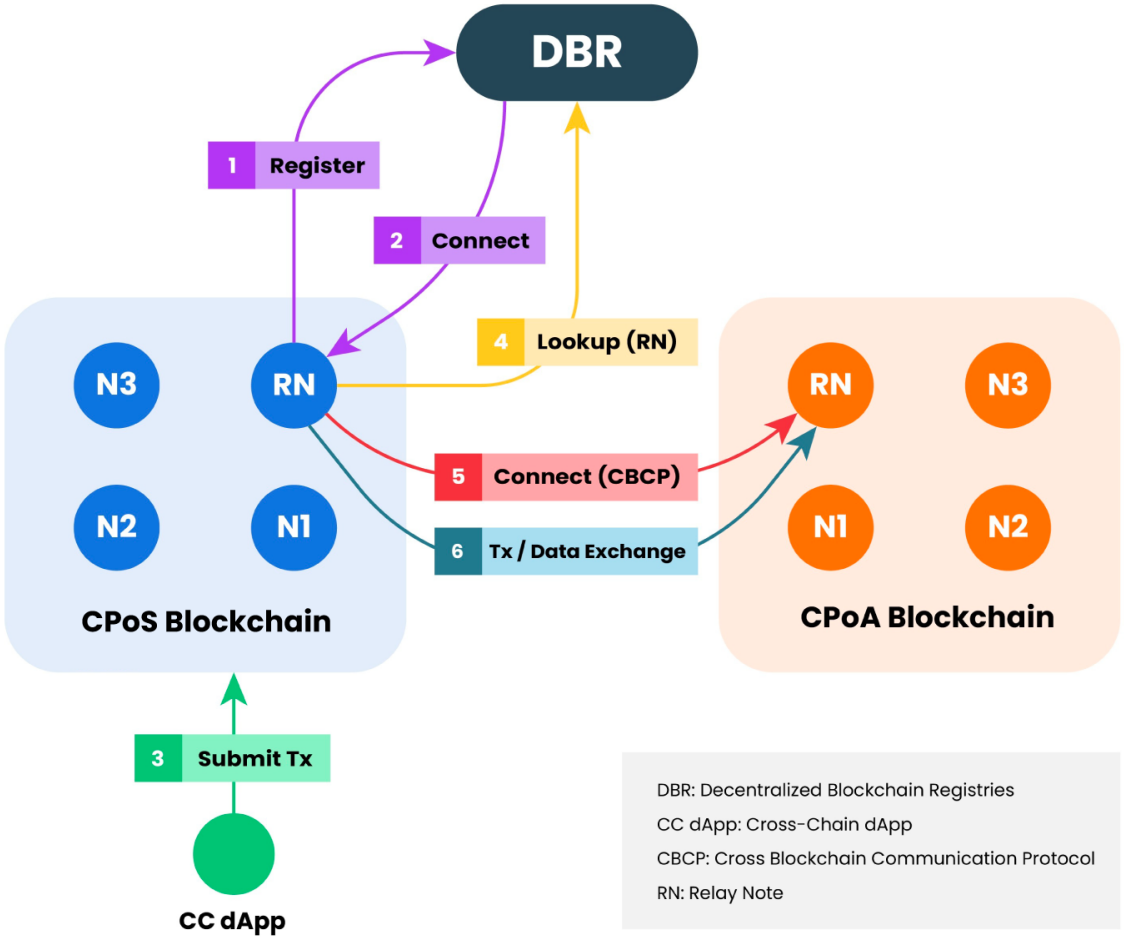

Figure 29: Cobe's Dual Blockchain Interoperability Architecture (CDBIA).

In Figure 30, we illustrate the process of transferring coins from the CPoS chain to a wallet on the CPoA chain. This process consists of several key steps, which are outlined below:

1. **Locking Coins on Initiator Blockchain**. The cross-chain bridge contract locks the coins on CPoS chain.
2. **Escrow & Verification**. The locked coins are held in escrow until they undergo verification.
3. **Minting Coins on Target Blockchain**. After verification, new coins are minted on the CPoA chain.
4. **Transferring Minted Coins**. The newly minted coins are then transferred to the user's wallet on the CPoA chain.
5. **Burning Locked Coins on Initiator Blockchain**. Finally, the locked coins are burned on the CPoS chain.

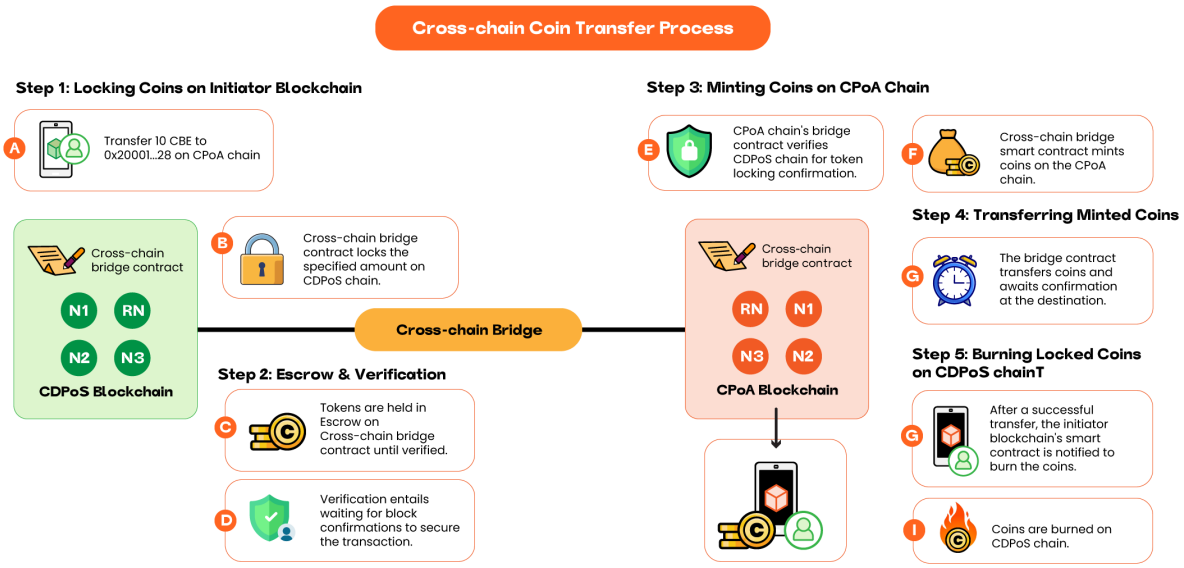

Figure 30: Cross-chain coin transfer process.

9.3 Interoperability with External Blockchains

Several approaches to achieving interoperability between different blockchain networks are currently being developed and utilized [10]. These include:

Sidechains

This approach involves two distinct blockchains, a mainchain and a sidechain. Each sidechain runs its own consensus protocol and may have different coins. A protocol for cross-chain communication connects the mainchain and sidechain, with each maintaining its own inventory of assets. Sidechains serve as a two-way peg because they include a system for transferring assets between the mainchain and sidechain. BTC Relay, Peace Relay, POA network, RSK, etc., are examples of interoperability blockchain projects based on sidechains.

Notary schemes

Transactions under this method rely on a third-party notary. The lack of trust between two parties involved in the transaction is managed by a trusted third party known as a notary. The notary may be a network of exchanges or a single exchange. Technically, notary schemes are simpler to implement when compared to other interoperability solutions. Like traditional banks and centralized exchanges, the efficiency of a notary scheme relies heavily on the notary's honesty and trustworthiness. Despite this drawback, notary schemes have been widely deployed due to their efficiency and flexibility. Centralized cryptocurrency exchanges like Coinbase and Binance are examples of businesses utilizing notary schemes.

Hashed Time Lock Contract (HTLC)

Hashed Time Lock Contract (HTLC) is a blockchain interoperability mechanism utilized to create smart contracts with the capacity to modify payment channels. HTLC primarily facilitates time-bound transactions. The recipient will not receive their funds, and the transaction will be voided, if they do not provide a cryptographic proof of payment receipt within a predetermined time frame. A time lock means that a certain quantity of cryptocurrency will be

inaccessible until a predetermined duration of time has passed.

Initially, Cobe plans to utilize the available solutions presented above to achieve interoperability with other blockchain networks. However, Cobe aims to eventually introduce its own middleware or framework to interoperate with other heterogeneous blockchains.

10 Cobe Smart Language Stack

Smart contracts are essentially computer programs stored on a blockchain network that run when predetermined conditions are met. They typically are used to automate the execution of an agreement. They can hold and transfer digital assets managed by the blockchain and can invoke other smart contracts stored on the blockchain. Smart contract code is deterministic in nature and immutable after deployment.

10.1 Overview of the Smart Language Landscape

Smart contracts are programs that are deployed on the blockchain to fully automate potentially complex transactions across the network. Typically, smart contracts react to 'triggers' (e.g., messages received from other smart contracts), and their behaviors depend on their internal states as well as on the global state of the blockchain. A typical example of a smart contract in the scope of Cobe is the Escrow service that holds assets until obligations stated in the contract have been met by the parties involved in the contract.

Smart contracts are written in a smart contract language, which is typically derived from traditional, general purpose programming languages (e.g., Solidity is based on JavaScript, while Vyper is based on Python). There is a wide range of smart contract languages, and they can be understood along two broad axes: expressivity and abstraction level.

Expressivity: the expressivity of a language characterizes the set of programs that can be written in it. For instance, Solidity is a Turing-complete language; hence any computable function can be expressed in it. This contrasts with Bitcoin script, which only supports simple conditionals related to Bitcoin transfer. Recently some blockchain researchers and communities have abandoned the Turing-completeness requirement to facilitate (sometimes automated) reasoning about smart contracts. While Turing-completeness is easily achieved (it suffices to support recursion, if-then-else, and some form of integer variable), it is often not needed to encode most smart contracts [11]. Languages such as Vyper, DAML, and Scilla are examples of smart contract languages that purposely avoid Turing-completeness to support automated verification and/or some form of auditability.

Abstraction level: the level of abstraction relates to the facilities offered by the language to the programmer in order to hide complexities of the underlying computation mechanism. We distinguish three levels that match the abstraction levels in general purpose programming languages.

- Surface (or high-level) languages (e.g., Solidity, Liquidity, Flint, Vyper) offer strong abstraction that hides away memory allocation, data serialization, etc. They often adopt one programming paradigm, such as object-oriented, functional, or procedural. They are meant to be written by programmers and compiled down to an intermediate representation (IR) or directly to a low-level language.

- Intermediate representation languages (e.g., Scilla, IELE, Yul) offer an intermediate step between high-level and low-level code to facilitate analysis, verification, and optimiza-

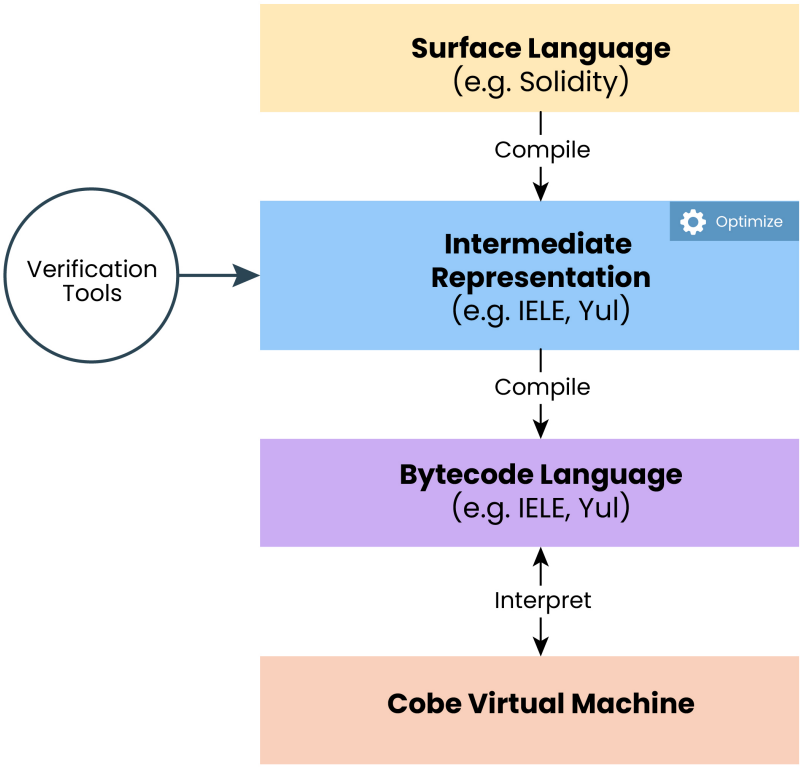

Figure 31: Cobe language stack.

tion. Typical procedures include identification of dangling references, reachability analysis (deadcode removal), approximation of gas usage, etc.

- Low-level languages (e.g., Bitcoin script, Michelson, EVM bytecode, eWASM) are closest to the (virtual) machine and are meant to be generated by compilers (i.e., not written by programmers). As they are very close to the machine, they depend heavily on its structure (e.g., stack-based vs. register-based vs. functional). Low-level programs are often referred to as bytecode. Bytecode is executed directly on a virtual machine, i.e., a runtime environment that keeps track of the data manipulated by the program and the state of its execution.

10.2 Overview of the Cobe's Language Stack

The Cobe smart contract language stack will leverage recent developments in smart contract languages while retaining alignment with common syntax that blockchain programmers are used to.

The language stack will consist of a three-layer abstraction, as depicted in Figure 31, to maximize the long-term flexibility of the system. Programmers will write contracts in Cobe's high-level language, which will be compiled to an automata-based IR on which several analyses and optimizations will take place. Finally, the optimized IR will be compiled to bytecode that runs on the Cobe Virtual Machine.

Observe that the intermediate layer will allow us to support other surface languages in future, while retaining the verification and optimization steps developed for the original language.

10.3 Surface Language

Cobe's smart contract language will need to strike a balance between easy access (to enable non-expert programmers to write contracts) and verifiability (to ensure quality and thus avoid costly bugs).

To minimize learning cost, we will base our language on Solidity (which itself follows the syntax of the very popular JavaScript) while avoiding its known vulnerabilities by following the state-of-the-art research on smart languages [12; 11].

10.4 Intermediate Representation

Our high-level language will compile down to a finite state machine-based representation. This representation will make explicit the states of the contract and which transitions are allowed from a state to another.

We will use this IR to verify and prove properties about contracts. We will target both safety (nothing bad happens) and liveness (something good eventually happens) properties

10.5 Low-level Language

Cobe's low-level language (bytecode) will be designed in conjunction with the Cobe Virtual Machine, which we describe in further detail in the next section. Our low-level language will be inspired by the EVM bytecode, to support the integration with the higher layers and make the behavior of programs more easily understandable by programmers.

11 Cobe Virtual Machine (CVM) and Middleware

Smart contracts execute on a virtual machine that provides a robust isolated execution sandbox for smart contracts. Before it can be deployed on the ledger, and thus on several instances of the Cobe Virtual Machine, smart contract bytecode will undergo a series of thorough checks which we describe in this section.

This section is organized as follows. Section 11.1 provides an overview of the blockchain Virtual Machines. A discussion on the Cobe Virtual Machine is presented in 11.2, and in 11.3 the architecture of Cobe's middleware is presented.

11.1 Overview of Blockchain Virtual Machines

In the traditional software stack, a Virtual Machine is a piece of software that provides an emulation of a real world physical machine, including a central processing unit (CPU) that executes instructions, some memory to hold code and data temporarily, and some persistent storage.

The first implementation of a decentralized virtual machine was introduced in 2014 by Ethereum [13], on which the Cobe Virtual Machine will be based. The Ethereum Virtual Machine (EVM) is a stack-based virtual machine that runs bytecode instructions to transform the 'system state' from one state to another. The EVM functions on a word size of 256-bits and a stack size limited to 1024 elements. The low-level language supported by the EVM is Turing-complete, but each program is limited by the amount of gas that is required to run each instruction. This means that infinite loops that can result in denial-of-service attacks are not possible as they would exceed limits on tolerated gas consumption. The EVM is an entirely isolated and sandboxed runtime environment. To guarantee consistent changes on the ledgers, the code that runs on the EVM does not have access to any external resources, such as a network or filesystem.

11.2 The Cobe Virtual Machine (CVM) Architecture

The basic architecture of the Cobe Virtual Machine is derived from the Ethereum Virtual Machine design, which is shown in Figure 32. There are three main memory components in the virtual machine: volatile, persistent, and immutable.

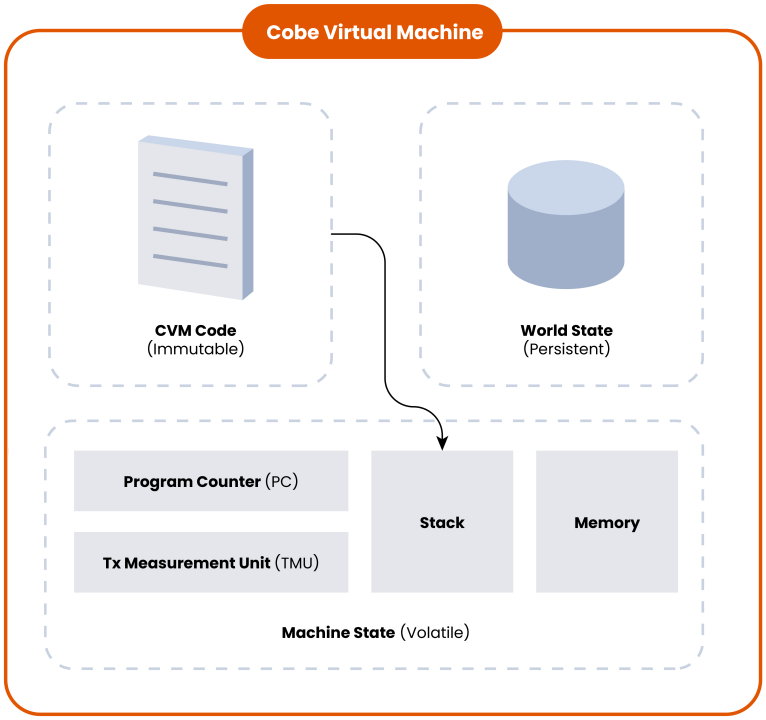

Figure 32: Basic design of the Cobe Virtual Machine.

Memory space can be further divided into volatile vs. non-volatile and mutable vs. immutable memory. The volatile section of the virtual machine memory comprises a Program Counter (PC) register, a stack, random access memory, Tx Measurement Unit (TMU), etc. They operate variable data that change frequently and are not required to be kept when a virtual machine is shut down.

The Program Counter is a special purpose register that stores the address of the next instruction to be executed by the virtual machine. The stack keeps track of function calls and their return addresses, local and parametric variables, etc. Memory in general stores code and data. The TMU calculates transaction cost and checks the availability of the fee. It is also responsible for transaction scheduling.

The immutable section of the virtual machine stores the operating system code of the virtual machine in read only state. When a virtual machine is started, the operating system is loaded from virtual Read Only Memory (ROM). Virtual ROM is a read only or immutable section of the machine that stores code/data that do not change.

The persistent section of the virtual machine is used to store information or data that must be kept after the machine has been shut down. This includes account information, world states, etc. The basic architecture of the virtual machine is presented in Figure 28.

Mutable memory consists of components discussed above, like registers, stack, memory, and persistent storage. Data stored in mutable memory is susceptible to frequent changes. Stack memory comprises 256 bits long x 1024 elements. All standard operations can be performed

on the stack, like push, pop, copy, swap, etc. Memory is a linear array of bytes. It can be addressed at the byte level; however its reading/writing size should be the same as that of the stack and persistent storage unit. It can be accessed with MSTORE, MSTORE8, MLOAD, etc. instructions. All locations in memory are always initially set to zero. Account information is stored in persistent storage. It is a key–value store that maps 256-bit words to 256-bit words. It can be accessed with SSTORE, SLOAD instructions. All locations in storage are always initialized to zero. All mutable memory units are represented in Figure 33.

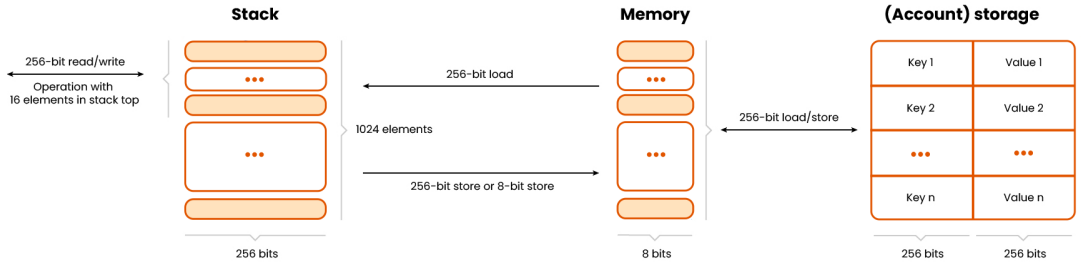

Figure 33: Mutable memory space inside a virtual machine.

11.3 Architecture of the Cobe Middleware

Smart contract bytecode will be deployed by clients on the Cobe ledger via the Cobe Middleware, which will ensure that contracts meet certain quality requirements through the use of digitally signed contracts and simulations of contracts. We give an overview of the design of the middleware in Figure 34, which we discuss further below.

11.3.1 Deployment Model of Smart Contracts

A high-level view of the deployment model of a smart contract is presented in Figure 34 and the step by step execution model is presented below.

1. Client programs submit a new smart contract (compiled bytecode) to the Cobe Virtual Machine Security Middleware
2. The Cobe Virtual Machine – Security Middleware (CVM-SM) intercepts the execution flow.
3. CVM-SM verifies the digital signature.
   (a) If the smart contract is digitally signed and its digital signature fails verification, the execution of the smart contract is stopped.
   (b) If smart contract is digitally signed and its digital signature is verified, this means the smart contract can be trusted. However, a score is added to it for further evaluation in later phases.
4. If the smart contract does not have digital signature, a score is assigned to it.
5. The Evaluation Sub System (ESS) will evaluate the smart contract on a number of different parameters. Each parameter will add a score.
6. After complete evaluation, the ESS will flag the smart contract as green, yellow, or red.

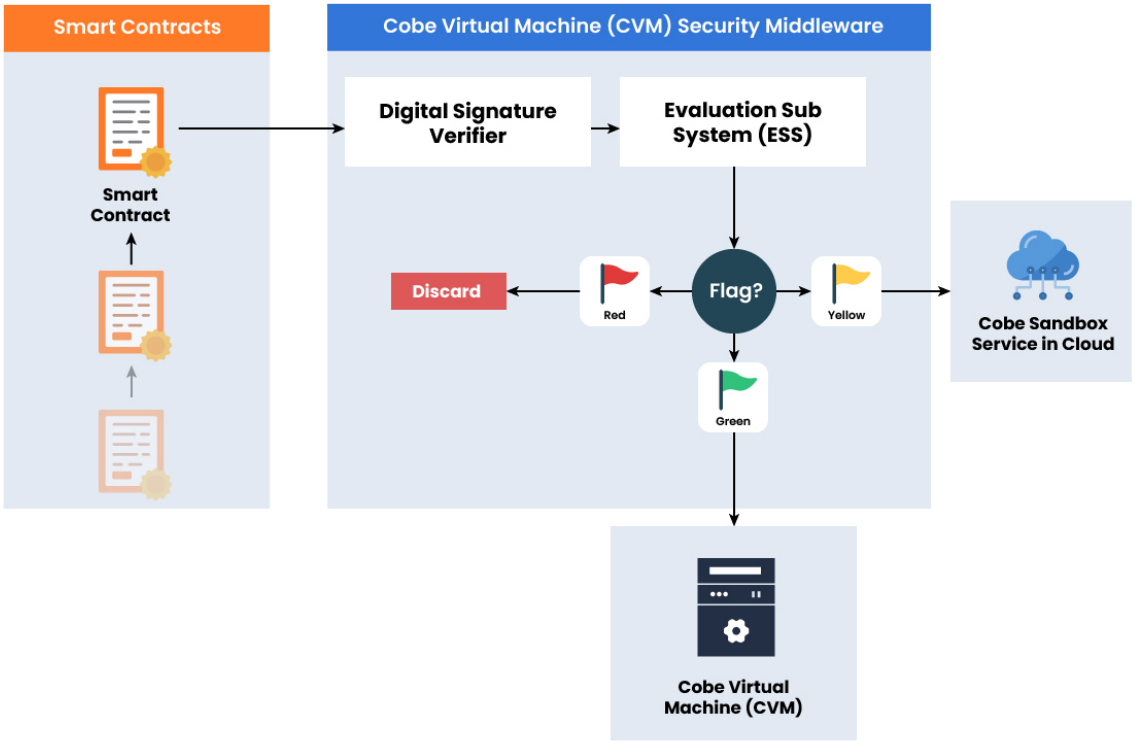

Figure 34: Cobe Virtual Machine – extended secure middleware design.

11.3.2 Digitally Signed Contract

Due to the decentralized nature of blockchain and the intrinsic immutability of smart contracts, ensuring the security and safety of a smart contract is challenging [14; 15], especially once deployed. Hence the first step the Cobe Middleware performs when receiving a request to deploy a new smart contract is to check its signature. The signature and verification are performed via the following steps.

1. The contract writer creates a new smart contract and compiles it down to bytecode using the Cobe language stack.
2. The contract writer or their proxy submits the digital contract to the Cobe Contract Signing Service (CCSS1), either manually or via API call.
3. CCSS1 is hosted in the Cobe blockchain in different locations.
4. CCSS1 verifies and, upon success, signs the contract and returns it back to the submitter. After receiving the signed contract, the smart contract can be run in the CVM.

12 Decentralized Apps (DApps)

DApp stands for decentralized app and is defined as a software system that uses distributed ledger technology (DLT), typically a blockchain, as a central hub to store and exchange information, through smart contracts. Note that it is not blockchain software able to manage a new cryptocurrency or other applications – that is, software enabling blockchain nodes, which needs different kinds of development practices and is not the subject of this section. Most present real

applications of DApps and smart contracts are intended for the management of digital currencies or tokens, which have a true monetary value. The use of DApps has also been introduced for other purposes, like notarization of information, identity management, voting, games and betting, goods provenance certification, and many others [16].

A DApp has its backend code, the smart contracts, running on a decentralized peer-to-peer network. A DApp can have frontend code and user interfaces written in any language to make calls to its backend.

In addition to decentralization, the main characteristics of a DApp are:

- Determinism: DApps perform on the blockchain the same function irrespective of the environment (specific node) in which they get executed.
- Isolation: DApps are executed in a virtual environment known as a Virtual Machine, so that if the smart contract has a bug it won't hamper the normal functioning of the blockchain network.

Another characteristic provided only by a subset of DApps, depending on the VM and on the high-level language they are written in, is Turing completeness: DApps can perform any action given the required resources.

In this section we briefly discuss the elements of a DApp and the typical architecture of a DApp running on the Cobe platform.

12.1 DApp Elements

A DApp based on a blockchain and on the web needs three elements:

- A web application to run on a web server.
- A smart contract "deployed" on the blockchain.
- A connection provider between the web component and the blockchain component.

Figure 35 shows the components of a typical DApp built using a blockchain. While many DApps are structured differently, this architecture introduces fundamental concepts.

The *DApp Client* provides the interface through which users interact with the DApp, i.e., web applications or mobile applications. A *User Wallet* is software or hardware that controls access to a user's address on the blockchain. The *Client Library* is a framework that provides a standard interface to connect client applications, user wallets, and the provider. The library manages two main tasks:

- Sending messages to the provider: allows the loading of contracts and the invocation of contract methods, both those requiring a signed transaction to be performed and those that can be transmitted directly (queries).
- Signing transactions: this is accomplished by sending the raw transaction to the User Wallet and receiving the transaction signed by the private key associated to the user's address.

The application client physically transmits the transaction to the blockchain on behalf of the user. However, it's important to note that the wallet has final say on what is signed or not, and therefore has control over any transactions that interact with the user's account. A user account is a record on the blockchain that stores the digital assets owned by a single user and is uniquely identified by an address.

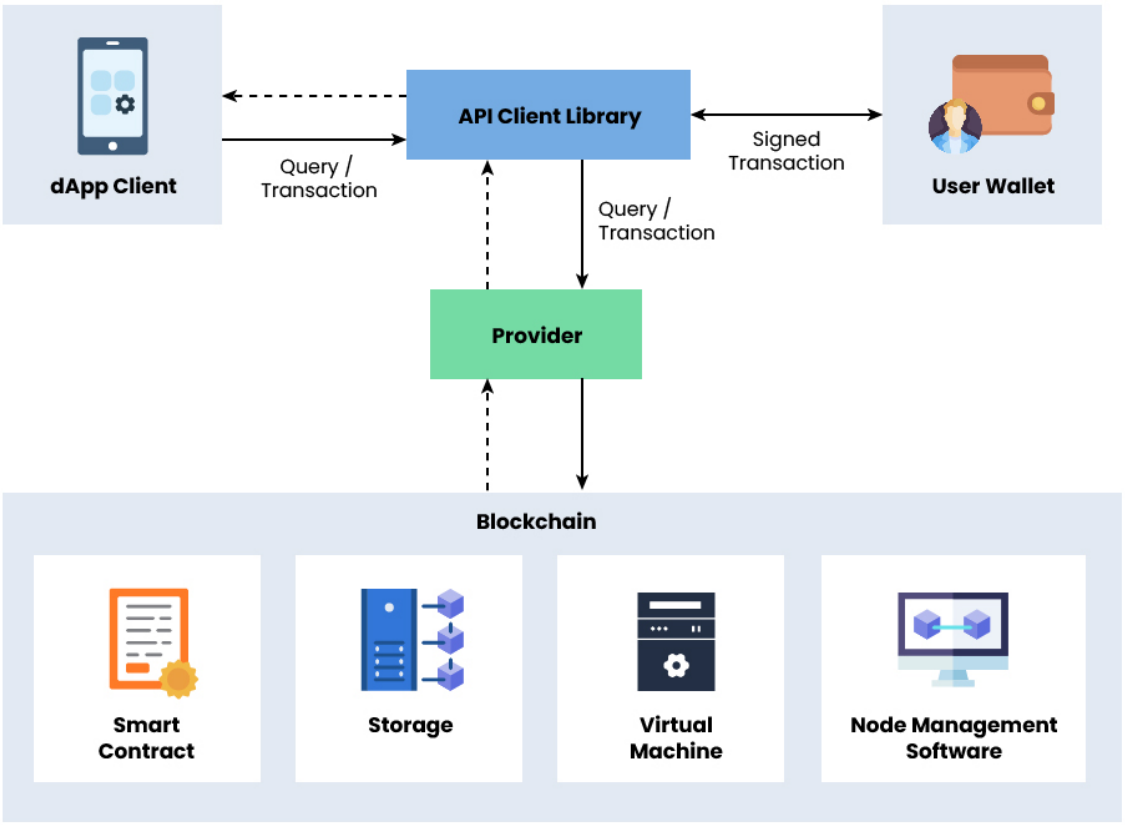

Figure 35: The typical DApp elements.

The nodes that you connect with when you need to interact with the blockchain (whether you set them up yourself or use existing ones from third-party services) are called *providers*. Once connected to the blockchain through a provider, it is possible to read the state stored on the blockchain (*query*) or to write the state after having signed the transaction using a private key (*signed transaction*).

A Smart Contract (SC) is a collection of code, deployed to a permanent location on the blockchain and immutable, that defines the core logic for a DApp and the related state of the application. The virtual machine executes the logic defined in the smart contract and processes the state changes.

12.2 Cobe DApp Architecture

The proposed Cobe DApp architecture is shown in Figure 36. This is a general-purpose architecture, showing all the possible components. In specific applications, some components might not be needed, and should not be considered.

As explained earlier in the paper, Cobe's blockchain architecture is dual-sided. On one side there is the permissionless blockchain, on the other the permissioned blockchain, interacting through the relay, shown in the middle.

The permissioned blockchain has three kinds of nodes:

- **Validators**, the nodes running the system, managed by the key consortium participants. These nodes hold a copy of the blockchain, validate transactions, group them in blocks, and decide to add blocks to the blockchain using the CPoA consensus mechanism.

- **Full nodes**, the nodes holding a copy of the blockchain, and able to receive, validate and broadcast transactions but not to participate in the consensus mechanism. These nodes are managed by organizations that have obtained the permission to do so but are not (yet) full members of the consortium.
- **Relay nodes**, at least one on each blockchain. These nodes can communicate with each other via cross-blockchain communication protocol (CBCP), allowing interaction between the two blockchains.

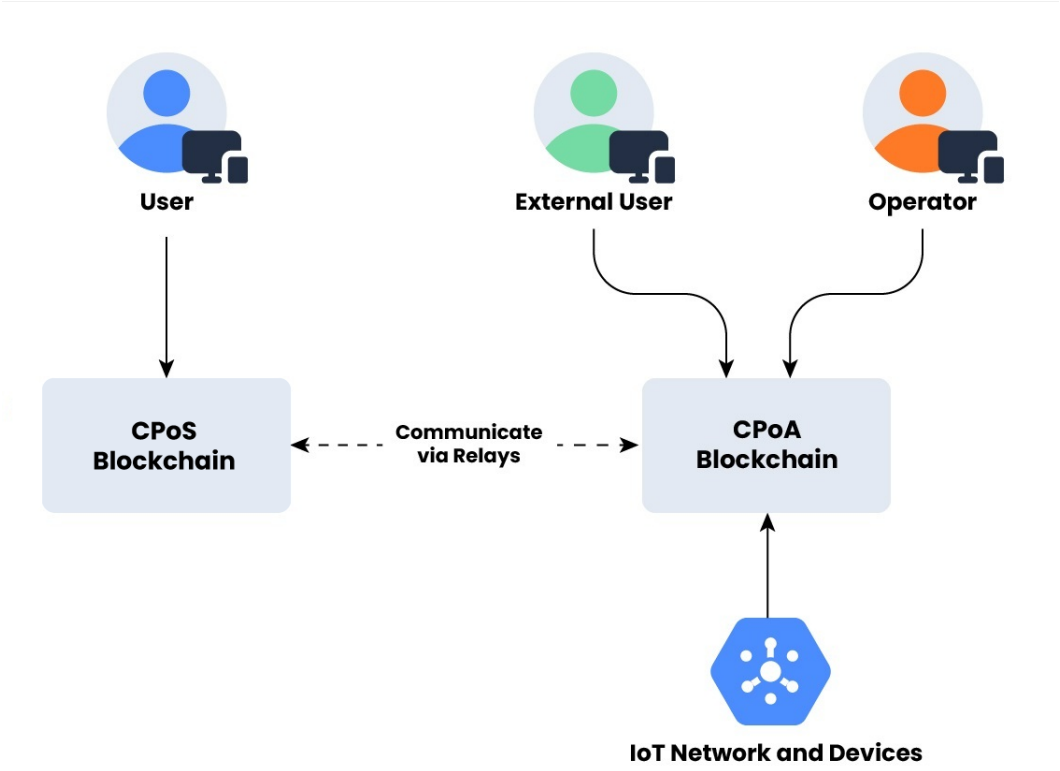

Figure 36: Cobe's proposed DApp architecture.

The validators are run by the organizations of the consortium, which should be independent from each other, to avoid the risk that a single organization might try to falsify the blockchain data, or simply decide to stop supporting the system.

It is important to assess the reasons why validators take on the burden of managing the blockchain. The main reasons are either to propose the blockchain as a service to customers for a profit or being involved in the management of the DApp(s), which in turn can provide a benefit to the validators. This benefit might be direct, coming from the sale of products or services, or indirect – think for instance to a public body promoting some service linked to its mission

An assessment of validators is necessary before they are added to the system, and includes a risk analysis, estimating the probability validators might turn off their node within one or more years, and thus computing the minimum number of them needed to guarantee the persistence of the DApp. Clearly, in the event some validators leave, this should trigger a search for new validators, to maintain the DApp's stability.

The actors that operate on the blockchains are:

- **Operators**, who are enabled to send transactions changing the blockchain state. Operators use terminals and GUI software that are part of the overall system and belong to organizations participating to the system.
- **External users**, who can access the system nodes in read-only mode, using standard terminals and with software provided by the system.
- **IoT devices**, which can send transactions, writing data automatically collected on the blockchain.

Regarding external users, there are two possibilities:

- Everyone can access the blockchain – in this case, the nodes (validators and full nodes) allow public access to the SC interfaces.
- The access is limited to authorized users – in this case, authentication and access control must be provided by the nodes (both validators and full nodes) before users can access the DApp.

Cobe's CPoS blockchain also includes validators, but these are chosen dynamically according to the money they put at stake and other quality parameters. So, in principle, all nodes start equal, and we identify only one kind of actor, that we call User.

The key components of Cobe's architecture are:

- CPoS and CPoA blockchains as shown in Figure 36.
- An external system, called the App System, holding the data and applications not residing in the blockchain.
- The terminals of operators and external users (top and bottom of the figure), running DApp software providing the user interface and able to manage the private keys of operators.
- A system to perform identity management and access control, integrated into the App System, and possibly also using an SC.
- Links to IoT devices that send data to the blockchain or receive commands from the system.

All nodes hold the blockchain-enabling software, which includes the Cobe Virtual Machine, running SCs. The SC bytecode, endowed with its permanent data (storage), is stored in the blockchain and is loaded into the node memory for its execution. All the nodes execute every SC, and execution results must be the same for all nodes, hence the impossibility for SCs to access the external world. They can access only their data and other SCs stored in the blockchain, which are the same in all nodes.

12.2.1 App System

Another key DApp component is a software system running on mobile devices and/or on servers, possibly on the cloud, which we call the 'App System'. It holds information that cannot stay in the blockchain because it is too large, or for privacy reasons. The App System exchanges information with users and with external systems and devices, and performs business computations.

Of course, it is also able to send transactions to the blockchain, having a direct connection with a node and being the owner of an address and of the corresponding private key. If the DApp must hold large amounts of information, such as documents and images, these documents are stored off-chain on one or more document management systems (DMSs), by the App System. The hash digest of the document and a link to retrieve it can be stored in the blockchain, guaranteeing the date of the document and its integrity. This approach is also called the "off-chain data storage" pattern. In the case of sensitive data stored off-chain, the App System also takes care of managing access rights to it, providing the information only to qualified users. Saving data in this way is compatible with privacy regulations, because no actual data is stored in a transparent and immutable medium such as the blockchain. Moreover, huge amounts of data can be managed, stored, and certified, despite the relatively limited room available in blockchains, most of which were never intended to substitute a DMS or a database. In fact, storing large amounts of information on a permissioned blockchain is not viable for the following reasons: (i) bulk data means big transactions to write them into the blockchain, which in turn means overloaded communications and reduced performance; (ii) the computation needed to assemble and communicate the block with these transactions again means reduced performance of the system; (iii) the size of the blockchain, which is an append-only repository, would quickly become huge and impair efficient data retrieval. To conclude this section, we stress that the App System is not necessarily a single, centralized system; nor does it have to manage a single, centralized database or DMS. The App System is a service that, if needed, can run on several physical or cloud servers. Operators who need to store a document can directly specify the URL of the DMS in which to store it, and there can be many of them. For instance, each organization storing data might manage its own DMS, including granting access permission to it. What is important is that the data can be accessed by whoever is entitled to access it and that the access permission is given by the owner of data, possibly also through the blockchain itself. Also, the database holding the system data might be a decentralized one, such as IPFS.

12.2.2 Terminals and Apps

This component includes the applications, running on PCs and/or mobile terminals, that enable interaction with human users. For external users, it can be a simple app, able to connect to a node or to an authentication server and to show to the user the requested information, gathered from Cobe's blockchain and/or from the App Server.

For operators, the app includes a wallet – that is, software able to generate and store the private key associated to the operator's blockchain address, to create transactions, sign them with the private key and send them to a node. The operator's private key is unblocked by a password and possibly by the ownership of a specific mobile phone. In this way, the identity of the sender of the transaction is guaranteed.

The operator's app will also facilitate any data input and control operations the operator is in charge of. Depending on the specific applications, the app is able to exchange data with Cobe's blockchain (by sending transactions) and/or with the App System.

12.2.3 IoT Devices

The Internet of Things (IoT) is the extension of the internet to connected physical objects that can be monitored, controlled, or interacted with, to enable ubiquitous industrial services. Examples of IoT industrial use are freight transportation – automatically registering temperatures, positions, arrival times, and status of shipping containers and trucks as they move; tracking components in aircraft, automotive, or other industries, which is critical for both safety and regulatory compliance; supply chain and Digital Product Passport digitalization and control; logging of operational maintenance data; and many others.

The interaction between blockchain and IoT has been proposed since the introduction of SCs, for two main reasons. The first is because the blockchain can provide IoT devices with security and the ability to be tamper-proof. The second is the fact that a blockchain is distributed, and an IoT device can connect to any of its nodes, avoiding the bottleneck of a single access point. An IoT sensor can be provided with an address, a private key, and a connection to the blockchain, and thus be able to send its data through a transaction, which guarantees timestamp and immutability of the registration. To this purpose, many initiatives aim to develop and field blockchains specifically suited to IoT management, such as IOTA and IoTex.

Things are not so simple, however, because the number of IoT devices can be huge, and the rate of transactions coming from each of them can be high, stressing both the throughput and the size of the blockchain. To solve this issue, sets of IoT devices are connected to some flexible and robust cloud computing environments, able to process and manage IoT services. This solution is called "Cloud of Things" (CoT), and its integration with the blockchain (BCoT) is the subject of a large amount of research, aptly reviewed and summarized by Nguyen et al. [17].

In Figure 36 we show IoT devices directly connected to Cobe's blockchain network. The IoT data is typically not entirely registered on the blockchain, but only a digest of it is written. If needed, the raw data can be stored in the cloud.

12.3 Frontend and Smart Contract Communication

The frontend needs to communicate with the SCs and be able to invoke their functions. Every node in Cobe's network keeps a copy of all SC states on the blockchains, including the code and data associated with every SC. When we want to interact with the data and code on a blockchain, we need to interact with one of these nodes. To broadcast a new transaction, there are two possible solutions: set up a node that runs the Cobe blockchain software or use nodes provided by third-party services (for instance, the services of Alchemy or Infura for Ethereum Mainnet). These nodes are known as *providers*. By connecting to the blockchain via a provider node, it is possible to read a status stored on the blockchain. To send a transaction that changes a status on the blockchain instead, it is necessary to first "sign" the transaction using the private key. To ease the tasks of DApps managing the keys and signing transactions, a tool called a *signer* is often used (for instance, Metamask). A signer stores a user's private keys in the browser and intervenes whenever the frontend needs the user to sign a transaction, possibly also providing the gas to pay for the transaction.

Figure 37 shows the architecture of a DApp, highlighting the communication between the frontend and the backend, meaning the SCs running on the Cobe blockchain. To interact with the DApp's SCs, a user needs an internet connection. He/she has access to the DApp frontend, whose UI typically runs on a web browser. A user can also store his/her private keys on the browser via a signer, shown on the right. In the middle, the provider, whether it is a node you set up or an existing one from third-party services, allows the interaction with the SCs. The Cobe Virtual Machine executes the logic defined in the smart contracts and processes their state changes on the blockchain.

Finally, the server component stores data that cannot be stored in the blockchain.

12.4 Cross Chain Decentralized App (CC-DApp) utilizing Cobe's Dual Blockchain Interoperability Architecture

As defined in Section 12, a decentralized App (DApp) is an application whose backend comprises smart contracts that run on a decentralized peer-to-peer (p2p) network like blockchain. A DApp can have a frontend or user interface (UI) through which it can be operated. The Cobe

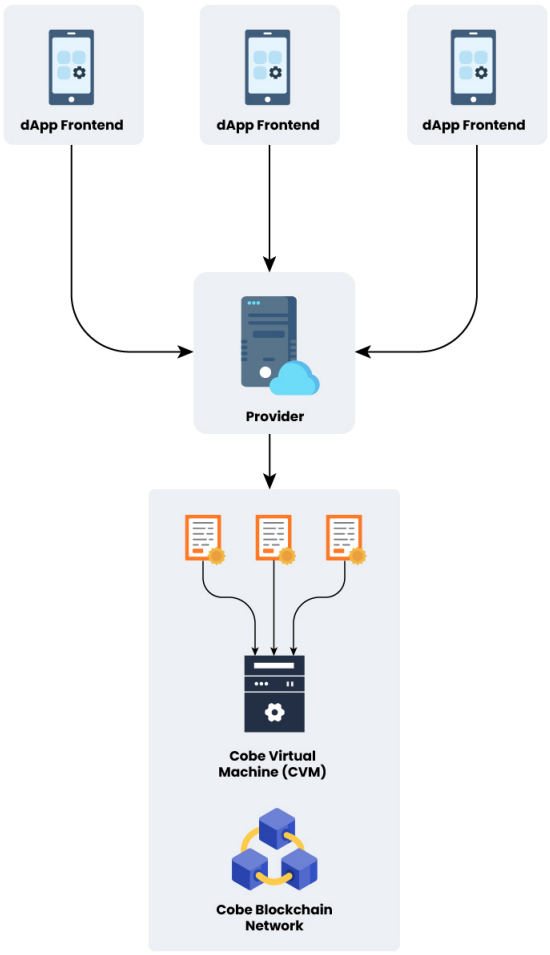

Figure 37: Frontend and backend communication of a DApp.

blockchain network has introduced the Cross Chain Decentralized App (CC-DApp). A CC-DApp is a hybrid application. It can utilize Cobe's dual blockchain interoperability architecture to operate on CPoS and CPoA chains simultaneously. In this section, we present and explain how a CC-DApp can utilize Cobe's dual blockchain interoperability architecture to operate on both blockchains at the same time.

Consider a logistics DApp through which users book their transportation. This DApp stores and processes shipment information, like arrival/departure time, date, etc., but it is also required to maintain sensitive user information like payment details, e.g., credit card information. Therefore, the CC-DApp hybrid application can be used to achieve high confidentiality by using the CPoA chain while functioning with the CPoS blockchain at the same time.

A CC-DApp may comprise several smart contracts, some hosted in one blockchain and some hosted in the other blockchain. Let SC_{A1}, SC_{A2}, ..., SC_{An} be smart contracts hosted in CPoS based Blockchain-A, while SC_{B1}, SC_{B2}, ..., SC_{Bn} are smart contracts hosted in CPoA based blockchain-B. So a hybrid CC-DApp will use smart contracts from both blockchains at the same time whenever required. The CC-DApp fully utilizes Cobe's dual blockchain interoperability architecture presented in Section 9.2. It will first register itself with one of the blockchains – for example Blockchain-A – which is basically a CPoS-based blockchain. Blockchain-A (a CPoS blockchain) will be considered as the primary blockchain or host or local blockchain for the CC-DApp, while Blockchain-B (CPoA) will be considered as the remote blockchain for the

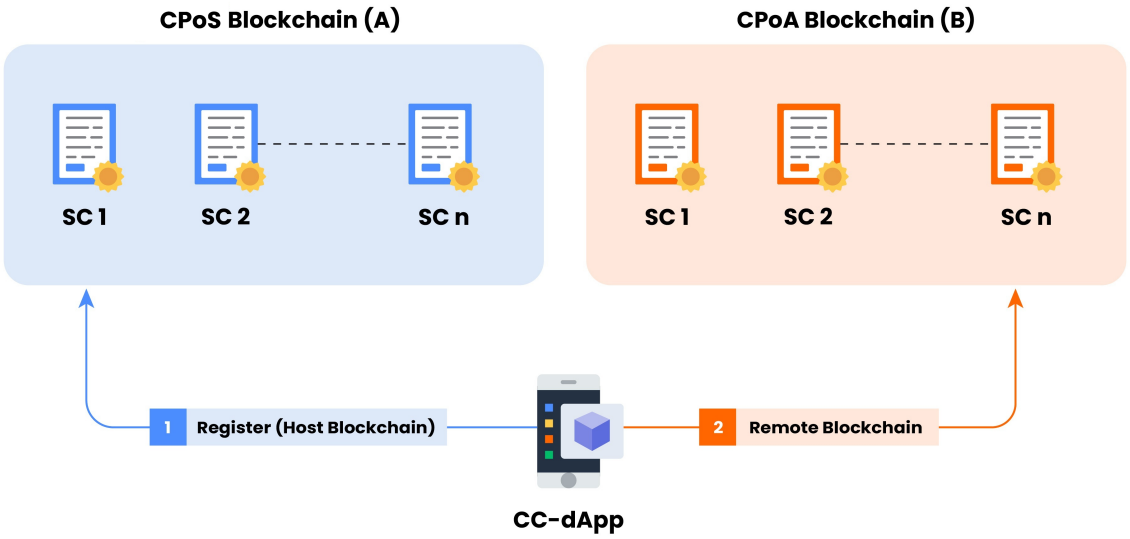

Figure 38: Cobe's Cross Chain-Decentralized App (CC-DApp); SC: smart contract.

CC-DApp. There can be more than one remote blockchain for a cross chain DApp; however, there will be only one primary or host blockchain for every CC-DApp, as shown in Figure 38.

13 Oracles

In this section we briefly discuss oracles. A blockchain may need to share or access information from outside of the network. For example, a smart contract needs the stock price of a security product that is required by the contract to release dividend payments. In such situations, oracles can be used to provide external data to smart contracts. An oracle is an interface that delivers data from an external source to smart contracts or the blockchain. Based on the requirements and needs, oracles can provide different types of data or information to the blockchain.

13.1 How Oracles Work

In this section we will present the detail working methodology of a generic oracle. The information flow between the different components is shown in Figure 39.

1. **Smart Contract Requests Data**
    - Technical Aspect: The smart contract, deployed on the Cobe blockchain network, executes a function call that specifies the type of external data required. This call is encoded in the blockchain's native scripting language.
    - Trigger Mechanism: The request is often triggered by a specific condition or event defined within the smart contract's logic.
2. **Request Evaluation and Execution**
    - Processing Request: The oracle evaluates the request to determine the type of data and the source from which it must be fetched. This evaluation involves parsing the request parameters and identifying the appropriate data source.
    - Data Fetching: The oracle then executes the necessary protocols to request the data. This might involve constructing API calls or database queries as per the requirements.

3. **Methods of Data Retrieval**

   - API/Web Service Calls: The oracle makes HTTPS requests to third-party APIs or web services to fetch the required data. This is common for financial data, weather information, etc.
   - Database Access: For more static or historical data, the oracle might access a specific database. This involves SQL or NoSQL queries to retrieve the data.
   - Inter-Blockchain Communication: In some cases, the oracle might fetch data from another blockchain. This involves blockchain interoperability protocols and possibly cross-chain smart contracts.

4. **Generating Cryptographic Proof**

   - Notarization Process: Once the data is retrieved, it is sent to a notary service (which can be an integral part of the oracle or an external service).
   - Digital Signature Creation: The notary generates a digital signature for the data using cryptographic algorithms. This signature acts as a proof of authenticity and integrity of the data.

5. **Sending Data Back to the Oracle**

   - Data Packaging: The retrieved data, along with its cryptographic proof, is packaged by the oracle in a format that is compatible with the smart contract.
   - Preparation for Blockchain Integration: The data is now ready to be integrated back into the blockchain environment, ensuring it adheres to the network's protocols and standards.

6. **Delivery to the Smart Contract**

   - Final Transmission: The oracle sends the data package back to the smart contract on the blockchain.
   - Smart Contract Processing: Upon receiving the data, the smart contract verifies the cryptographic proof to ensure data integrity and authenticity. Then, it processes the data as per its programmed logic, which could lead to the execution of subsequent actions or transactions.

13.2 Types of Blockchain Oracles

In this section we will present the different types of oracles supported in the blockchain ecosystem. There are two main types of oracles: inbound oracles and outbound oracles.

13.2.1 Inbound Oracles

This class represents oracles that receive incoming data from external services and feed it to the smart contract.

13.2.2 Outbound Oracles

This type, also called reverse oracles, is used to send data out from blockchain smart contracts to the outside world.

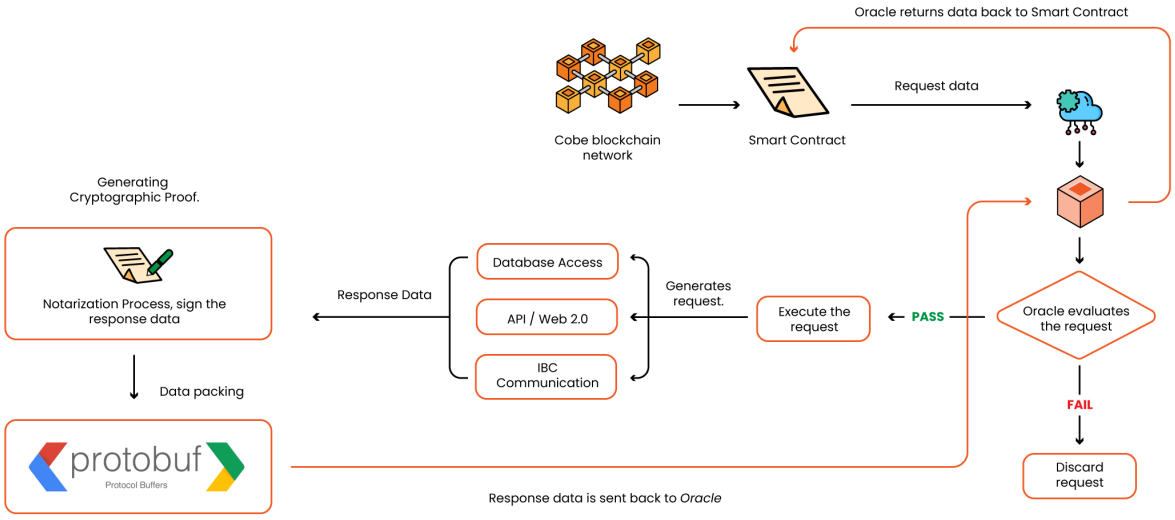

Figure 39: Data flow between smart contact, oracle, and attestation service.

14 Blockchain Security

In a decentralized distributed network, maintaining a synchronized valid state of the ledger is a difficult task. In this section we discuss some of the key security consideration for both Cobe and blockchain networks in general.

14.1 Brief Overview of Blockchain Attacks

Recent studies on blockchain networks have explored different attack scenarios. Saad et. al. [18] presented attacks on blockchain networks from different perspectives, such as blockchain structure, blockchain applications, attacks on the peer-to-peer network, etc. In this section, we briefly touch on the common attacks on blockchains network.

1. **Double Spending Attack:** In a double spending attack, an attacker tries to spend the same cryptocurrency unit twice by creating two conflicting transactions. This attack is particularly relevant in cryptocurrency networks that use proof-of-work consensus mechanisms.

2. **51% Attack:** A 51% attack occurs when a single entity or a group controls more than 50% of the network's computational power in a proof-of-work blockchain. This control allows the attacker to manipulate transactions, potentially leading to double spending and rewriting transaction history.

3. **Sybil Attack:** In a Sybil attack, an attacker creates multiple fake nodes to gain control over a blockchain network. This can be used to manipulate consensus, disrupt communication, and potentially launch other attacks.

4. **Smart Contract Vulnerabilities:** Flaws in smart contracts' code can lead to vulnerabilities that attackers exploit to drain funds or manipulate the contract's intended functionality. This was demonstrated by the infamous DAO attack.

5. **Eclipse Attack:** An eclipse attack involves isolating a target node from the rest of the network by surrounding it with malicious nodes. This can lead to the victim node receiving manipulated information or being excluded from consensus.

6. **Routing Attacks:** Attackers can manipulate the routing of data packets between nodes to intercept or alter communication. This can compromise the integrity and confidentiality of data.

7. **Denial of Service (DoS) and Distributed DoS (DDoS) Attacks:** These attacks aim to overwhelm the network with a flood of traffic, rendering it slow or completely unresponsive.

8. **Consensus Algorithm Exploits:** Different consensus algorithms have their own vulnerabilities. For example, proof-of-stake networks can be susceptible to long-range attacks where an attacker creates a fork of the blockchain from an early point and accumulates a larger stake over time.

9. **Timejacking:** An attacker can manipulate the network's time to control block timestamps, potentially altering the order of transactions.

10. **Front-Running:** In blockchain networks with transparent transactions, attackers can try to exploit the time delay between transaction submission and confirmation to manipulate transactions in their favor.

With respect to the consensus protocol, one of the most common attacks on a blockchain network is a fork. A fork represents a situation in which nodes in the network have diverging views about the state of the blockchain network. Forks can be created intentionally, or unintentionally through protocol malfunction. Forks created intentionally are of two types – one that is created for malicious intent and the other to modify the rules of a blockchain, which are known as hard forks and soft forks. Other forms of inconsistencies can also occur with the consensus process that can leave valid blocks out of the blockchain. The first form is a 'stale block' and the second one is an 'orphaned block'. A stale block is a block that was successfully mined but is not accepted in the current best blockchain. An orphaned block is a block whose parent block's hash field points to an unauthenticated block that is detached from the blockchain

14.2 Proof of Turn (PoT) Protocol – Security Considerations

Both Cobe's permissionless Concurrent Proof of Stake (CPoS) and permissioned Concurrent Proof of Authority (CPoA) chains utilize its novel 'Proof of Turn (PoT)' protocol. Proof of Turn (PoT) reduces the time required to process transactions by enabling advanced block scheduling. PoT includes a number of salient features that enhances the security of the protocol.

14.2.1 Key Security Considerations

1. **Forkless Operation:** PoT's advanced block scheduling enables forkless chain operation, which is more secure as it eliminates the risks associated with forks, such as double spending.

2. **Early Scheduling:** Validators are determined through an early scheduling approach, utilizing synchronized random numbers and Cellular Automaton (rule 30), ensuring transparency and reducing the risk of forks.

3. **Longest Chain Rule:** In case of a fork, the protocol adheres to the longest chain rule, which is a standard method in blockchain for validating the transaction history.

4. **Slashing Mechanism:** PoT incorporates a slashing mechanism to penalize or remove dishonest nodes, thereby enforcing honest participation within the network.

The PoT protocol effectively mitigates against common blockchain attacks by its design and the implementation of additional measures that ensure the integrity and security of the blockchain. These security considerations make PoT a significant contribution to the field of blockchain protocols.

14.3 Security of Smart Contracts

We tackle code vulnerabilities by deploying tools for static and dynamic analysis of smart contracts. When a smart contract is loaded onto the Cobe Virtual Machine for execution, it will be analyzed by the evaluation subsystem (ESS). The core responsibility of the ESS is to analyze the smart contract for common errors, divergence from best practices, and suspicious code. Based on this analysis, the ESS updates the assigned score of the smart contract and assigns one of the following:

1. Green Flag: the smart contract is benign and can be executed safely
2. Yellow Flag: the smart contract is suspect and must be submitted to the Cobe Cloud Sandbox Service (CCSS2) for detailed analysis.
3. Red Flag: the smart contract is malicious and should not be executed.

Depending on the flag, the smart contract can be monitored at runtime to ensure its execution complies with its prescribed logic and to prevent costly errors.

The process is presented in Figure 34 in Section 11.3.

14.4 Cobe's Blockchain: Threshold Signature Scheme (TSS)

Cobe has planned to utilize a threshold signature scheme to enhance its security and operational efficiency of the blockchain network. Below we presented several areas where threshold signature scheme will be helpful

- **Sonic Wallet Security:** Cobe will utilize threshold signature scheme (TSS) to enhance the security of Sonic wallets. By using TSS, signing keys can be distributed across multiple parties to protect against the compromise of a single key.
- **Cross-Chain DApps:** When deploying DApps that operate across both CPoS and CPoA chains, TSS can be instrumental in managing permissions and actions that require consensus across the chains. For instance, a DApp may require actions to be taken when certain conditions are met on either chain. TSS ensures that these actions are only executed after achieving the required level of consensus, effectively coordinating between the two different governance models.
- **Multi-Chain Smart Contracts:** TSS can greatly enhance the functionality of smart contracts that span multiple chains. It can serve as a trustless method for verifying that a smart contract on one chain has met the pre-defined conditions before executing a related smart contract on the other chain. This multi-signature approval process can be crucial in automated cross-chain interactions, ensuring that contracts act in harmony and only when appropriate thresholds of agreement are met.
- **Oracle Services:** Oracle services provide external data to blockchains, and their reliability is crucial. TSS can be used by oracles to sign the data they provide to the blockchains, ensuring that the data is not considered valid until it receives the necessary number of signatures from trusted entities. This can be particularly useful when combining data from various sources or when verifying real-world information that affects multiple chains.

- **Staking Pools and DeFi:** Cobe will apply TSS to decentralize transaction signing within staking and DeFi protocols, adding an extra layer of security and system robustness.
- **Identity Management:** Cobe will integrate TSS into identity management systems to protect privacy and enhance security without revealing individual signers.
- **Secure Voting Systems:** Cobe will make use of TSS in blockchain-based voting systems to ensure votes are valid only when a threshold of signatures is reached, thus preventing unauthorized changes.
- **Governance Protocols:** Cobe plan to utilize TSS to enforce governance by enacting protocol or smart contract changes only after achieving consensus through a threshold of stakeholder approvals.
- **Interoperability Protocols:** Cobe will harness TSS for secure interoperability, enabling multi-signature operations that underpin cross-chain agreements and transfers.

14.5 Cobe's Blockchain: Quantum-Resilience Strategy

Current blockchain infrastructures predominantly utilize elliptic curve cryptography (ECC) for security, particularly the Elliptic Curve Digital Signature Algorithm (ECDSA), over the RSA algorithm. Wallet addresses are derived by hashing the public keys, which are generated from points on an elliptic curve. The security of these methods is predicated on the computational infeasibility of certain mathematical problems: the integer factorization problem for RSA and the discrete logarithm problem for ECC.

Quantum computers, however, are predicted to efficiently solve these problems using algorithms like Shor's algorithm, which can compute discrete logarithms and integer factorizations in polynomial time, rendering RSA and ECDSA insecure.

14.5.1 Transitioning to Quantum-Resistant Cryptography

The National Institute of Standards and Technology (NIST) has initiated efforts to standardize post-quantum cryptographic algorithms that are secure against quantum computer-based attacks. Among the algorithms selected for standardization, CRYSTALS-Kyber has been chosen for key encapsulation mechanisms, and CRYSTALS-Dilithium, FALCON, and SPHINCS+ for digital signatures.

CRYSTALS-Kyber

CRYSTALS-Kyber, a key encapsulation mechanism, is based on the hardness of solving the Learning With Errors (LWE) problem, a problem believed to be secure against both classical and quantum computers.

Digital Signature Algorithms

- **CRYSTALS-Dilithium** is based on the hardness of the Module-LWE problem, an adaption of LWE for module lattices.
- **FALCON** uses the hardness of the Shortest Vector Problem (SVP) in lattice cryptography.
- **SPHINCS+** is a stateless hash-based signature scheme, relying on the security of hash functions, which do not have known vulnerabilities to quantum attacks.

14.5.2 Cobe's Approach to Quantum-Secure Blockchain

Cobe will initially deploy ECDSA for digital signatures but has engineered its system for a seamless transition to the NIST-approved post-quantum algorithms. The ECDSA public keys will continue to be hashed for wallet addresses. In anticipation of quantum-resilience requirements, these addresses will be linked to identifiers compatible with post-quantum algorithms for signature verification purposes.

$$\text{Address} = \text{Hash}(\text{Post-Quantum Public Key})$$

The mapping from ECDSA to post-quantum algorithms will involve a dual-key strategy:

$$\text{Dual-Key}(PK_{ECDSA}, PK_{PQ}) \rightarrow \text{Address}$$

where PK_{ECDSA} is the current public key and PK_{PQ} is the post-quantum public key. This dual-key approach will allow Cobe to maintain current operations while being prepared to switch to a quantum-resistant algorithm instantly.

14.5.3 Security Against Quantum Threats

Cobe's strategic implementation ensures robust protection against quantum threats. By preparing the infrastructure for an upgrade to post-quantum cryptography, Cobe not only secures its current operations but also future-proofs the blockchain against evolving quantum computational capabilities.

15 Cobe Blockchain Economics

In this section we first present Cobe's ecosystem coin stack; we present the different types of coins offered in Cobe's blockchain solution. Then, in Section 15.2, we present Cobe's transaction fee structure for both its CPoS and CDPoA chains. In Section 15.3, we present and discuss Cobe's inflation rate economics, which includes the CBE inflation schedule, CBE coin burning mechanism, and staking reward.

15.1 Cobe Ecosystem Coin Stack

Below is a summary of the different coins and tokens that are part of Cobe's ecosystem. Each one has been discussed in detail in the relevant sections of the paper, which can be referred to for more information.

Table 6: Cobe ecosystem coin stack.

Name	Description
CBE	Cobe's primary native coin. It can be used for staking on Cobe's blockchain and across the Nucleus platform to receive a fee discount. Voters must hold CBE to take part in governance voting on both Cobe's blockchain and the Nucleus platform.
CBS	This is Cobe's collaterally backed stable coin, which will be pegged at a 1:1 ratio with the USD.
CBR-1	Cobe's non-fungible redemption coin. This coin can be used on the Nucleus platform to keep a ledger of funds owed to each user from those held in smart escrow vaults.
CB-100	Cobe's native fungible token, which can be used by DApp developers to create their own applications and currencies on Cobe's blockchain.

15.2 Transaction Fees

In blockchain systems, users are required to pay transaction fees for each transaction they submit. Transaction fees are an important component of a blockchain system, to build a robust economic model. It is considered as necessary in many situations; for example, it prevents individual users from consuming too many resources and creating Distributed Denial of Service (DDoS) attacks. It also helps validators to get incentives for the transactions they process and store in the ledger.

The Cobe blockchain ecosystem is based on Cobe Concurrent Proof of Stake (CPoS) and Concurrent Proof of Authority (CPoA) chains. CPoS is basically a permissionless chain while CPoA is Cobe's permissioned blockchain. We have developed different transaction fee models for each chain, which are presented in the next sections.

15.2.1 Transaction Fee for CPoS Chain

The Cobe Concurrent Proof of Stake (CPoS) chain uses a relatively simple model that covers transaction processing fee and storage of transactions. Cobe CPoS transaction fees can be calculated using Eq. 9. Equation 9 includes protocol parameters (ϱ and ϑ) and transaction size, which is measured in bytes.

$$TxFee_{CPoS} = \varrho + size(tx) * \vartheta \tag{9}$$

Protocol Parameters (ϱ and ϑ):
ϱ and ϑ are protocol parameters. ϱ refers to a fixed fee that must be charged for each transaction regardless of the size of transaction. This parameter is basically used to deter DDoS attack. ϑ refers to a transaction fee paid for each transaction byte.

Transaction Size (tx):
Transaction size refers to the length of transaction expressed in bytes.

15.2.2 Transaction fee for CPoA Chain

CPoA offers a fixed transaction fee structure for users. This helps applications (DApps) to calculate all the associated costs related to transaction processing in advance to predict their revenue. A fixed transaction fee helps to avoid unwanted variations. For example, in the case of a provenance application that performs a large number of micro-transactions on a daily basis,

a small fluctuation in transaction fees can result in dramatic variations in total transaction fee, which makes it impractical for most businesses.

15.2.3 Stabilized Elastic Fee Model for CPOA Chain

Cobe has introduced a stabilized elastic fee model, which will help customers to pay for what they process on the blockchain. Consider a scenario in which two DApps perform different types of transactions. For example, DApp 'A' performs micro-transactions, e.g., scanning a product barcode and storing its information on the blockchain, while a decentralized Artificial Intelligence (AI) app utilizes the blockchain for some high-end task. As the nature of the transactions performed by each app is different, their transaction fees should also be different. However, all transactions generated for the same task will be fixed.

The stabilized elastic fee model helps Cobe users pay for the type of workload (transactions) they perform while enjoying a fixed transaction fee. A mathematical representation of the stabilized elastic fee model is presented in Eq. 10. In the equation, η represents the stability coefficient that is used to peg the cost of transaction to the dollar value, ω represents transaction weight from 0 to 1, and $CBE_{discount}$ represents the discount in terms of CBE coins (if any).

$$TxFee_{cpoa} = (\eta * \omega) - CBE_{discount} \tag{10}$$

Consider a scenario in which a DApp executes a large number of micro-transactions on a daily basis. If the corresponding weight of each micro-transaction is 0.4 and the value of the stability coefficient η is 1, then the transaction fee for a single micro-transaction will be \$0.4 or equivalent CBE coins (if $CBE_{discount}$ is ignored) for simplicity.

Table 7: Summary of weights assigned to each type of transaction.

Weight (ω)	**Value**
Nano-transaction	0.2
Micro-transaction	0.4
Mini-transaction	0.6
Transaction	0.8
Macro-transaction	1.0

15.3 Cobe's Inflation Economics

As with fiat, a well-designed blockchain ecosystem must have carefully considered measures in place to appropriately manage its currency's inflation.

Similarly to all major blockchains, such as Ethereum, Solana, Polkadot, and Cardano, Cobe will mint new coins on an annual basis. This is required to pay staking rewards, CBE holders, and validators while fueling the growth of the ecosystem. As with other cryptocurrencies, a certain number of coins will be accidentally lost each year by their holders, so minting is necessary to substitute these coins in the ecosystem. Below in Table 8 we present the terminology required to understand Cobe's inflation model.

Table 8: Inflation terminology.

Terminology	**Definition**
Total Current Supply (TCS_{CBE})	The number of CBE coins (either locked or unlocked) that have been created to date, excluding any burned coins.
Inflation Rate [%] (IR)	The annual rate of increase in the Total Current Supply of CBE.
Initial Inflation Rate (IIR)	The inflation rate when it is initially implemented. The initial target range for this is 8–10%.
Inflation Reduction Coefficient (IRC)	The rate at which inflation falls over time. The target range for this coefficient is 12–18% per annum.
Long-term Inflation Rate [%] (LIR)	The expected long-term inflation rate. The target range for this is 1.5–2% per annum.
Staking Reward (SR)	The annualized rate of return given to CBE coin holders for staking their coins.
CBE Locked (CBE_{Locked})	To receive staking rewards, a user must lock their CBE coins for a minimum number of epochs. CBE_{Locked} represents the number of CBE coins that are locked.
Minimum Staking Period (ϵ)	The minimum number of epochs for which CBE coins are required to be locked to receive staking rewards.

15.3.1 CBE Inflation Schedule (CIS)

The CBE inflation schedule determines the annual inflation rate, which is set to decrease year on year (Figure 36). The inflation rate for the new cycle (year) can be calculated as follows:

$$IR_{NEW} = \zeta * (LIR) + IR_{CUR} * \left(\frac{100 - IRC}{100}\right) \tag{11}$$

where,
IR_{NEW} = Inflation Rate for next year / cycle
IR_{CUR} = Inflation rate for this year / cycle
IRC = Inflation Reduction Coefficient
LIR = Long term inflation rate
ζ = Long Term inflation Coefficient (0 or 1).

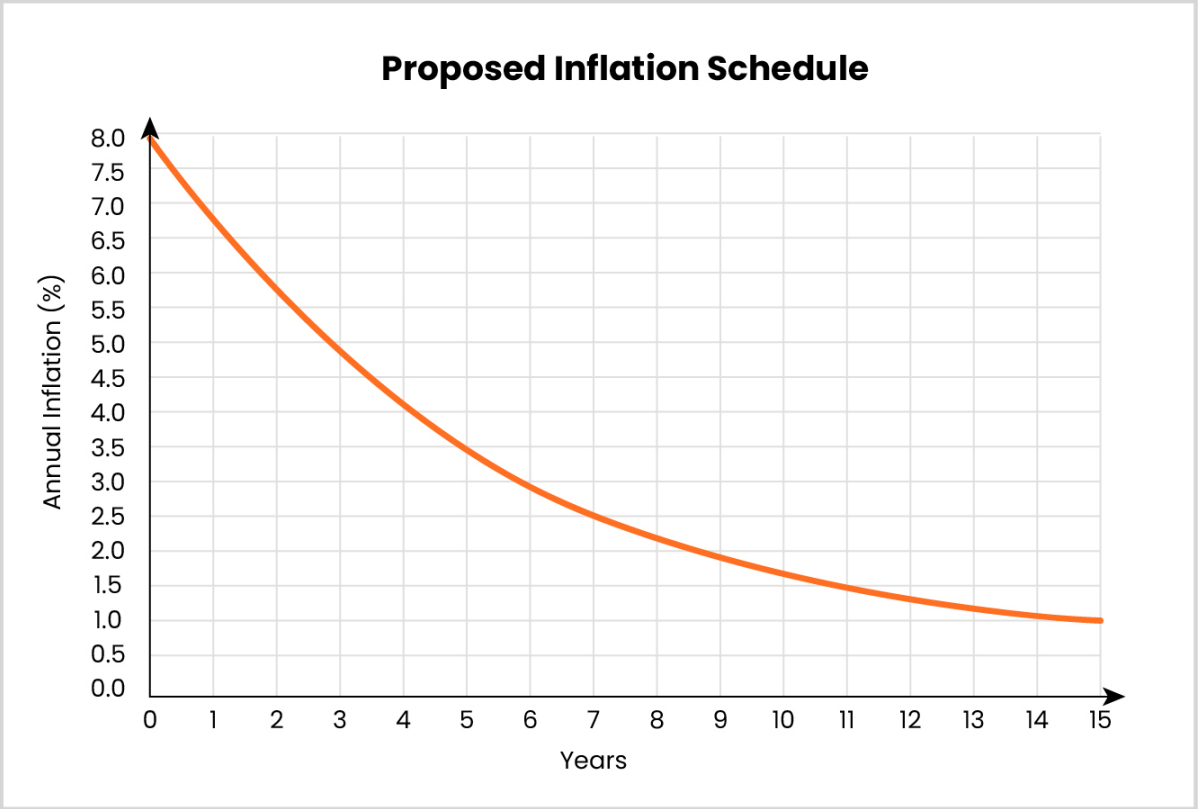

Figure 40: The hypothetical graph above shows how inflation rate will decrease over time.

15.3.2 CBE Coin Burning

To ensure inflation of CBE always remains within the desired range, in addition to its inflation schedule, Cobe will utilize three distinct coin burning methods:

Circulating Supply CBE Burning: This involves burning a percentage of the CBE supply in circulation. Implementing this mechanism will require a governance vote by Cobe's community.

Total Supply CBE Burning: This involves burning a percentage of the total number of CBE minted to date. Implementing this mechanism will again require a governance vote by Cobe's community.

Transaction Fee Burning: This involves burning a proportion of transaction fees while allocating the remainder toward staking rewards. This mechanism will only be instigated if inflation is deemed to be too high and Cobe's community votes for a reduction in the circulating supply of CBE.

Coin burning, particularly total supply and circulating supply coin burning, are intended to be used only as a last resort measure to curb inflation. They can only be implemented via a community-wide governance vote where a large majority of voters believe it would be of benefit to the ecosystem.

15.3.3 Staking Rewards (SR)

CBE coin holders who stake their coins will be issued with staking rewards. In addition to the base minimum staking reward, validators can further increase their staking rewards based on the following parameters:

Lock in Duration (LiD): The longer a validator locks in their CBE for, the greater the staking rewards they will receive above the base minimum.

$$LiD = \frac{LiD_o}{LiD_t} \tag{12}$$

where,
LiD_o = No. of epochs a user locked his coins
LiD_t = Total no. of epochs under consideration.

Reputation Score: the higher the validator's reputation score, the greater the staking reward they will receive above the base minimum.

The Staking Reward issued to a validator for both Cobe's permissionless CPoS and permissioned CPoA chain can be calculated via the equation below:

$$SR = SR_{base} + \frac{\rho_{th}}{1 + (e^{SR_{base} - \rho)})} \tag{13}$$

where
ρ_{th} = User's reputation threshold, $(SR_{base} + \rho_{th}) \leq SR_{Max}$
ρ = User's reputation score $(0 \leq \rho \leq 10)$

A node's reputation score (ρ) will be calculated using the equation below and then inputted into the Staking Reward equation above:

$$\rho = \omega_1 * O + \omega_2 * \frac{CBE_{Locked}}{TCS_{CBE}} + \omega_3 * LiD - (\omega_4 * \beta_{missed} + \omega_5 * \beta_{bad}) \tag{14}$$

where,
ω_1, ω_2, ... ω_5 are weights
$\omega_1 + \omega_2 + \omega_3 + \omega_4 + \omega_5 = 1$
O = Validator's online age, $0 \leq O \leq 1$
β_{missed} = Blocks missed by the validator, can be 0 or more
Its value is normalized between 0 - 1
β_{bad} = Bad blocks created by the validator, can be 0 or more
Its value is normalized between 0 - 1.

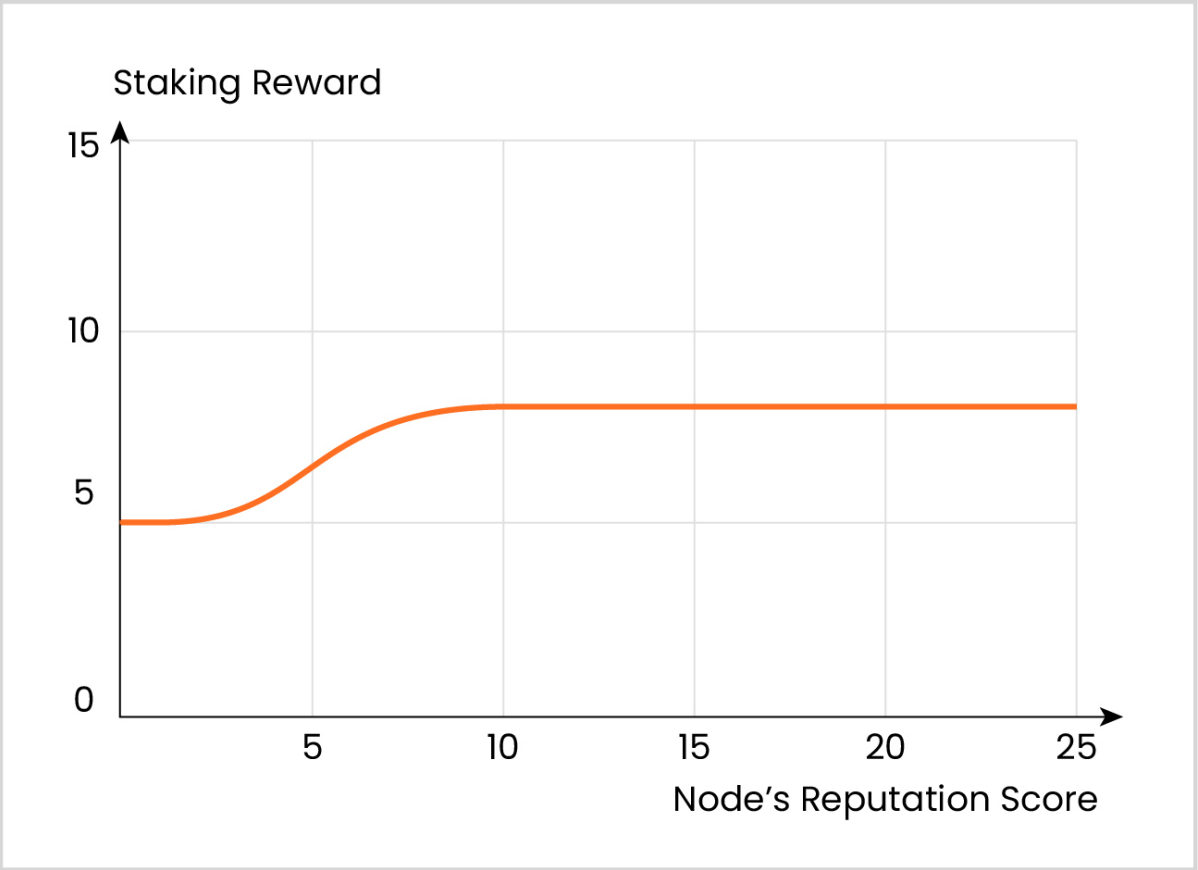

Figure 41: How a node's reputation score affects its staking reward.

16 Governance

Cobe's blockchains will have an independent governance protocol that will give a subset of users the ability to vote on the protocol's changes. Holders of Cobe's native coin (CBE) will be able to vote on all new changes that are put forward. In order to be able to cast their vote, users will temporarily lock their CBE coins in a governance smart contract. They will then be able to vote to support or reject a proposal, and their vote will be weighted proportionally to the number of CBE staked in the governance contract.

Cobe's governance protocol will be divided into four stages:

- Stage 1 (Proposition Forwarding): topics for voting on will be put forward for consideration at this stage - an opportunity available to all users and not just Cobe team members or the developer community.
- Stage 2 (Proposition Selection): the decision as to whether a vote will take place on a matter will be determined by Cobe's governance panel. The panel will consist of a minimum of 21 members made up of members from different areas of the community, including the core team, developers, service users, and Cobe's native coin (CBE) holders. All new members will be added to the panel via community vote. On top of this, regular surveys, polls, and other feedback channels will be put in place to help the panel determine which topics will be put forward for voting.
- Stage 3 (Voting): this will take place after a topic has officially been approved for voting. To maximize transparency, all voting results will be announced in a timely manner on

Cobe's website. To maximize participation, the opening time, main topic, and closing time of the voting will be communicated by email to potential voters.

- Stage 4 (Implementation): all changes that require implementation will be executed as soon as possible, with a level of urgency being assigned to each change. This will depend on critical factors that will affect the security of the platform, the resources required to implement the change, their complexity, other pending tasks, and the expectations of the community. The time frame in which each change will be implemented will be announced on Cobe's website.

Cobe's governance topics will include, but not be limited to:

- System and protocol updates.
- Approving new cryptocurrencies that can be used as collateral.
- Adjusting the risk parameters associated with a currency such as its LTV.
- Market data feed selection.
- Governance processes.
- Emergency actions.

The Cobe blockchain will have two different governance layers, one for the CPoS chain and a second one for the CPoA chain, described in the next sections.

16.1 CPoS Blockchain Governance Structure

16.1.1 Types of Stakeholders, Eligibility, and Voting Rights

Any user of the Cobe network can in principle submit a proposal to be voted on by other peers in the network. Before being implemented as votable, the proposal needs to be submitted as an "improvement proposal" on the Cobe public forum, where it can be further elaborated taking in consideration suggestions and comments from peers.

A user (or group of users) interested in formally submitting a proposal needs to deposit a minimal quantity of CBE coins into a smart contract. The deposit activates the smart contract and initiates the process that leads to voting and – if the outcome of the vote is positive – to the implementation of the changes. Any user of the Cobe network owning CBE coins can, in principle, vote on the blockchain. The actual possibility of voting at a given moment is, however, restricted to individuals having a sufficiently good reputation (R) within the network, and the weight of the actual vote is established following the rule below. The voting reputation – as clarified below – is linked to the active participation of the member in the governance of the network.

In order to vote, each network participant needs to deposit a sum of coins in the aforementioned smart contract. This money is kept in the smart contract until the end of the voting period. The higher the deposited number of coins, the higher the weight of their vote (subject to having a sufficiently high reputation for their vote to matter).

Calling S_i the sum deposited for voting by agent i, their vote counts for:

$$VW = \begin{cases} 0 \ (or \ no \ vote \ allowed) \ if R_i < R_{min} \\ S_i \frac{R_i - R_{min}}{R_{max} - R_{min}}, \ if \ R_{min} \leq R_i \leq R_{max} = 100 \end{cases} \tag{15}$$

where VW = Vote weight.

The parameter R_{min} (minimum value of Reputation) is fixed to 50 initially but can be modified through a CPoS vote, via the procedure described herein.

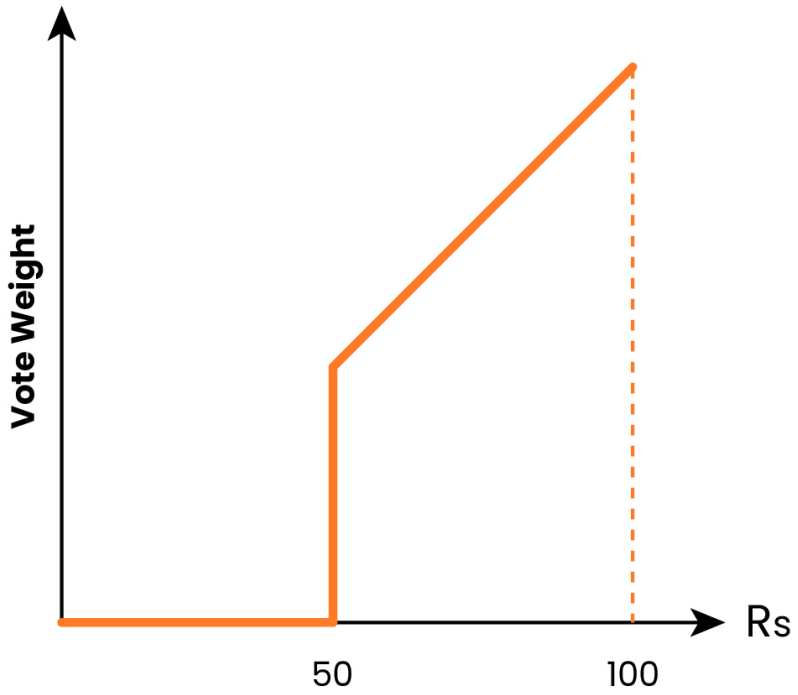

Figure 42: Graphical representation of Eq.15.

- Voting time: voting needs to be allowed for a minimum of three weeks and a maximum of three months. The exact time is decided by the proponent of the proposal.
- In order to be valid, the vote needs to reach a quorum of at least 66% of eligible members or users voting. The proposal is accepted if the weight of the supportive votes exceeds 50% of the total votes.
- Once the voting process is closed, total votes are revealed and computed, and the following actions are enabled:
  - The election has not reached the quorum: in this case, no action is taken, the smart contract is rescinded, and all coins are returned to the original owners, except those of the original proponent, which are kept as penalty.
  - The election reaches the quorum, but fewer than 50% of weighted votes were 'yes': no action is taken, the smart contract sends back the coins to all voters (subtracting a small fee), including the original proponent.
  - The vote reaches the quorum, and more than 50% of weighted votes were yes: the proposal is approved; the change is immediately made effective if algorithmically possible, or authorization is given to the relevant stakeholders to enact the approved change.
  - Implementation of changes is confirmed through a vote of executive nodes. Executive nodes are chosen among validators with the highest reputation, monitored and calculated as described in Section 8.3.

- If the vote reaches the quorum, regardless of the result of the voting, all accounts that participated in the voting enhance their reputation by a fixed quantity Rv. The condition according to which the reputation is increased only if the quorum is reached prevents malicious users from being able to exploit random vote submissions to increase their reputation within the Cobe ecosystem. All voting reputations are initially set to 50. Once a new profile is created and starts to vote, then it increases by discrete steps of Rv (initially set to 10, but modifiable via on-chain governance voting). Conversely, inactivity (each time a person does not vote) is penalized by an amount Rp set initially set to 5. If the reputation of a user falls below 50, then the user is blocked from voting for an extended amount of time (initially 5 votes). After that, the reputation is restored to its minimum value Rmin.

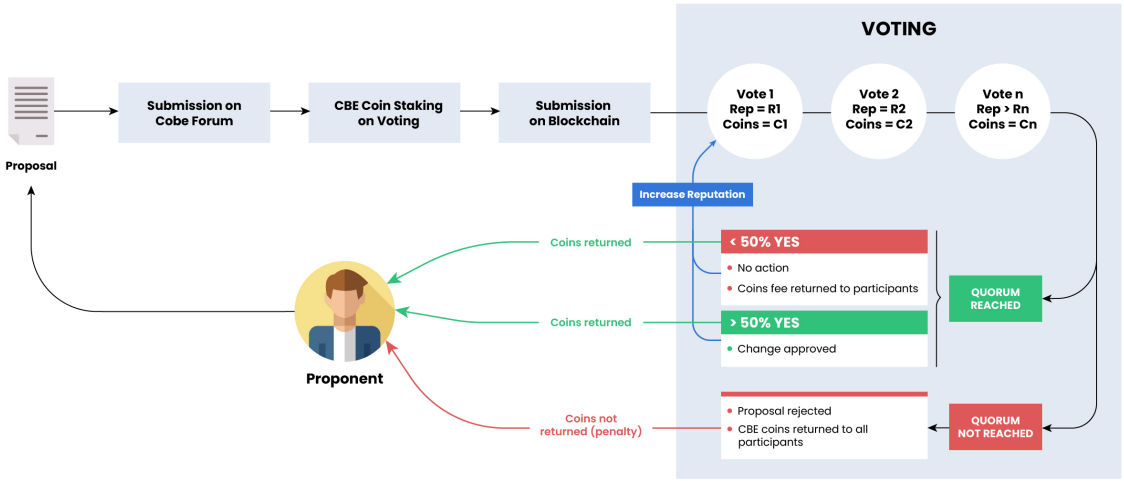

Figure 43: Summarizes the voting process.

16.2 CPoA Blockchain Governance Structure

All nodes will undergo a mandatory process of authentication before they are allowed to become part of the network. In this phase, a node must submit documentary evidence to prove its true identification. After passing through a rigorous verification process, a node can become a validator. Unlike with Concurrent Delegated Proof of Stake, where nodes are elected to become validators, the nodes in the Concurrent Proof of Authority chain are selected based on their compliance with the identity stack. This is a one-time process, and nodes will be assigned a unique identity after the completion of the process. This unique identity will be used as an identifier in all subsequent communications between the nodes. A node that wants to become validator in a CPoA blockchain must hold a certain minimum number of Cobe native coins (CBE) in its account. At the beginning a node must have at least 350,000 CBE. Any node that does not meet these criteria will not be eligible. It is fundamental that nodes becoming validators are autonomous and not affiliated with one another. The CPoA permissioned blockchain will allow only a limited number of validators initially. It will run with 21 validators at the beginning, and the number may increase in the future. As in the case of the CPoS chain, before being implemented as votable, a proposal needs to be submitted as an "improvement proposal" on the Cobe CPoA blockchain public forum.

17 Nucleus Platform

The Cobe native cross-border trade platform, called 'Nucleus', will consist of a series of interlinked applications built on top of its native permissioned and permissionless blockchains. The platform will help solve the three fundamental issues that pose major obstacles to cross-border trade: trust, finance, and product authentication. The goal of the platform is to make cross-border trade cheaper, faster, easier, and more transparent.

17.1 Nucleus's Cross-Border Trade Platform

Most businesses have serious concerns when trading across borders. For buyers, it's trusting that they'll receive their goods upon payment, while for sellers, it's hoping that they'll receive payment for the order placed.

The primary financial instrument businesses use for international trade is the Letter of Credit. A Letter of Credit (LC), issued by the buyer's bank, is a guarantee that payment will be made to the seller once the necessary documents have been produced, including proof of shipment (bill of lading), certificate of origin (CO), and certificate of inspection (CI).

While an LC provides both buyer and seller with increased assurance and security, these pose several challenges for small businesses.

First, they are costly, and the fee will usually exceed 2,000–3,000 USD [19] regardless of the transaction amount. They are also not easily accessible and can take weeks or months to set up. In addition, LCs are complex, resulting in fraud due to false documentation being prevalent. For all these reasons, LCs are an impractical option for small and medium-sized businesses.

17.1.1 Nucleus's Cross-border Trade Platform – Key Features

The Nucleus cross-border trade platform works by allowing buyers and sellers to easily create sales contracts. These are then stored on Cobe's secure blockchain via smart contracts signed by both parties. By using smart contracts, Nucleus will simplify the payment protection process for buyers and sellers without needing a centralized third party.

As part of the sales contract, both parties will agree upon what percentage of the transaction to pay the seller upfront, upon shipment, and upon receipt of the goods. Once the buyer and seller agree on the terms of their sales contract, they will then secure the trade via Nucleus's smart escrow feature. Using this facility, Nucleus will enable the buyer to transfer funds into its smart escrow vault where they will be held securely.

Holding the funds in a secure smart escrow contract will ensure that both parties meet their obligations as defined in their sales contract, as funds will not be released to the seller if they don't act in accordance with it. Similarly, the buyer will be obliged to pay the seller from the funds that are held in escrow once the seller meets their obligations.

Before commencing trading, users must submit their Know your Customer (KYC) information on the platform. Nucleus's KYC automation processes will reduce fraud, increase accountability, and improve performance.

To provide the greatest level of flexibility, Nucleus will allow for transactions to be conducted in either fiat or cryptocurrency. To avoid volatility problems, Nucleus will use its own stable coin, called Cobe Stable Coin (CBS), pegged at a 1:1 ratio to the USD.

Also, each user's trading history will be tracked. The better a user's track record, the greater the confidence others will have in their ability to meet obligations. Table 9 compares a traditional LC to Nucleus's solution.

Table 9: Nucleus's cross-border transaction vs. Letter of Credit.

	Letter of Credit (LC)	**Nucleus**
Transaction Cost	>$2,000	<$100
Execution Speed	> 4 Weeks	< 30 Minutes
Transaction Currency	Fiat	Cryptocurrency or Fiat
KYC	Manual	Automated
Fraud Risk	High	Low

17.1.2 Dispute Resolution

Disputes between buyers and sellers are not uncommon, which is why Nucleus will resolve them as effectively as possible using minimal centralization. Nucleus won't make business decisions on behalf of users, which is why it will take a decentralized approach to disputes.

This is made feasible via a well-designed system where numerous resolution mechanisms are put in place, including transparent sales contracts, independent expert reviews, and minimal benefits for bad actors.

Independent third-party experts, like legal professionals, notaries, and consultants, will be able to sign up to the platform, giving both buyers and sellers access to document validation for certificates and KYC for an agreed fee. These experts will need to provide a report for each trade, improving their rating and credibility on the platform. In addition to resolving disputes, this feature will reduce KYC fraud, as expertly validated KYC will be more credible than non-validated documentation.

17.1.3 Nucleus's Cross-border Trade Platform Technology Stack

Table 10: Nucleus's cross-border trade platform technology stack.

Layer	**Features**
Presentation	User Dashboards, Web & Mobile Interfaces, etc.
Application/Service	Frontend and Backend Apps, Smart Contracts, DApps (Languages: React, Node, Solidity, Go)
Database	NoSQL (MongoDB), SQL
Coins/Tokens	Cobe's Native Coin (CBE) Cobe Stable Coin (CBS) Approved Coins & Tokens for Collateral Deposits (CBR-1)
Blockchain	Cobe's Native Blockchain, Consensus (CPoS, CPoA)
Infrastructure	Network, Compute and Storage (Cloud Services: Amazon Web Services, Google Cloud, etc.)

17.1.4 Cross-Border Transaction Architecture

Nucleus's cross-border payment mechanism will be fast, efficient, secure, and low-cost. Users will have the ability to transact in either fiat or the cryptocurrency of their choice.

Fiat Cross-Border Transaction Architecture

Once the buyer and seller agree on the terms of their sales contract (terms signed), the fiat transaction mechanism works as follows:

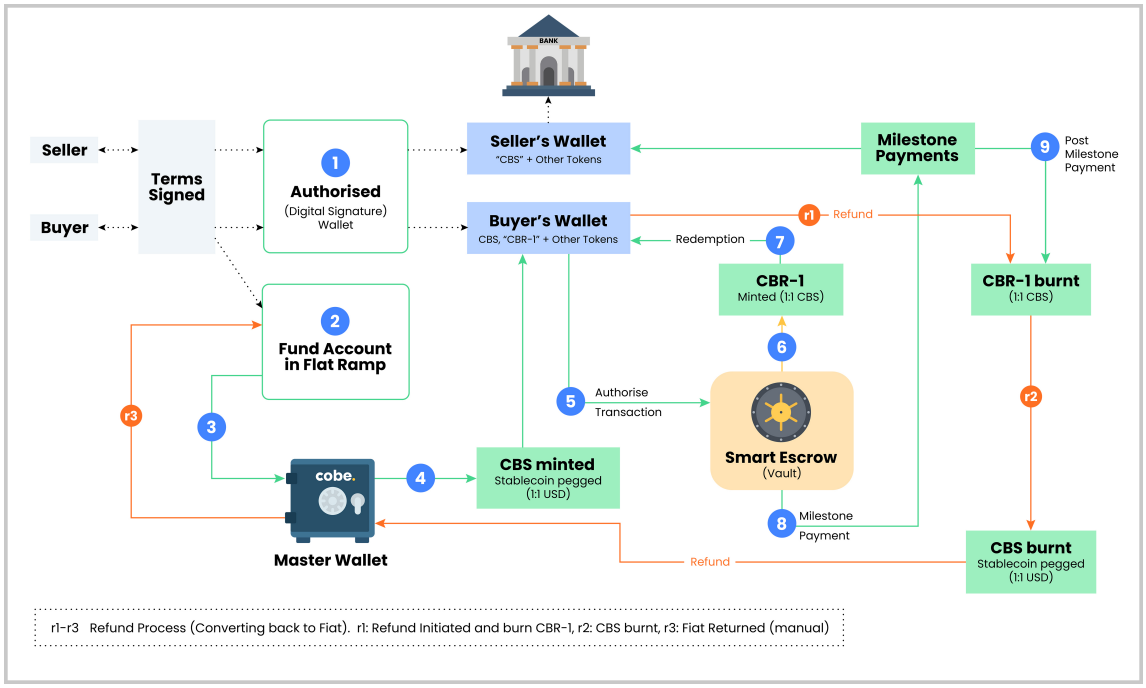

Figure 44: Fiat cross-border transaction architecture.

- **Step 1:** The buyer and seller are verified and authorized to proceed with the transaction.
- **Step 2:** Funds are transferred by the buyer from their fiat bank account to Nucleus using a fiat gateway.
- **Step 3:** Funds are transferred from the fiat gateway to Cobe's Fiat Master Wallet. Cobe's Master Wallet is an audited fiat account held with a licensed bank, which safely stores funds transferred from Nucleus's users to conduct transactions.
- **Step 4:** As soon as the fiat funds reach Cobe's Master Wallet, an equivalent amount of Cobe stable coin (CBS) will be minted, pegged at a 1:1 ratio to USD.
- CBS minted is transferred to the buyer's wallet.
- **Step 5:** Once the buyer authorizes, the stable coins (CBS) are transferred to Nucleus's Smart Escrow Vault on behalf of the buyer to pay the seller as per the terms of the sales contract.
- **Step 6:** As CBR-1 (Cobe's Redemption Token) is pegged at a 1:1 ratio with CBS, the equivalent amount of CBR-1 is minted to the amount of CBS that the buyer has transferred to Nucleus's Smart Escrow Vault.
- **Step 7:** This newly minted CBR-1 is transferred to the buyer's wallet.
- **Step 8:** The buyer releases tranche (milestone) payments to the seller as the seller fulfills their obligations.

- **Step 9:** When funds are released from the smart escrow to the seller, the equivalent amount of redemption tokens (CBR-1) in the buyer's account are burned.

Alternatively, if the funds are refunded to the buyer, the stable coins are transferred back to the buyer and the equivalent amount of redemption tokens are burned.

Once the transaction has been completed, the seller (or the buyer in the case of a refund) has the option to convert their stable coins back to fiat and have them transferred to their fiat bank account.

Once a user converts their stable coins to fiat, the stable coins associated with the transaction are burned. This mechanism of minting and burning CBS in line with the amount of USD held on the Nucleus platform will ensure that it always remains adequately collateralized at a 1:1 ratio with USD, making it resistant to price fluctuations.

Refund Process.

In the case of a refund being issued to the buyer, the process shown in orange in Figure 44 is initiated. This includes:

R1: the refund process is initiated. Once the CBS is refunded to the buyer, the equivalent amount of CBR-1 in the buyer's wallet is burned.
R2: Once the buyer requests their refunded CBS to be converted back to fiat, the concerned CBS is burned.
R3: The equivalent amount of fiat (USD) to the amount of CBS burned is transferred from the Cobe's Master Wallet to the buyer's bank account via the fiat ramp.

Cryptocurrency Cross-Border Transaction Architecture

Performing a cross-border trade transaction using a cryptocurrency has a similar process to that detailed above but is simpler as it does not include the minting of CBS and does not involve CBR-1. The below figure describes this process:

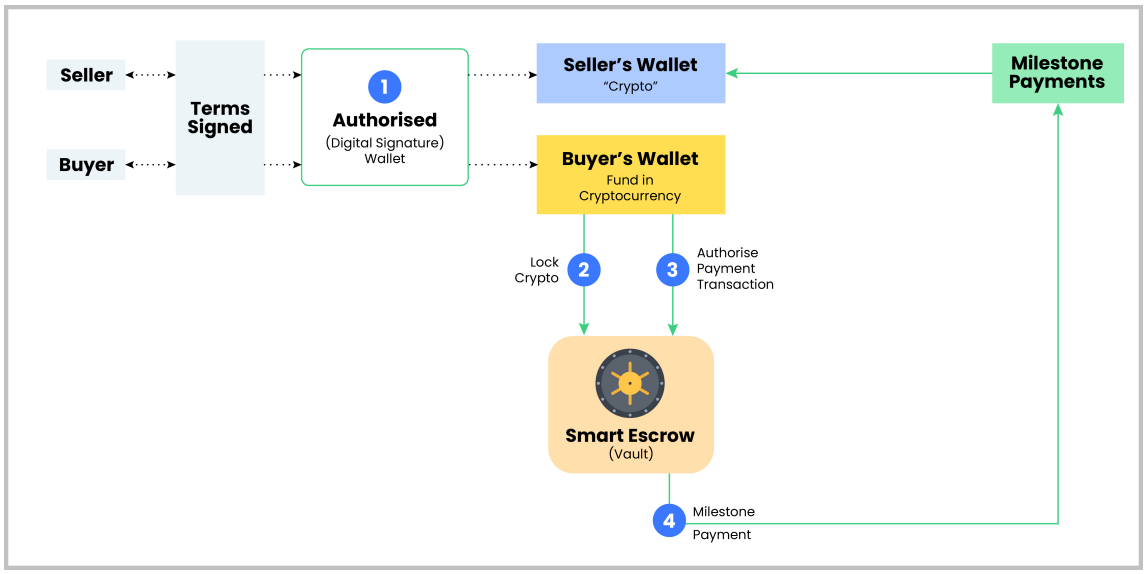

Figure 45: Crypto cross-border transaction architecture.

17.2 Nucleus's DeFi Platform

SMEs continue to struggle to obtain trade finance from traditional financial institutions, with over 50% of SME commercial financing applications being rejected [20]. The current financing model relies heavily on centralized financial institutions and has resulted in the exclusion of a significant percentage of small businesses from the global economy. When it comes to trade finance, there is currently no effective decentralized solution available.

The Nucleus platform will address this problem by creating the most comprehensive trade DeFi solution to date

17.2.1 Cryptocurrency Backed Trade Finance (DeFi)

Loan applications usually take weeks to process with traditional trade finance. With Nucleus, both buyers and sellers will be provided with instant loans against cryptocurrency they place as collateral on the platform.

Unlike traditional finance or regular cash, Nucleus's DeFi solution offers the following benefits:

- **Faster Execution:** when being issued with trade finance from traditional centralized lenders, SMEs are expected to submit assets, such as property, as collateral – which takes time and money to value. It's much faster and easier to implement cryptocurrency as a source of collateral.
- **Lower Costs:** the amount of work required from the lender to issue a traditional trade finance loan makes it costly. Nucleus's automated algorithmic protocols bypass this and bring down the price of application processing – resulting in cheaper borrowing rates.
- **Retain Portfolio Gains:** cryptocurrency is usually held onto as a form of investment, so buyers and sellers who cash out run the risk of missing potential gains.
- **Income Tax Deferment:** income tax can be deferred by placing cryptocurrency as collateral, maintaining cash flow.
- **Track Record:** by taking out Defi loans and meeting their obligations, borrowers can build a strong track record on Nucleus's platform - resulting in favourable lending terms.
- **Cobe's Native Coin (CBE) as Collateral:** Nucleus offers lower net borrowing costs to those who place its native coin (CBE) as collateral, by charging a lower interest rate and loan processing fee. This will, in turn, benefit investors in CBE by increasing demand for the coin.

17.2.2 Centralized Trade Finance (CeFi)

In addition to decentralized lending (DeFi), Nucleus will also facilitate lending for buyers and sellers from traditional trade financiers as well. Acting as a marketplace, Nucleus will facilitate lending from traditional trade finance houses by allowing both borrowers and lenders to source relevant partners. This will involve leveraging trustable track records to validate the information required to approve loans and negotiate terms – which will then be finalized using smart contracts.

This form of lending can be highly beneficial in the following situations:

- **Lending Without Collateral:** applicants unable to provide collateral will be able to obtain trade finance against other factors; such as proof of order, track record on the platform or through allowing the lender to place a charge on the funds in their smart escrow.

- **Lending Against Alternative Assets such as Property and the Seller's Inventory:** Cobe's immutable blockchain will help prevent fraud by making it easy for lenders to verify the information provided by a borrower and to limit multiple loans being taken out against the same asset.

Nucleus will charge a fee for each successful lending transaction, with users being rewarded with lower fees if Cobe's native coin (CBE) is used to make payment. This will help increase both the adoption and value of CBE.

Figure 46 shows the overall architecture and steps of CeFi trade finance.

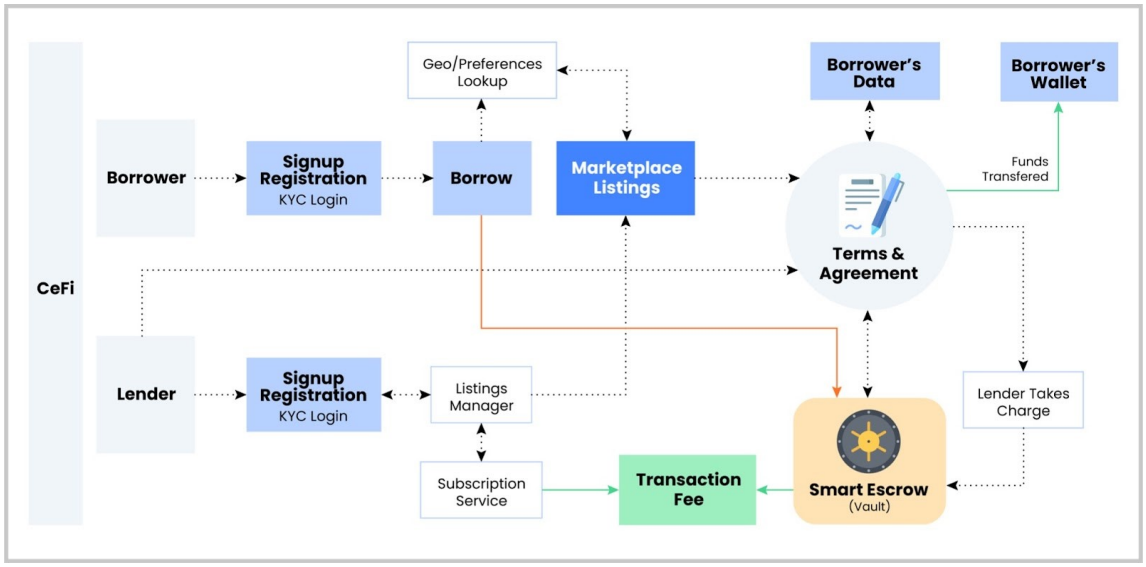

Figure 46: Nucleus's CeFi trade finance.

17.2.3 Nucleus DeFi Platform Technology Stack

Table 11: Nucleus DeFi platform technology stack.

Layer	**Features**
Presentation	User Dashboards, Web & Mobile Interfaces, etc.
Aggregation	Data Consolidation (Market & Lending Data, etc.)
Application/Service	Frontend and Backend Apps, Smart Contracts, DApps (Languages: React, Node, Solidity, Go)
Database	NoSQL (MongoDB), SQL
DeFi Protocol	Loans - Exchanges - Asset Management - Smart Escrow - Governance
Coins/Tokens	Cobe's Native Coin (CBE) Cobe Stable Coin (CBS) Approved Coins & Tokens for Collateral Deposits
Blockchain	Cobe's Native Blockchain, Consensus (CPoS, CPoA)
Infrastructure	Network, Compute and Storage (Cloud Services: Amazon Web Services, Google Cloud, etc.)

Nucleus will introduce a decentralized finance protocol (DeFi) where users can borrow Cobe's stable coin (CBS) against cryptocurrency – placed as collateral in a dedicated vault.

17.2.4 DeFi Borrowing

Figures 47 and 48 below show the DeFi borrowing process and seller milestone-based borrowing, respectively

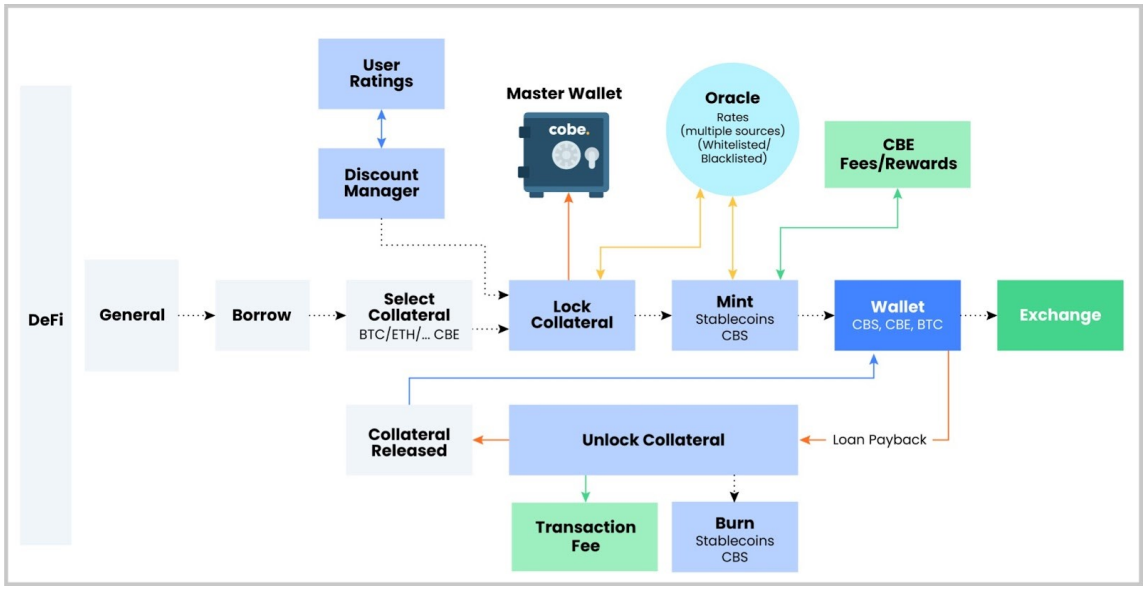

Figure 47: DeFi borrowing process.

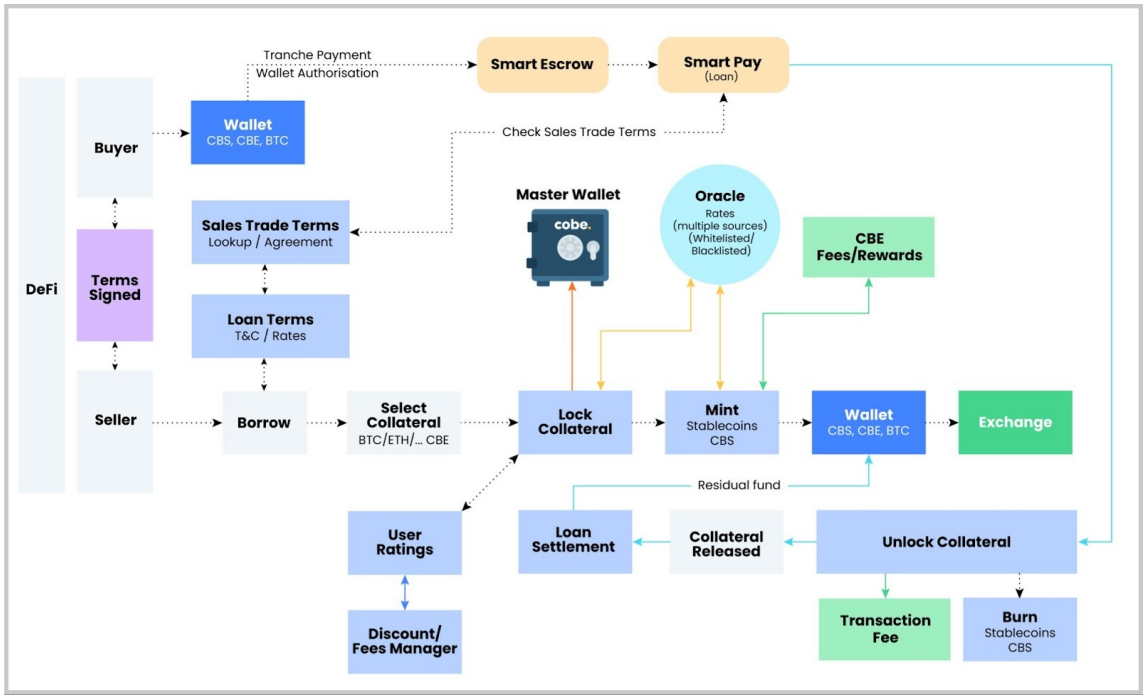

Figure 48: Seller milestone-based borrowing.

17.2.5 Minting New Cobe Stable Coins

Whenever cryptocurrency is placed as collateral in a dedicated vault, new Cobe stable coins (CBS) that are backed by the borrower's collateral and pegged at 1:1 ratio to USD will be minted.

The amount of stable coin generated will be proportional to the Loan to Value (LTV) ratio. For example, if a currency has an LTV ratio of 50%, then the number of stable coins minted will be equivalent to 50% of the value of the asset (USD) being placed as collateral.

This can be expressed as:

$$\begin{aligned} New\ Cobe\ Stable\ Coin\ (CBS)\ minted = {} & LTV\ \ Ratio \\ & * Market\ Price\ of\ Collateral\ Currency\ (In\ USD) \\ & *\ \ Amount\ of\ Collateral\ Currency \end{aligned} \tag{16}$$

As soon as a user returns their borrowed stable coins and their collateral has been transferred back to them, the stable coins associated with the transaction will be burned. This will ensure that Cobe's stable coins always remain adequately collateralized against the USD, retaining their stability.

17.2.6 Interest Rates

Nucleus's DeFi platform will provide users with highly competitive interest rates ranging between 0.5 and 3% per annum, making it approximately 10 times more cost effective than traditional trade finance. Through cost-effective lending, Nucleus will facilitate trade and adoption of its ecosystem.

To further incentivize holistic growth of the platform, users will be offered progressively lower interest rates based on their personal user rating score and by placing Cobe's native coin (CBE) as collateral.

Each user on the Nucleus platform will be given a user rating between 1 and 100, which will grow as they increase their activity and meet their obligations. The higher a user's rating, the greater the discount they will receive on their interest fee.

Users will receive further discounts on their interest rate fee if they place CBE as collateral, increasing its demand and adoption.

However, to ensure a sufficiently diversified basket of crypto assets backing the stable coin (CBS), the discount will only be available on a maximum of 2.5% of Cobe's native coin (CBE) locked as collateral. The discount will increase as the amount of CBE gets progressively closer to 2.5%.

The final interest rate can be expressed as follows:

$$I = I_{min} + \frac{I_{max} - I_{min}}{e^{\mu_c R + \chi C}} \tag{17}$$

where,

- I_{min} is the minimum Interest achievable by applying discount ($I_{min} = 0.5\%$)
- I_{max} is the maximum Interest ($I_{max} = 3\%$)
- R is the User's Platform Rating (Range limit 0-100)
- μ_c is the User Rating Coefficient (Range 0-1)
- χ is the Cobe Collateral Coefficient (Range 0-1)
- C is the Cobe Native Coin Collateral Percentage (Range Limit: 0-2.5%).

The user rating coefficient μ_c and the Cobe collateral coefficient χ are the two primary parameters that will be used by the protocol to adjust how much a user's rating 'R' and the fraction of Cobe native coin (CBE) placed as collateral will impact the interest rate discount. For instance, if $\mu_c > \chi$, the most important aspect in determining the discount will be the user's platform rating.

Figure 49 shows an example where we have set $\mu_c = \chi = 0.5$ – user's rating and the usage of CBE as collateral are weighted equally – and C = 1%. I_{min} is set to 0.5%. It can be seen that as the user's rating increases, the interest rate decreases and reaches the minimum value.

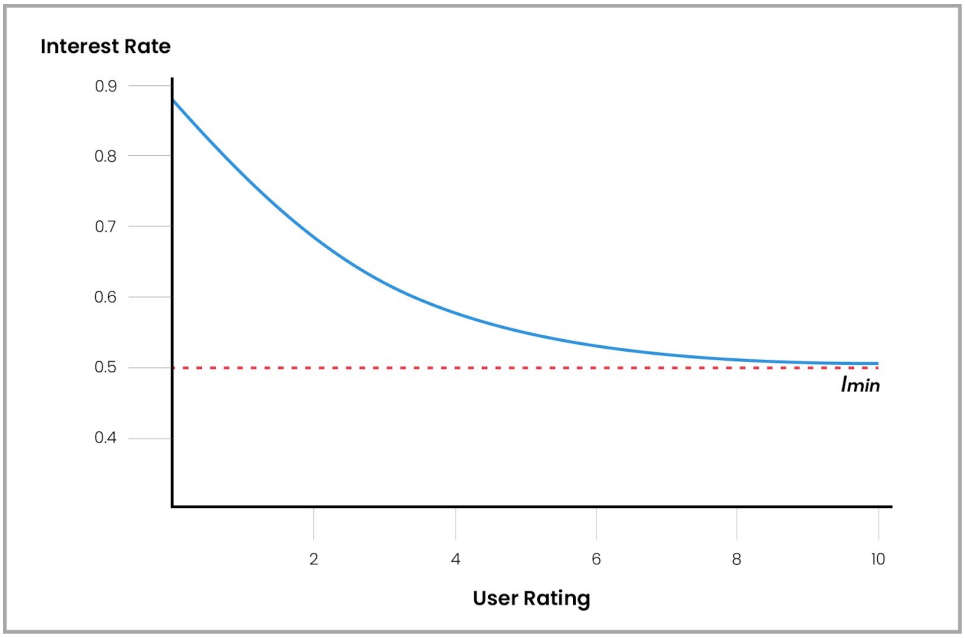

Figure 49: Example of Cobe's interest rate.

17.2.7 Collateral Loan to Value (LTV) Ratio

To ensure that loans are well collateralized, users will need to place a minimum collateral of 150% against the amount that they are looking to borrow. The collateral amount required will also vary from currency to currency, with the required collateral being lower – i.e., closer to 150% – for less risky assets. Liquidations will automatically occur when the collateral value drops below the set LTV value for the associated asset.

If a loan is liquidated, the borrower will incur a liquidation penalty. This measure will incentivize users to remain adequately collateralized and minimize the number of liquidations. By connecting with reliable market data feeds that are selected via community votes (as explained in the next section), Nucleus's DeFi platform will be updated in real time with accurate data on the prices of assets that have been approved as sources of collateral.

The risk associated with each crypto asset used as collateral will be calculated based on its liquidity and volatility. The liquidity of an asset is defined as the average of its volume traded in a given time period, and an asset's volatility is defined as the standard deviation of its price over a set time period. Both the liquidity and volatility of an asset will be measured relative to other collaterals, and the total risk will be weighted as a sum of the two. Therefore, the risk of an asset 'A' will be calculated by the weighted sum weights being indicated by (ω) of its liquidity (λ_A) and volatility (δ_A) relative to the total liquidity and volatility of the other collaterals, using the following formula:

$$\Psi_A = \omega \frac{\delta_A}{\sum_i \delta_i} + (1 - \omega) \frac{\lambda_A}{\sum_i \lambda_i} \tag{18}$$

To maximize the accuracy of the formula, the weighted sum of both an asset's liquidity and volatility will be calculated using daily, weekly, and monthly data. In addition, to account for strong shifts in market conditions, Nucleus's DeFi governance protocol will allow for both the liquidity and volatility weights to be adjusted through transparently designed processes. This will help minimize the number of forced liquidations in downturns and provide users with the best possible LTV ratios during periods of high growth.

17.3 Nucleus's Product Authentication Platform

There is a growing demand for transparency in the authentication of goods, both from customers and governments. At the end of 2020, global counterfeit losses were estimated to be around 1.82 trillion USD – with the losses continuing to grow [21]. A recent survey has shown that 54% of consumers want as much detail as possible on their goods, due to a growing distrust in the claims made by manufacturers [22]. The adoption of blockchain technology to enable secure traceability for the management of a product supply chain, providing information such as the provenance of a product and preventing fraud, is emerging rapidly due to the inherent trust and inalterability provided by this technology. Nucleus will build assurance by integrating goods authentication into a single, cohesive, transparent, and easy-to-use platform.

There are three components to Nucleus's product authentication solution:

1. **Document Authentication:** all data related to the validation of a product, such as certificates of origin and/or conformity assessment certificates, will be notarized on Cobe's unalterable blockchain, which can then be cross-examined with the consent of the seller. Each user's documents will also be linked to their KYC data, de-incentivizing fraud. Moreover, both buyers and sellers can have documents independently reviewed by a neutral third-party to prevent fraud. These parties will be registered to the platform and be experts in their respective fields, including notaries, legal experts, and specialist consultants.

2. **Track & Trace:** Nucleus provides a semi-decentralized end-to-end blockchain-based track & trace ecosystem, covering the whole lifecycle of goods from manufacturing to distribution. The system will be built using highly secure encrypted data matrix code technology. Using this, manufacturers will be able to mark products with unique data matrix codes that can then be scanned at any point in the product's lifecycle to ensure authenticity, prevent counterfeiting, and track logistics.

3. **Provenance:** Nucleus's provenance applications will provide users with access to data on a product's entire supply chain history, using IoT devices to collect and store data on location and custody history, environmental conditions of the journey, and accelerometer information for damage assessment. This will all be stored on Cobe's blockchain. Moreover, Nucleus's provenance applications will be integrated with Nucleus's cross-border trade and decentralized finance platforms. This means that the release of payments from the smart escrow account can be synergized with the provenance data, which can track metrics such as the location of goods, their authenticity, and potential damage. Such data can also be used to estimate the lending requirements and associated risk for a particular transaction.

Although several projects have attempted to utilize blockchain technology for the authentication of goods, their scope has been relatively limited. Nucleus, in comparison, offers a cohesive three-pronged approach that includes document validation, track & trace, and provenance. Going further, no other solution synergizes individual cross-border trades and decentralized lending

with product authentication, enabling Nucleus to offer the most comprehensive solution available, as highlighted in Table 12 below.

Table 12: Nucleus's product authentication platform vs. competitors.

	Competitor Authentication Platforms	**Nucleus**
Holistic Approach Including Document Validation, Track & Trace & Provenance	No	Yes
Transaction Fees	Higher	Lower
Adoption Channels	One	Multiple
Focus	Large Enterprises	Large & Small
APIs	No	Yes
Synergy With Cross-Border Payments	No	Yes
Synergy With DeFi	No	Yes

17.3.1 Nucleus's Product Authentication Architecture

Figure 50 below provides an overview of the Nucleus platforms production authentication architecture. This includes its document authentication, track trace, and provenance services.

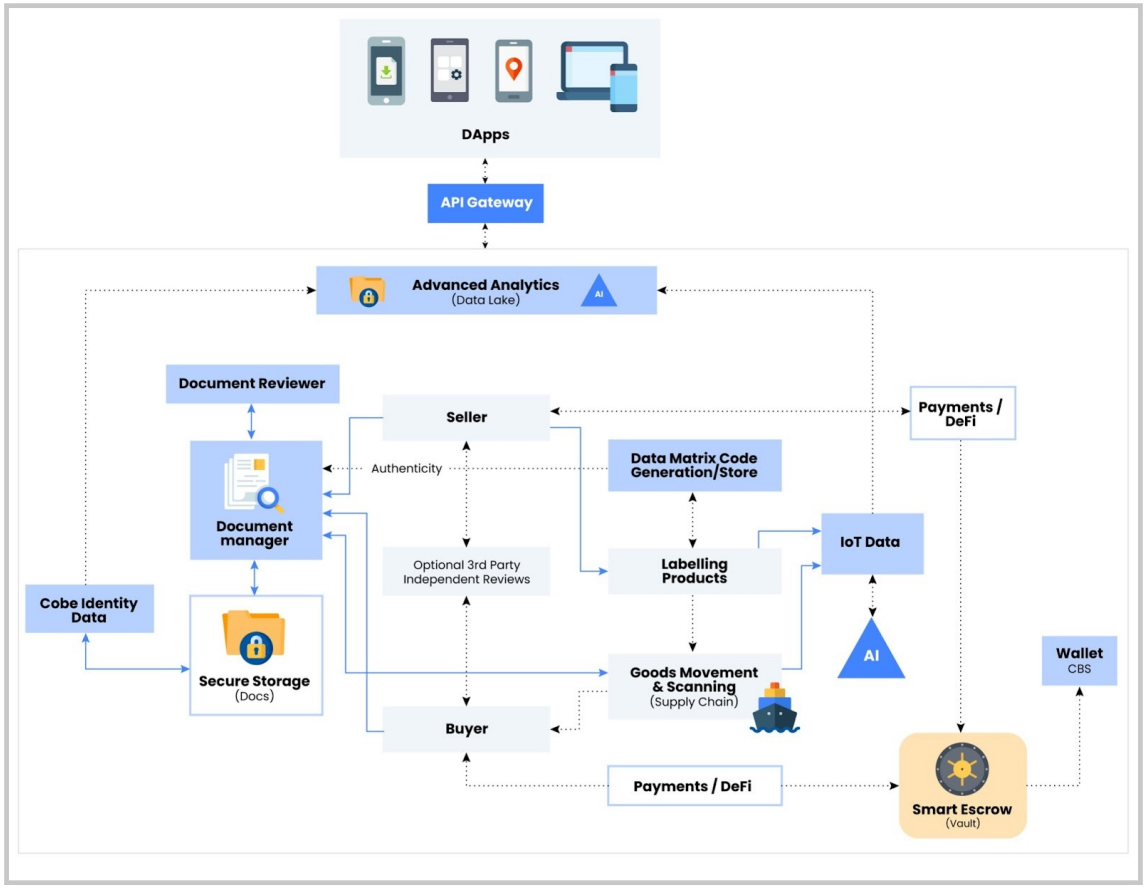

Figure 50: Nucleus's product authentication architecture

17.3.2 Nucleus Product Authentication Platform Technology Stack

Table 13: Nucleus product authentication platform technology stack.

Layer	Features
Tags	RFID tags, NFC, QR Code, Barcode
Scanners	RFID Scanners, Smart Phone With (NFC / QR Code Readers), Barcode Scanners
Application/Service	Frontend & Backend Apps, Application Programming Interfaces (APIs), Smart Contracts, DApps (Languages: React, Node, Solidity, Go)
Database	NoSQL (MongoDB), SQL
Coins/Tokens	Cobe's Native Coin (CBE) Cobe Stable Coin (CBS) Approved Coins & Tokens
Blockchain	Cobe's Native Blockchain, Consensus (CPoS, CPoA)
Infrastructure	Network, Compute and Storage (Cloud Services: Amazon Web Services, Google Cloud, etc.)

17.4 Nucleus APIs

Rather than having to build everything from the ground up, DApp developers should be provided with APIs that facilitate the development of their applications. The Nucleus platform will provide developers looking to build decentralized cross-border trade applications on Cobe's blockchain with a suite a APIs. These include smart escrow, track trace, document authentication, provenance, and analytics APIs. This will provide Cobe with a distinct competitive advantage when incentivizing cross-border trade DApp developers to build on its blockchain compared with more generic blockchains that are unable to provide this facility.

For example, Nucleus's track & trace solution will include an API that can be integrated with third-party modules, allowing for greater utilization and adoption. Desktop and mobile versions, with user-friendly interfaces, will also be developed and provide comprehensive analytics.

Nucleus's provenance applications will also come with a suite of APIs that developers building DApps on Cobe's blockchain will be able to utilize. This is because the provenance requirements of each business differ depending on the nature of goods, production process, and logistics. By providing a robust set of provenance APIs, Nucleus will dramatically reduce the time, skills, and cost required to build decentralized product authentication applications. No other blockchain provides this facility to developers looking to build on its network.

Moreover, users will have access to data on a product's entire supply chain history, based on IoT devices to collect and store data on location and custody history, environmental conditions of the journey, and accelerometer information for damage assessment.

17.5 Nucleus User Rating

For better decision-making, Nucleus will provide a suite of analytics that includes a rating score for each user.

Each user's rating will be based on their performance on the Nucleus platform, with the metrics including the number of successful transactions performed and obligations met. Every user's rating score will be visible on the platform.

To maintain confidentiality, in-depth analytics regarding user activity will not be publicly available – including a user's individual transactions and the feedback they've received. However,

users will have the option of sharing this data if they want to prove their track record to a trading partner.

Each user's rating score on the platform will be calculated detailing the parameters that will be taken into consideration. More parameters are likely to be added to the user rating formula as the platform evolves. Note that all parameters must be scaled between 0 and 1. The parameters under consideration are:

- **Parameter 1:** Number of cross-border transactions successfully completed – CBT.
- **Parameter 2:** Total revenue of transactions completed – TRT.
- **Parameter 3:** Number of different users successfully transacted with – NDT.
- **Parameter 4:** Rating given by counterparties transacted with – CPR.
- **Parameter 5:** Number of DeFi loan obligations successfully met – NDLO.
- **Parameter 6:** Penalties for failing to meet transaction obligations – P.

The User Rating Score (μ_s) can thus be calculated using the equation.

$$\mu_s = \omega_1 * CBT + \omega_2 * TRT + \omega_3 * NDT + \omega_4 * CPR + \omega_5 * NDLO - \omega_6 * P \tag{19}$$

Where $\omega_1, ..., \omega_6$ are weights assigned to each parameter. The minimum value of a weight can be 0 and maximum is 2. The value of μ_s is scaled between 0 and 10. Therefore we have:

$$R = \frac{R_{max}}{1 + (R_{max} * \mu_{max} * e^{-\mu_s})} \tag{20}$$

Where R is the User Platform Rating $(0 \leq R \leq 10)$, R_{max} is its maximum value and μ_{max} is the User Rating Score maximum value.

17.6 Nucleus Platform: User Adoption Growth

Nucleus aims to deliver high adoption rates via the following mechanisms:

- **Focused Launch:** a lean approach to launching new products will be adopted that focuses on delivering one high value service at a time. New features will then be introduced to each product based on real-time user data – maximizing user adoption and retention.
- **Unparalleled Value:** Nucleus's smart cross-border trade platform, which secures cross border transactions for both buyers and sellers, will be launched first. It will cost around 5% of the price of a typical LC and take just minutes to execute as opposed to weeks. This, in combination with the fact that there is no direct competitor, will boost the rate of user adoption and retention, which will continue to grow with each new service launched – including Nucleus's DeFi and product authentication services.
- **Incentives for Early Adopters:** Nucleus's will implement the following measures to generate a high level of early adoption:
  - **Lower Transaction Fee**: early adopters will be rewarded with a substantial discount on transactions, followed by additional benefits that will keep them committed.
  - **Referral Programs**: a referral program will be put in place, with free transactions being awarded for each referral.
- **Expert Team**: The Nucleus team includes members dedicated exclusively to meeting adoption and retention targets, each with a strong background in creating multi-sided platforms from both marketing and product development perspectives.

18 Sonic: Cobe's Native Wallet

Cobe's dedicated wallet, Sonic, will work seamlessly to empower the entire Cobe ecosystem with enhanced security and functionality. Sonic facilitates safe storage, sending, receiving, staking, borrowing, voting, and token swapping on Cobe's blockchain. The Sonic wallet will include the following advanced features:

Seamless Cross-Border and P2P Transactions: alongside P2P transactions, Sonic will make B2B cross-border transactions seamless. Users will be able to transact using Cobe's native tokens, fiat, bitcoin, ERC20 tokens, and all other major cryptocurrencies that meet the standards of Cobe's governance protocols.

Easy Borrowing: Sonic will be fully integrated with Cobe's DeFi and CeFi lending protocols, ensuring both easy and low-cost borrowing.

Staking on Cobe's Blockchain: using their wallets, Sonic users will be able to seamlessly stake their CBE to generate revenue.

Built in DEX: Cobe's built-in DEX will allow users to safely swap Cobe and ERC20 tokens, as well as trading in all major cryptocurrencies and stable coins that have been approved by governance protocols.

Low Transaction Fees: Cobe's business model relies on the holistic growth of its ecosystem to generate value rather than wallet fees, meaning that Sonic users will benefit from some of the lowest transaction fees available.

Web3 Support: Sonic users will be able to explore blockchain applications built on Cobe with ease.

Multi-Signature Wallet Support: Sonic will support multi-signature functionality, requiring multiple parties to authorize transactions, thereby enhancing security for corporate or group transactions.

Threshold Signature Schemes (TSS): This feature distributes trust among multiple parties, making digital signatures more secure and efficient, especially for DAOs within the Cobe ecosystem.

Post-Quantum Cryptographic Algorithms: To ensure long-term security against quantum computing threats, Sonic will be integrated with post-quantum cryptographic algorithms.

18.1 Sonic Wallet – Technology Stack

Table 14: Sonic wallet – technology stack.

Layer	Features
Presentation	Mobile Wallet, Desktop Wallet, Online Wallet, etc.
Application/Service	Frontend and Backend Services APIs (Languages: React, Node, Solidity, Go)
Database	NoSQL (MongoDB), SQL
Coins/Tokens	Cobe Native Coin (CBE) Cobe Stable Coin (CBS) Cobe Redemption Coin (CBR-1) Cobe Native Token (CB-100) Other Approved Coins & Tokens (e.g., BTC, ETH, ADA)
Blockchain	Cobe's Native Blockchain, Consensus (CPoS, CPoA)
Infrastructure	Network, Compute and Storage (Cloud Services: Amazon Web Services, Google Cloud, etc.)

19 Cobe Labs

Blockchain technology is evolving at a rapid speed, with innovative new solutions being continuously proposed. Distributed ledgers and blockchain are undoubtedly some of the most hyped technologies in recent history, demonstrating the current need for sound and informed analysis. However, despite the great advantages blockchain has to offer, many projects have failed due to weak technological foundations. Cobe believes that this is the result of both a lack of experimentation and divergence from the proper scientific method. That is why Cobe Labs has been established, a platform where its research results will be published for peer review and knowledge sharing with the blockchain community. Cobe's works will be submitted to top conferences and journals to gain insights from the academic world and to guarantee maximum adherence to scientific and technical principles. Cobe uses both empirical and theoretical approaches in its research, which is motivated by practical problems and the underlying theory.

Blockchain technology is impacting multiple aspects of our world, which is why Cobe Labs includes some of the best minds from the fields of computer science, software engineering, mathematics, game theory, data science, economics, AI, and the behavioral sciences.

In addition to collaborating with world-renowned universities, Cobe's team includes leading academics and practitioners from different disciplines who are working together to accelerate innovation. Our research projects solve specific problems where distributed ledgers and blockchain technologies are the focus, based on solid research methods and technical knowledge.

20 The Cobe Foundation

Many regions of the world suffer from high levels of corruption and poor financial and legal infrastructure. This, coupled with the unnecessary complications of cross-border trade, results in millions of people being excluded from the global economy through no fault of their own.

Cobe believes that everyone should have access to global trade. By removing reliance on centralized institutions, Cobe will empower those currently excluded with greater access to

Figure 51: Cobe Labs.

funding and reduced trading costs. The lives and living standards of vulnerable communities will significantly improve should cross-border trade be more readily available.

Cobe's foundation will donate 1% of the Nucleus platform's gross margin to humanitarian initiatives, particularly those that seek to level the playing field and make the global economy accessible to people from underprivileged backgrounds.

To enable the highest levels of integrity and participation, Cobe will ensure that CBE coin holders are able to participate in the governance of the foundation actively and fairly. It will limit the possibility of a few individuals having too much control over the foundation by building its governance around a decentralized quadratic protocol. The equation below describes the Cobe Foundation's quadratic voting formula:

$$c(v) = kv^2 \tag{21}$$

where,
$k \geq 0$, a positive number defined or set by the blockchain platform
v is the number of votes
$c(v)$ is voting cost.

On top of this, users of the Nucleus platform will have autonomy in deciding which philanthropic cause will receive the 1% donations associated with their transaction.

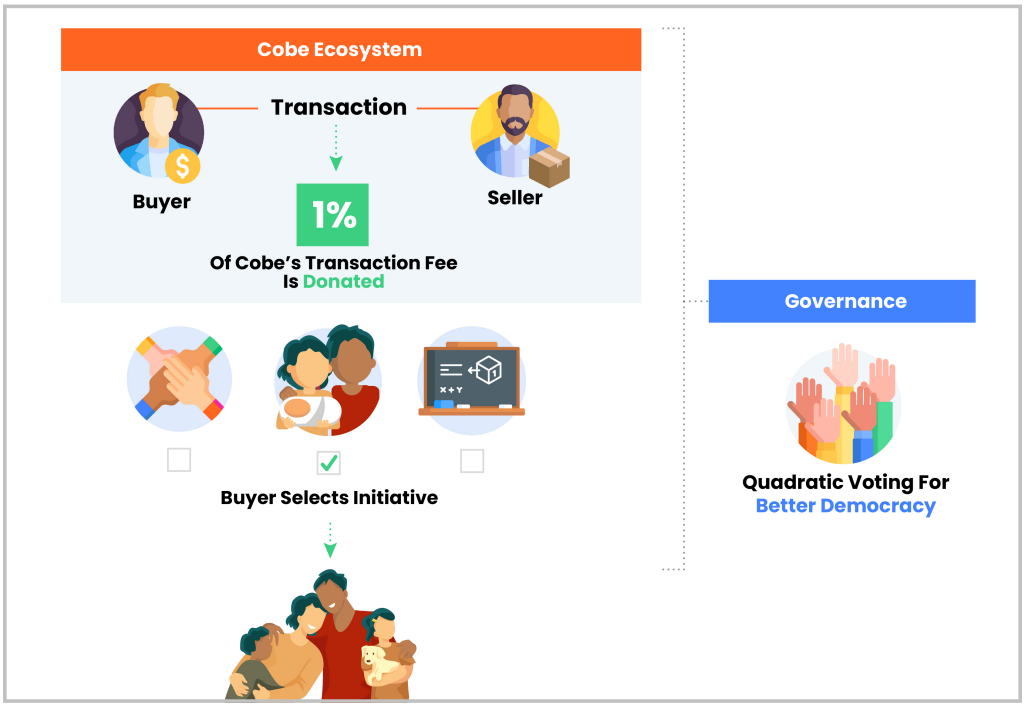

Figure 52: The Cobe Foundation.

21 Cobe Ecosystem

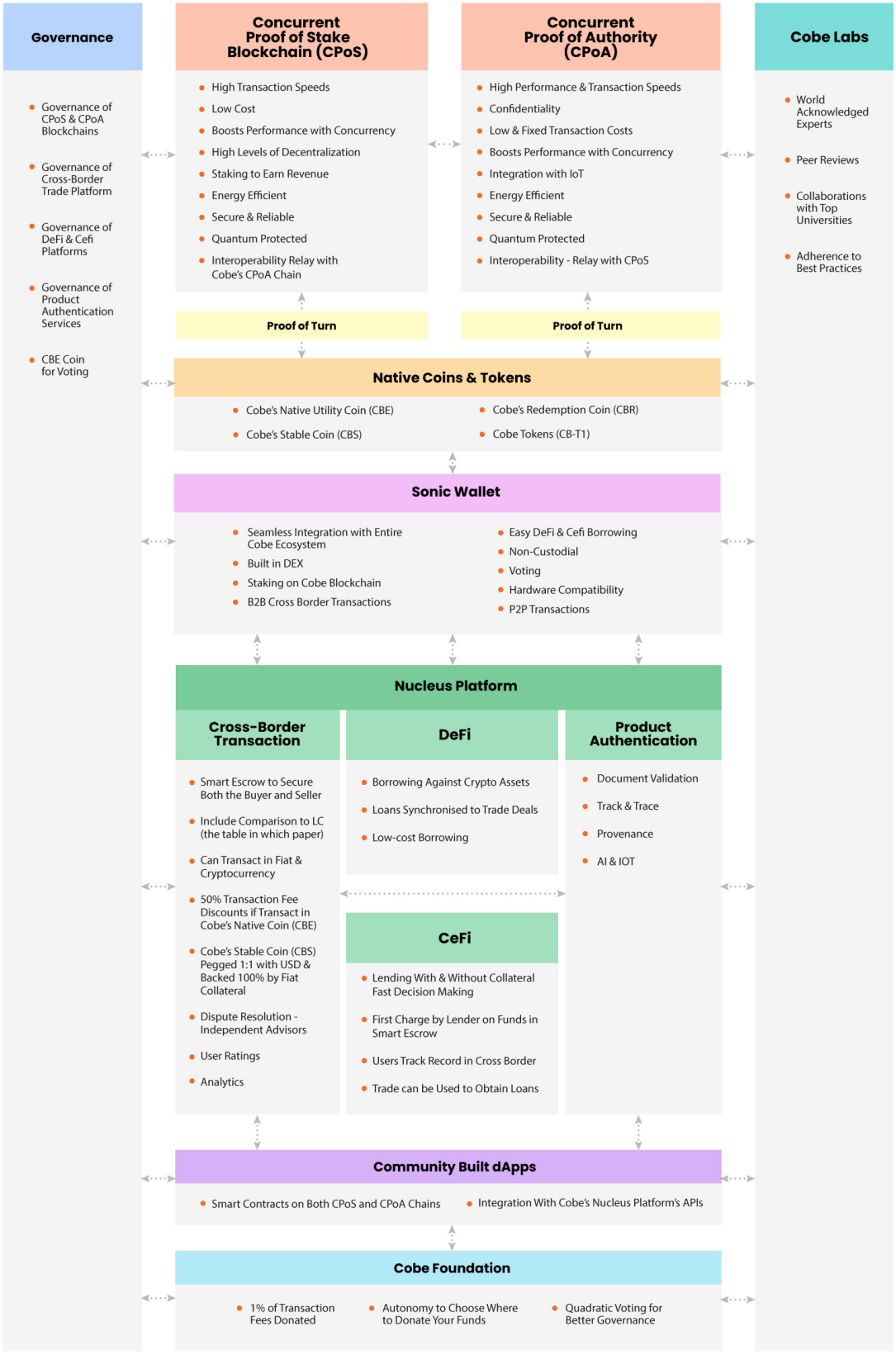

Figure 53: The Cobe ecosystem.

22 Disclaimer

This document is a technical yellow paper that presents the current status and future plans for Cobe. The sole purpose of this document is to provide information and is not to provide a precise description of future plans. Unless explicitly stated otherwise, the products and innovative technologies organized in this document are still under development and are yet to be incorporated. Cobe does not provide a statement of quality assurance or affidavit for the successful development or execution of any of such technologies, innovations, or activities described in this document. Also, within legally permitted scope, Cobe rejects any liability for quality assurance that is implied by technology or any other methods. No one possesses the right to trust any contents of this document or subsequent inference, and the same applies to any of the mutual interactions between Cobe's technological interactions that are outlined in this document. Notwithstanding any mistake, default, or negligence, Cobe does not have legal liability for losses or damages that occur because of errors, negligence, or other acts of an individual or groups in relation to this document. Although information included in this publication has been referred from data sources that were deemed to be trusted and reliable by Cobe, Cobe does not write any statement of quality assurance, confirmation, or affidavit regarding the accuracy, completeness, and appropriateness of such information. You may not rely on such information, grant rights, or provide solutions to yourself, your employee, creditor, mortgagee, other shareholder, or any other person. Views pre- sented herein indicate current evaluation by the writer of this document and are not necessarily representative of the views of Cobe. Views reflected herein may change without notice, and do not necessarily comply with the views of Cobe. Cobe does not have the obligation to amend, modify, and renew this document, and it is not obliged to make notice to its subscribers and recipients if any views, predictions, forecasts, or assumptions in this document change, or any errors arise in the future. Cobe, its officers, employees, contractors, and representatives do not have any responsibility or liability to any person or recipient (whether by reason of negligence, negligent misstatement, or otherwise) arising from any statement, opinion, or information, ex- pressed or implied, arising out of, contained in, derived from, or omitted from this document. Neither Cobe nor its advisors have independently verified any of the information, including the forecasts, prospects, and projections contained in this document. Each recipient is to rely solely on its own knowledge, investigation, judgement, and assessment of the matters that are the subject of this report and any information that is made available in connection with any further investigations and to satisfy him/herself as to the accuracy and completeness of such matters. While every effort has been made to ensure that statements of facts made in this paper are accurate, and that all estimates, projections, forecasts, prospects, and expression of opinions and other subjective judgments contained in this document are based on the projection that they are reasonable at the time of writing, this document must not be construed as a representation that the matters referred to therein will occur. Any plans, projections, or forecasts mentioned in this document may not be achieved due to multiple risk factors, including limitation defects in technology developments, initiatives or enforcement of legal regulations, market volatility, sector volatility, corporate actions, or the unavailability of complete and accurate information. Cobe may provide hyperlinks to websites of entities mentioned in this paper, but the inclusion of a link does not imply that Cobe endorses, recommends, or approves any material on the linked page or accessible from it. Such linked websites are accessed entirely at your own risk. Cobe accepts no responsibility whatsoever for any such material, or for consequences of its use. This document is not directed to, or intended for distribution to or use by, any person or entity who is a citizen or resident of or located in any state, country, or other jurisdiction where such distribution, publication, availability, or use would be contrary to law or regulation.

This document is only available on www.cobe.network and may not be redistributed, repro-

duced, or passed on to any other person or published, in part or in whole, for any purpose, without the prior, written consent of Cobe. The manner of distributing this document may be restricted by law or regulation in certain countries. Persons into whose possession this document may come are required to inform themselves about, and to observe such restrictions. By ac- cessing this document, a recipient hereof agrees to be bound by the foregoing limitations. This yellow paper is an information paper subject to update pending final regulatory review. This paper does not constitute an offer. Any such offer will be subject to final regulatory review and governed by a revised paper and conditions of sale document that will prevail in the event of any inconsistency with the paper set out below. Accordingly, any eventual decision to buy Cobe tokens must only be made following receipt of the final paper, and tokens cannot be purchased until the final paper has been issued by Cobe when all final regulatory requirements have been satisfied. This paper is not a prospectus, product disclosure statement, or other regulated offer docu- ment. It has not been endorsed by, or registered with, any governmental authority or regulator. The distribution and use of this paper, including any related advertisement or marketing ma- terial, and the eventual sale of tokens, may be restricted by law in certain jurisdictions, and potential purchasers of tokens must inform themselves about those laws and observe any such restrictions. If you come into possession of this paper, you should seek advice on, and observe any such restrictions relevant to your jurisdiction, including without limitation the applica- ble restrictions set out in the Regulators' Statements on Initial Coin Offerings at the website of the International Organization of Securities Commissions ('IOSCO') (https://www.iosco. org/publications/?subsection=ico-statements). Restrictions are subject to rapid change. If you fail to comply with such restrictions, that failure may constitute a violation of applicable law. By accessing this paper, you agree to be bound by this requirement.

References

[1] Y. Xiao, N. Zhang, W. Lou, Y. T. Hou, A survey of distributed consensus protocols for blockchain networks, IEEE Communications Surveys & Tutorials 22 (2) (2020) 1432–1465.

[2] D. Gage, E. Laub, B. McGarry, Cellular automata: is rule 30 random, in: Proceedings of the Midwest NKS Conference, Indiana University, Citeseer, 2005.

[3] M. Bartoletti, L. Galletta, M. Murgia, A true concurrent model of smart contracts executions, in: International Conference on Coordination Languages and Models, Springer, 2020, pp. 243–260.

[4] C. Regueiro, I. Seco, S. de Diego, O. Lage, L. Etxebarria, Privacy-enhancing distributed protocol for data aggregation based on blockchain and homomorphic encryption, Information Processing & Management 58 (6) (2021) 102745.

[5] D. Hopwood, S. Bowe, T. Hornby, N. Wilcox, Zcash protocol specification, GitHub: San Francisco, CA, USA (2016) 1.

[6] E. Ben-Sasson, A. Chiesa, E. Tromer, M. Virza, Succinct {Non-Interactive} zero knowledge for a von neumann architecture, in: 23rd USENIX Security Symposium (USENIX Security 14), 2014, pp. 781–796.

[7] D. Yaga, P. Mell, N. Roby, K. Scarfone, Blockchain technology overview. national institute of standards and technology, Tech. rep., Technical Report (2018).

[8] L. Besançon, C. F. Da Silva, P. Ghodous, Towards blockchain interoperability: Improving video games data exchange, in: 2019 IEEE international conference on blockchain and cryptocurrency (ICBC), IEEE, 2019, pp. 81–85.

[9] Q. Zhu, S. W. Loke, R. Trujillo-Rasua, F. Jiang, Y. Xiang, Applications of distributed ledger technologies to the internet of things: A survey, ACM computing surveys (CSUR) 52 (6) (2019) 1–34.

[10] R. Belchior, A. Vasconcelos, S. Guerreiro, M. Correia, A survey on blockchain interoperability: Past, Present, and Future Trends (2020).

[11] I. Sergey, V. Nagaraj, J. Johannsen, A. Kumar, A. Trunov, K. C. G. Hao, Safer smart contract programming with scilla, Proc. ACM Program. Lang. 3 (OOPSLA) (2019) 185:1–185:30. doi:10.1145/3360611.
URL https://doi.org/10.1145/3360611

[12] P. Daian, Y. Falcone, P. O. Meredith, T. Serbanuta, S. Shiriashi, A. Iwai, G. Rosu, Rv-android: Efficient parametric android runtime verification, a brief tutorial, in: RV, Vol. 9333 of Lecture Notes in Computer Science, Springer, 2015, pp. 342–357.

[13] K. Bhargavan, A. Delignat-Lavaud, C. Fournet, A. Gollamudi, G. Gonthier, N. Kobeissi, N. Kulatova, A. Rastogi, T. Sibut-Pinote, N. Swamy, et al., Formal verification of smart contracts: Short paper, in: Proceedings of the 2016 ACM workshop on programming languages and analysis for security, 2016, pp. 91–96.

[14] Z. Wang, H. Jin, W. Dai, K. R. Choo, D. Zou, Ethereum smart contract security research: survey and future research opportunities, Frontiers Comput. Sci. 15 (2) (2021) 152802.

[15] D. He, Z. Deng, Y. Zhang, S. Chan, Y. Cheng, N. Guizani, Smart contract vulnerability analysis and security audit, IEEE Netw. 34 (5) (2020) 276–282.

[16] Z. Zheng, S. Xie, H.-N. Dai, X. Chen, H. Wang, Blockchain challenges and opportunities: A survey, International Journal of Web and Grid Services 14 (2018) 352. doi:10.1504/IJWGS.2018.095647.

[17] D. C. Nguyen, P. N. Pathirana, M. Ding, A. Seneviratne, Integration of blockchain and cloud of things: Architecture, applications and challenges, IEEE Communications Surveys Tutorials 22 (4) (2020) 2521–2549.

[18] M. Saad, J. Spaulding, L. Njilla, C. Kamhoua, S. Shetty, D. Nyang, D. Mohaisen, Exploring the attack surface of blockchain: A comprehensive survey, IEEE Communications Surveys & Tutorials 22 (3) (2020) 1977–2008.

[19] C. M. World, Letter of credit fees. (2018 [Online]).
URL http://www.creditmanagementworld.com/letterofcredit/lcinternationallocfees.html

[20] W. T. Organization, Trade finance and smes bridging the gaps in provision., Tech. rep. (2017 [Online]).
URL https://www.wto.org/english/res_e/booksp_e/tradefinsme_e.pdf

[21] L. Wood, Global brand counterfeiting report 2018-2020., Tech. rep. (2018 [Online]).
URL https://www.businesswire.com/news/home/20180515005775/en/Global-Brand-Counterfeiting-Report-2018-2020---ResearchAndMarkets.com

[22] A. Boisseau, Dairy plants seek sustainability solutions. (2021 [Online]).
URL https://www.dairyfoods.com/articles/95085-dairy-plants-seek-sustainability-solutions

Legal Disclaimer Nothing in this Yellow Paper is an offer to sell, or the solicitation of an offer to buy, any tokens. Cobe is publishing this Yellow Paper solely to receive feedback and comments from the public. If and when Cobe offers for sale any tokens (or a Simple Agreement for Future Tokens), it will do so through definitive offering documents, including a disclosure document and risk factors. Those definitive documents also are expected to include an updated version of this Yellow Paper, which may differ significantly from the current version. If and when Cobe makes such an offering in the United States, the offering likely will be available solely to accredited investors.

Nothing in this Yellow Paper should be treated or read as a guarantee or promise of how Cobe business or the tokens will develop or of the utility or value of the tokens. This Yellow Paper outlines current plans, which could change at its discretion, and the success of which will depend on many factors outside of Cobe control, including market-based factors and factors within the data and cryptocurrency industries, among others. Any statements about future events are based solely on Cobe analysis of the issues described in this Yellow Paper. That analysis may prove to be incorrect.