# Towards a Scalable Dual-Sided Blockchain Architecture with Concurrency Protocols

A. Nazir[1], M. Singh[1], G. Destefanis[2], M. Kassab[3], J. Memon[1], R. Neykova[2], and R. Tonelli[4]
[1]*Cobe LTD*, U.K. {anjum, michael, jamshed}@cobe.network
[2]*Brunel University London*, U.K. {giuseppe.destefanis, rumyana.neykova}@brunel.ac.uk
[3]*Pennsylvania State University*, muk36@psu.edu
[4]*University of Cagliari, IT*, roberto.tonelli@unica.it

***Abstract*—This paper explores the potential of blockchain technology to improve cross-border trade by providing a secure and transparent way to track and verify the movement of goods, services, and funds across borders. By creating a tamper-proof record of transactions, blockchain can reduce fraud and increase transparency in the supply chain, as well as streamline the process of documenting and verifying transactions. The paper presents a new technology proposal, developed by Cobe, for a comprehensive cross-border trade ecosystem that includes both native permissioned and permissionless chains, connected via a relay system, and featuring a suite of cross-border trade APIs. The study also examines the concurrency protocols in the context of the proposed dual-sided blockchain architecture, providing a comprehensive analysis of the proposed system's potential to improve cross-border trade.**

***Index Terms*—concurrency, blockchain, fork-chain**

## I. Introduction

One of the major challenges in cross-border trade is the coordination and verification of transactions among multiple parties, frequently situated in different countries. Blockchain technology presents a potential solution by providing a secure, transparent, and efficient method for tracking and verifying the movement of goods, services, and funds across international borders. One of the key benefits of using blockchain in cross-border trade is the ability to create an immutable record of transactions, which can help reduce fraud and increase transparency in the supply chain. Additionally, it can simplify the process of documenting and verifying transactions, minimizing the need for manual paperwork and reducing the likelihood of errors.

Although the application of blockchain in cross-border trade is still in its nascent stage, it holds enormous potential to enhance the efficiency and transparency of global trade. For instance, the Global Shipping Business Network (GSBN) [4] is a consortium of leading shipping companies that utilizes blockchain to streamline and automate international shipping processes. Similarly, the TradeLens platform [7], developed by IBM and Maersk, is a blockchain-based system that tracks and verifies the movement of goods across international borders. However, there is currently no blockchain-based platform that provides a comprehensive solution for cross-border trade in its entirety.

In cross-border trading, some applications may require a blockchain with fixed transaction fees, while others may prefer a variable fee structure. Currently, there is no blockchain network that offers both solutions in a comprehensive platform.

To address this challenge, Cobe presents a new technology proposal for a comprehensive cross-border trade ecosystem that incorporates state-of-the-art innovations in the blockchain field at different levels, including consensus protocols. The proposed solution is a dual blockchain architecture that includes both native permissioned and permissionless chains. The permissioned chain uses a "Proof of Authority" consensus and offers fixed transaction fees, while the permissionless chain uses a "Delegated Proof of Stake" consensus and offers variable transaction fees. The two chains are interconnected via a Relay system that executes the Cross Communication Blockchain (CCB) protocol, as illustrated in Figure 1. The Nucleus platform also accompanies this system, providing a comprehensive suite of cross-border trade APIs that dApp developers can utilize when creating applications on the blockchain.

When designing the proposed cross-border trading blockchain platform, scalability was a key consideration. Scalability refers to the ability of a blockchain network to handle a high volume of transactions without experiencing delays or congestion. In order to address scalability challenges, the following factors were taken into account:

1) Limited capacity: To avoid network congestion and long wait times for transactions to be processed, the proposed platform is designed to have a higher capacity for the number of transactions that can be processed in a given period of time.
2) Data size: The proposed platform is designed to minimize the size of data being stored and processed to reduce the resource-intensive nature of node participation in the network.
3) Consensus mechanisms: To increase scalability, the proposed platform employs consensus mechanisms that are less resource-intensive than traditional proof-of-work mechanisms [16].

Another important consideration in the design and operation

This article has been accepted for publication in a future proceedings of this conference, but has not been fully edited. Content may change prior to final publication. Citation information: DOI10.1109/ICSA-C57050.2023.00056, 2023 IEEE 20th International Conference on Software Architecture Companion (ICSA-C)

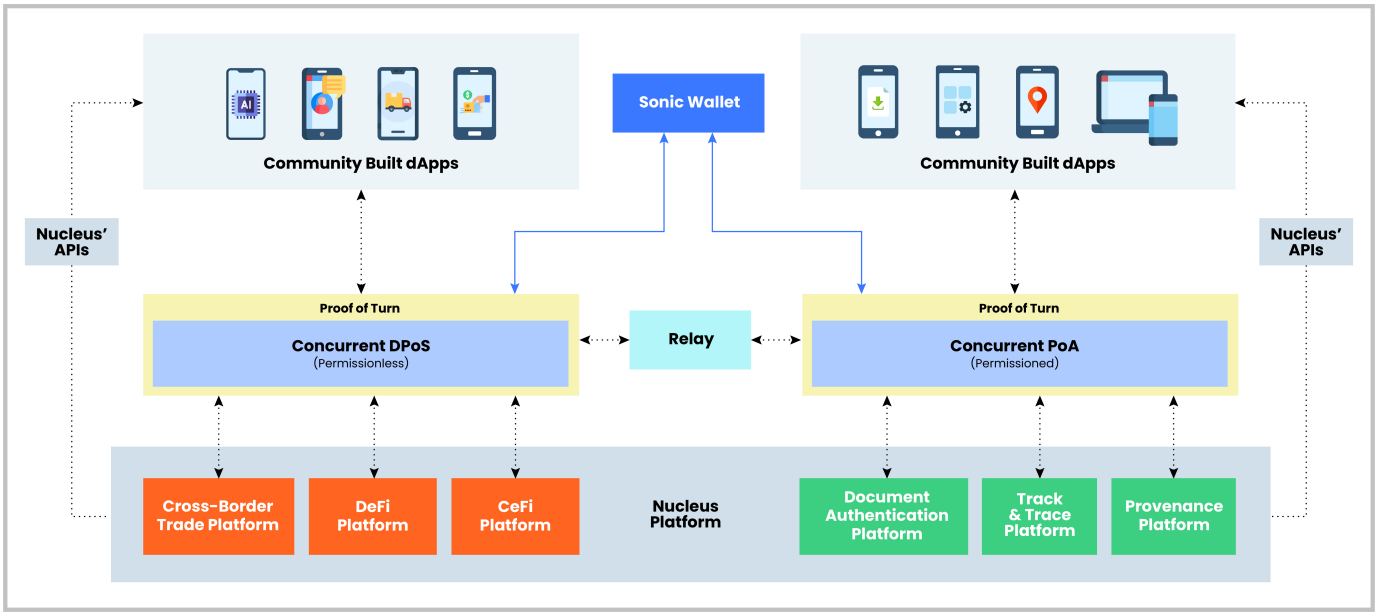


Figure 1: Proposed dual-sided blockchain architecture for cross-border trading

of the proposed cross-border trading blockchain platform is implementing a set of concurrency tactics, which is the main focus of this paper. Concurrency in blockchain refers to the ability of a blockchain network to process multiple transactions simultaneously. In particular, the proposed blockchain network will utilize two state-of-the-art techniques, namely:

1) Sharding, a technique used to improve scalability by dividing the network into smaller units, or "shards," which can process transactions independently of one another. This allows the network to process transactions concurrently, rather than sequentially, increasing overall transaction throughput. Our design incorporates two types of sharding; static and dynamic, which will be discussed in this paper.
2) The CTEV protocol, which enables multiple transactions within a single block to be executed in parallel, boosting the speed at which they are processed.

The goal of this paper is to present the concurrency protocols in the context of the proposed dual-sided blockchain architecture. The remainder of this paper is structured as follows: Background is summarized in Section 2. Section 3 presents the implemented dynamic sharding approach, while Section 4 presents the static approach. Then, Section 5 presents the implemented synchronization mechanism that complements the static and dynamic sharding approaches. Section 6 presents the Concurrent Transactions Protocol. Section 7 discusses related work, and finally, Section 8 presents conclusions.

## II. Background

Improving concurrency is important in cross-border trade in order to ensure that transactions are processed quickly and efficiently, minimizing delays and enabling goods and services to be delivered on time. But there are several factors that can impact the concurrency of a blockchain network; for example:

1) Consensus mechanism: Some consensus mechanisms, such as proof-of-work (used by Bitcoin), are designed to process transactions sequentially, which can limit the concurrency of the network. Other consensus mechanisms, such as proof-of-stake, may be more conducive to the concurrent processing of transactions.
2) Network architecture: The architecture of a blockchain network can also impact its concurrency. For example, some networks are designed to allow transactions to be processed in parallel, while others rely on sequential processing.
3) Transaction throughput: The transaction throughput of a blockchain network refers to the number of transactions that can be processed per second. Higher transaction throughput can enable a blockchain network to process transactions more quickly and efficiently [10; 17].

To combat the above factors, and enabling blockchain networks to process more transactions per second, blockchain networks may utilize a database partitioning technique; sharding. In particular, sharding splits a blockchain's entire network into smaller partitions, known as "shards." Each shard is comprised of its own data, making it distinctive and independent when compared to other shards.

Several research studies exist that provide an overview of the various approaches that have been proposed for implementing sharding in blockchain networks along with the benefits and drawbacks of each approach; for example [14], [13], [19], and [15]. Nevertheless, to the extent of our knowledge, only a few blockchains have attempted to implement sharding, and only to a limited extent.

Among the several ways that sharding can be implemented

This article has been accepted for publication in a future proceedings of this conference, but has not been fully edited. Content may change prior to final publication. Citation information: DOI10.1109/ICSA-C57050.2023.00056, 2023 IEEE 20th International Conference on Software Architecture Companion (ICSA-C)

in a blockchain network, static sharding and dynamic sharding are two common approaches. Static sharding involves dividing the network into a fixed number of shards, which are predetermined at the time the network is created. In static sharding, the number of shards is fixed, and the assignment of transactions to specific shards is also fixed.

On the other hand, dynamic sharding involves dividing the network into a variable number of shards, which can change over time based on the needs of the network. In dynamic sharding, the number of shards can be adjusted as needed in order to improve the scalability and efficiency of the network.

There are advantages and disadvantages to both static and dynamic sharding. Static sharding can be simpler to implement, as the number of shards is fixed and does not need to be adjusted over time. However, static sharding may be less flexible than dynamic sharding, as the number of shards is fixed and cannot be adjusted to meet the changing needs of the network. Alternatively, dynamic sharding can be more flexible and scalable than static sharding, as the number of shards can be adjusted over time to meet the needs of the network. However, dynamic sharding can also be more complex to implement, as it requires the ability to adjust the number of shards and the assignment of transactions to specific shards in real-time.

Overall, the choice between static and dynamic sharding depends on the specific needs and requirements of the blockchain network. Both approaches have their own advantages and disadvantages, and the most appropriate approach will depend on the specific use case and the goals of the network.

In our proposed dual-blockchain architecture, both dynamic and static shardings were implemented:

1) Dynamic: implemented through the **"Load Aware concurrency protocol"** such that when the transaction load on the network increases, this enables the creation of parallel fork-chains (fchains). 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. At the end of each round, all the fork-chains created are integrated into the main chain.
2) Static: The **"DApp Based Concurrent Fork-Chains protocol"**, enabling fork-chains (fchains) to be created for different dApps, which then integrates into the main chain after each round, reducing the load on the main chain.

The next three sections present how both approaches are implemented in our proposed blockchain architecture.

## III. Load-Aware Concurrent Fork-Chains

Load-aware fchains are fork-chains that will be created dynamically on demand as per the requirements of the blockchain network when there is a need to increase transactions per seconds (tps) or throughput of the network. The protocol is realized through the following steps:

1) At the beginning of each round; where a round refers to the time it takes to find a new block, a few validators are chosen randomly to monitor the transaction processing system, which includes a transaction pool.
2) If there is a need for fork-chains, the monitor nodes will multicast a fork() message on the network, and a fork-block will be added in the parent chain.
3) From this point, the 'Random Block Schedule' generated at the beginning of the 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 8.

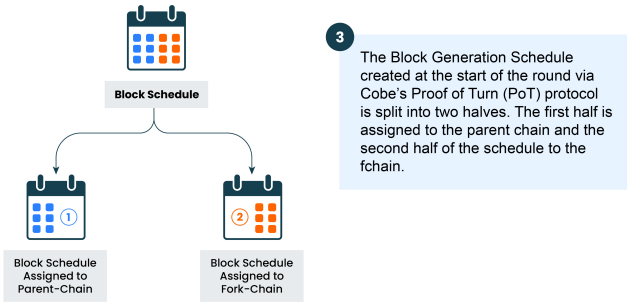


Figure 2: Block schedule splitting process

4) All fchains and the parent chain will use the same transaction pool, as shown in Figure 3.

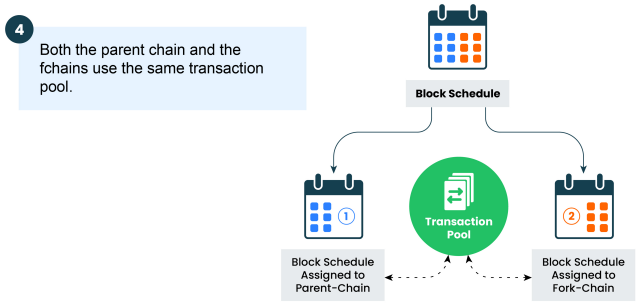


Figure 3: Parent and fchains will share the same transaction pool

5) A new ftable will be created. In load-aware concurrent fork chains, a ftable is created, which is used to track which fork chain processed the transaction related to which address. It consists of (i) the sender's or receiver's address and (ii) the chain ID.
When a new transaction is submitted to the transaction pool, the transaction pool manager searches the ftable for the sender and receiver's addresses. If the address is not found in the ftable, this means that it is a new independent entry. The transaction pool manager will then assign this transaction to any one of the chains in a round-robin manner and update the ftable.
If a previous entry of the sender and receiver addresses is found in the ftable, the transaction pool manager will assign the transaction to the same chain.
This ftable will include the following fields (Figure 4):
   a) address (either of sender or receiver).

This article has been accepted for publication in a future proceedings of this conference, but has not been fully edited. Content may change prior to final publication. Citation information: DOI10.1109/ICSA-C57050.2023.00056, 2023 IEEE 20th International Conference on Software Architecture Companion (ICSA-C)

b) fchain ID, that is used to track which fork-chain processed transactions related to which wallet.

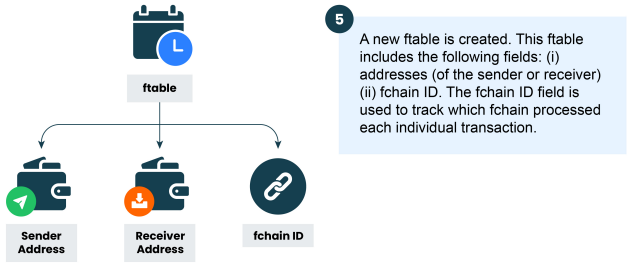


Figure 4: Creation of ftable

6) When a new transaction is submitted to the transaction pool, the transaction pool manager 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 pool manager will then assign this transaction to any one of the chains in a round-robin manner and update the ftable, as shown in Figure 5.

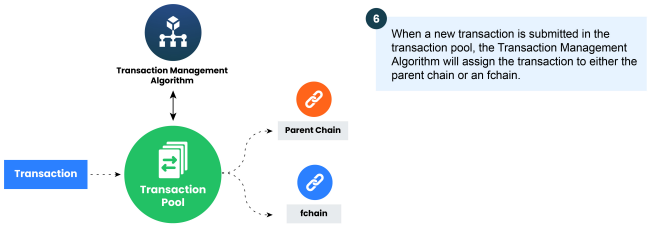


Figure 5: Transaction assignment process for fchains

7) If a previous entry of the sender and receiver addresses is found in the ftable, the transaction pool manager will assign the transaction to the same chain. The whole process is represented in Figure 6.

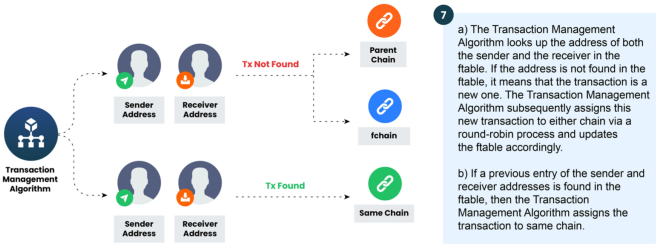


Figure 6: Transaction assignment process for fchains

## IV. DApps-Based Concurrent Fork-Chains

"DApps-Based Concurrent Fork-Chains" represents a static approach with which the proposed blockchain presented in Figure 1 will host multiple parallel independent chains. Each subchain will be used for a specific dApp(s) hosted on the platform. That is why this approach is known as dApp-based concurrent fork-chains. Features and operation of the dApp-based parallel fork-chains are presented below:

1) The blockchain will host multiple dApps, where dApps will be organized into groups.
2) At the start, a unique subchain (fchain) will be created for each group of dApps, as shown in Figure 7. For simplicity, here we assume that each dApp is its own group.

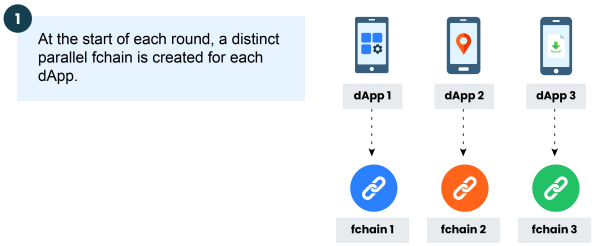


Figure 7: Each dApp will host its own separate fchain

3) Each fchain will run separate instances of the Consensus Protocol; thus each chain will have separate block schedule. This process is shown in Figure 8.

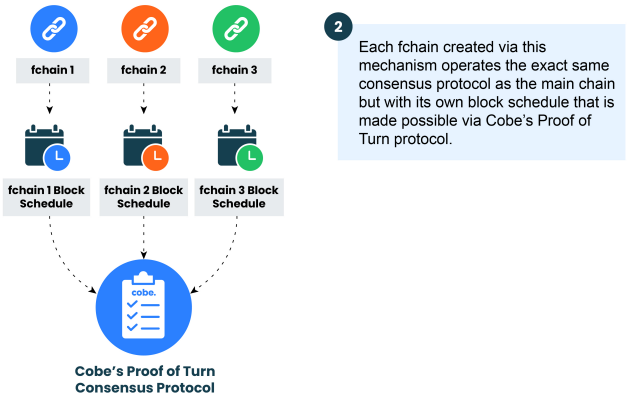


Figure 8: DApp-based block schedule for each fchain

4) At the end of each round, Each dApp-based fchain will be synchronized to main chain, as shown in Figure 9.

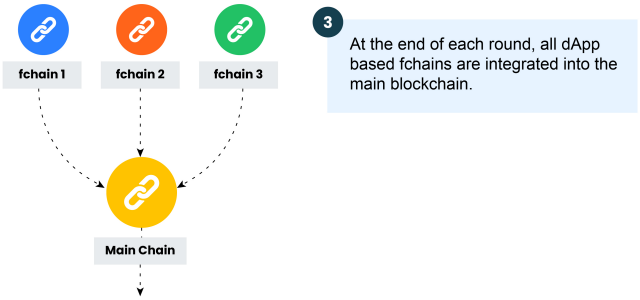


Figure 9: Synchronization of dApp-based fchains

This article has been accepted for publication in a future proceedings of this conference, but has not been fully edited. Content may change prior to final publication. Citation information: DOI10.1109/ICSA-C57050.2023.00056, 2023 IEEE 20th International Conference on Software Architecture Companion (ICSA-C)

5) Throughput of the network is directly proportional to the number of independent chains.

## V. Synchronization of fchains

Chain synchronization is an important step when nested or parallel chains are operated, as it helps to ensure the integrity and consistency of the shard across all nodes in the network. Chain synchronization is an ongoing process that can be invoked when a round is completed, or the network load falls. Chain synchronization is achieved with the help of 'Monitor' nodes, as presented below:

1) The monitor nodes; responsible for the synchronization of nested subchains, will multicast a join-message on the network.
2) Upon receiving a join-message, validators will stop further block creation. Typically, the join-message is created and shared at the end of an epoch or round, but this will be adjusted depending on the overhead of the synchronization.
3) All fork-chains are merged or linked together with the help of a join-block.
4) The join-block is basically a jumbo block that will contain the reference (pointer) to the last block of each sub/parallel chain.
5) A join-block is created after each epoch to synchronize the state of fork-chains across the nodes. It is possible to increase or decrease the number of nested chains after each join-block depending on the load on the network. Figure 10 visualizes the synchronization process.

## VI. Concurrent Transaction Protocol (CTEV Algorithm)

Concurrent transactions per block in a blockchain refer to the processing of a number of transactions simultaneously within a single block of the blockchain. By allowing more transactions to be processed within a single block, concurrent transactions can help improve the overall performance, scalability of the network. In addition, in some cases, increasing the number of concurrent transactions per block can also improve the security of the blockchain, as it allows more transactions to be validated and included in the blockchain in a shorter period of time. This can help prevent potential attacks on the network by making it more difficult for attackers to alter the blockchain or reverse transactions.

In our proposed blockchain architecture, we utilize a concurrent Transaction Protocol (CTEV) based on static transaction analysis techniques to execute and verify transactions concurrently within a block.

The steps of the CTEV algorithm are described below:

1) Gather from the network a set of transactions and put them in an arbitrary linear sequence, which is the mined block, as shown in Figure 11.
2) Compute a relation, using the Concurrent Transaction Execution and Verification (CTEV) protocol [9], as shown in Figure 12, which evaluates whether two transactions are independent. If two transactions are independent, they can be executed in any order, with no effect on the final state of the blockchain.
3) Construct an occurrence net of transactions, following the algorithm in [9]. 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 13.
4) Execute transactions concurrently according to the occurrence net, exploiting the available parallelism on the node; see Figure 14.

We are aware of the several challenges that can arise when increasing the number of concurrent transactions per block in a blockchain, and we are currently in the process of evaluating the CTEV protocol against these challenges. In particular, with respect to:

1) Centralization: One challenge is that increasing the number of concurrent transactions per block can lead to centralization, as it requires more resources to validate and process larger blocks. This can make it more difficult for smaller participants in the network to compete with larger, more well-resourced participants, which can lead to a less decentralized network.
2) Security risks: Increasing the number of concurrent transactions per block can also introduce new security risks to the blockchain. For example, if the block size is increased too much, it can make it more difficult for validators to validate and secure the blockchain, which could lead to a higher risk of attacks on the network.
3) Difficulty in reaching consensus: One challenge is that increasing the number of concurrent transactions per block can make it more difficult for participants in the network to reach consensus on the contents of a block. This is because there are more transactions to consider, which can make it more difficult for validators to agree on which transactions should be included in the block.
4) Increased storage requirements: Another challenge is that increasing the number of concurrent transactions per block can also increase the storage requirements for the blockchain. This is because larger blocks take up more space on the network, which can make it more difficult for users with limited storage capacity to participate in the network.
5) Network congestion: In some cases, increasing the number of concurrent transactions per block can also lead to network congestion, as there may be more transactions than the network can handle at any given time. This can lead to delays in processing transactions and can make it more difficult for users to access the blockchain.
6) Changes to the economic model: Finally, increasing the number of concurrent transactions per block can also require changes to the economic model of the blockchain, as it can affect the incentives for validators and other participants in the network. This can be a

This article has been accepted for publication in a future proceedings of this conference, but has not been fully edited. Content may change prior to final publication. Citation information: DOI10.1109/ICSA-C57050.2023.00056, 2023 IEEE 20th International Conference on Software Architecture Companion (ICSA-C)

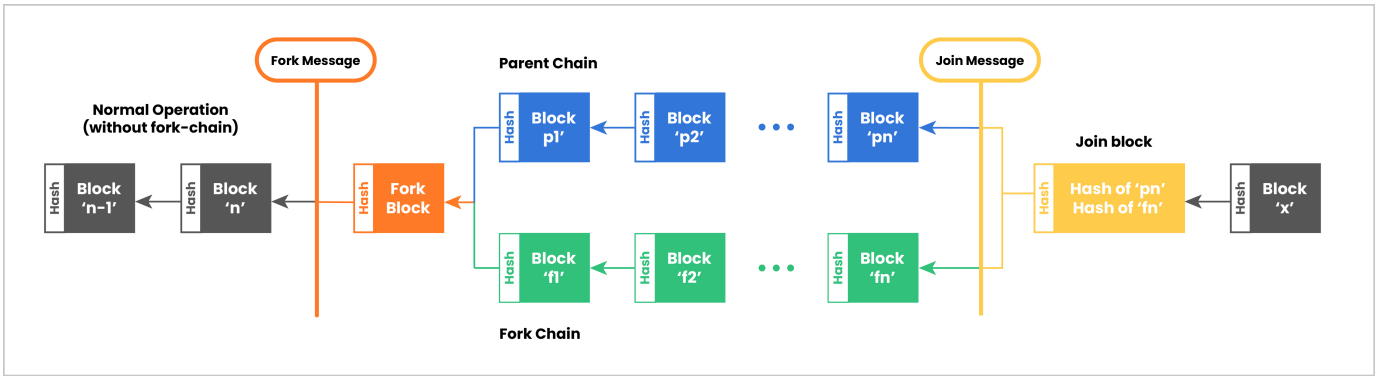


Figure 10: Fork fchain synchronization

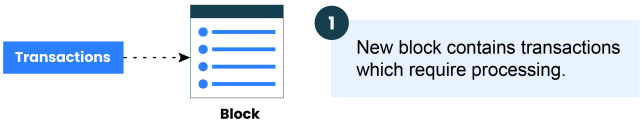


Figure 11: Transaction block that contains transactions

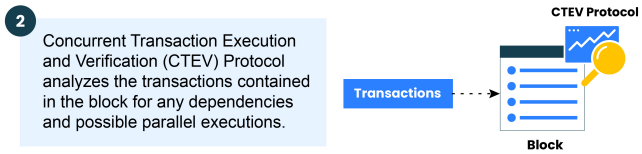


Figure 12: CTEV analyzes the transactions in the block

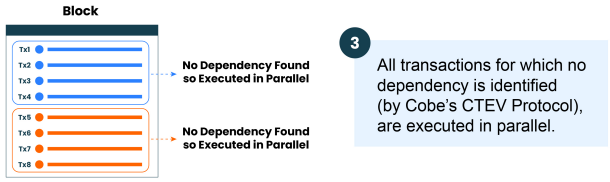


Figure 13: Construction of separate occurrence net

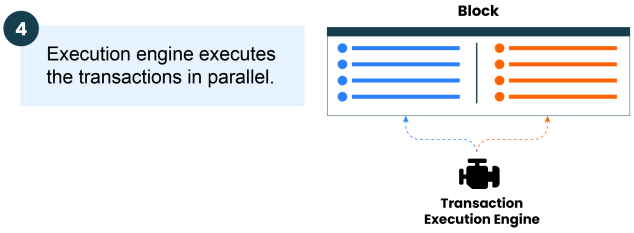


Figure 14: Parallel execution of transactions

complex and controversial process, as it can have far-reaching effects on the overall structure and operation of the blockchain.

## VII. Related Work

Both sharding and concurrent transactions per block are active areas of research and development in the blockchain industry, and there are many ongoing research projects that are focused on improving the scalability of blockchain networks by using both approaches.

For example, Zilliqa [8] is a blockchain platform that uses a form of sharding called "network sharding" to improve the scalability of its network. In network sharding, the network is divided into smaller units, or "shards," which can operate independently of one another and process transactions concurrently.

Ethereum 2.0 [3] is a second example and it is a major upgrade to the Ethereum blockchain that aims to improve the scalability of the network through the use of sharding. Ethereum 2.0 uses a form of sharding called "state sharding," which involves dividing the blockchain's state (i.e., the current state of all transactions and accounts on the network) into smaller units, which can be processed concurrently by different nodes.

The Coda Protocol [1] is a third blockchain platform example that uses a form of sharding called "recursive zk-SNARKs" to improve the scalability of its network. In recursive zk-SNARKs, the blockchain's state is compressed into a small, constant-sized "snark," which can be verified by nodes in the network without the need for each node to store the entire state of the blockchain.

Other examples include Elrond blockchain platform [2] that uses a form "Adaptive State Sharding", Harmony blockchain [5] that uses a form of sharding called "Cross-Shard Communication", and QuarkChain blockchain platform [6] that uses "Horizontal Sharding".

On the other hand, research on concurrent transactions per block in a blockchain typically focuses on ways to increase the number of transactions that can be processed within a single block, as this can help improve the overall performance and scalability of the blockchain. One approach that has been explored in research is to increase the block size, which allows more transactions to be included in a single block' e.g., [11]. However, increasing the block size can also lead to problems

with centralization, as it requires more resources to validate and process larger blocks. Other research has focused on alternative approaches to increasing the number of concurrent transactions per block, such as using off-chain transactions or using layer 2 protocols to process transactions outside of the main blockchain; e.g. [18]. Some blockchain networks, such as Ethereum, have implemented these types of approaches to improve scalability and allow for more concurrent transactions per block. Researchers have also studied the trade-offs associated with different approaches to increasing concurrent transactions per block, including the impact on performance, scalability, security, and cost [12].

## VIII. Conclusion

Cobe Blockchain technology has the potential to greatly improve the efficiency and transparency of cross-border trading while acknowledging that scalability is one of the major challenges in blockchain networks in general. This paper presented the scalability tactics that the Cobe Blockchain network is utilizing, namely sharding and concurrent execution of transactions. Sharding enables parallel execution of independent chains, while concurrent execution of transactions allows for faster processing of data blocks. The proposed blockchain also supports both dynamic and static sharding, allowing for the creation of shards on-demand to handle increased network load and the deployment of dApps to specific fork-chains. Overall, the Cobe Blockchain presents a comprehensive solution for addressing scalability in cross-border trading.

## References

[1] codaprotocol. https://codaprotocol.com/ (2022)
[2] elrond. https://elrond.com/ (2022)
[3] ethereum. https://ethereum.org/en/upgrades/ (2022)
[4] "global shipping business network (gsbn)". https://www.gsbn.trade/ (2022)
[5] harmony. https://www.harmony.one/ (2022)
[6] quarkchain. https://quarkchain.io/ (2022)
[7] "tradelens". https://www.tradelens.com/ (2022)
[8] zilliqa. https://www.zilliqa.com/ (2022)
[9] Bartoletti, M., Galletta, L., Murgia, M.: A true concurrent model of smart contracts executions. In: International Conference on Coordination Languages and Models. pp. 243–260. Springer (2020)
[10] Bartolucci, S., Destefanis, G., Ortu, M., Uras, N., Marchesi, M., Tonelli, R.: The butterfly "affect": Impact of development practices on cryptocurrency prices. EPJ Data Science **9**(1), 21 (2020)
[11] Buterin, V., Wilcke, J.: The ethereum blockchain size will exceed 1tb, and it's a good thing" (2016)
[12] Croman, K., Decker, C., Eyal, I., Gencer, A.E., Juels, A., Kosba, A., Miller, A., Saxena, P., Shi, E., Gün Sirer, E., et al.: On scaling decentralized blockchains. In: International conference on financial cryptography and data security. pp. 106–125. Springer (2016)
[13] Dang, H., Dinh, T.T.A., Loghin, D., Chang, E.C., Lin, Q., Ooi, B.C.: Towards scaling blockchain systems via sharding. In: Proceedings of the 2019 international conference on management of data. pp. 123–140 (2019)
[14] Hashim, F., Shuaib, K., Zaki, N.: Sharding for scalable blockchain networks. SN Computer Science **4**(1), 1–17 (2023)
[15] Kaur, G., Gandhi, C.: Scalability in blockchain: Challenges and solutions. In: Handbook of Research on Blockchain Technology, pp. 373–406. Elsevier (2020)
[16] Nazir, A., Singh, M., Destefanis, G., Memon, J., Neykova, R., Kassab, M., Tonelli, R.: An optimized concurrent proof of authority consensus protocol. In: Proceedings of the 2023 International Workshop On Blockchain Oriented Software Engineering. p. to appear (2023)
[17] Pierro, G.A., Rocha, H., Ducasse, S., Marchesi, M., Tonelli, R.: A user-oriented model for oracles' gas price prediction. Future Generation Computer Systems **128**, 142–157 (2022)
[18] Poon, J., Dryja, T.: The bitcoin lightning network: Scalable off-chain instant payments (2016)
[19] Wang, G., Shi, Z.J., Nixon, M., Han, S.: Sok: Sharding on blockchain. In: Proceedings of the 1st ACM Conference on Advances in Financial Technologies. pp. 41–61 (2019)