Privacy-Preserving Hyperledger Fabric Implementation for Regulated Carbon Markets with Trusted Execution Environments and Spatio-Temporal Data

Johann Westphall
Computer Security Laboratory,
Department of Informatics and Statistics
Federal University of Santa Catarina (UFSC)
Florianópolis, Brazil
johann.westphall@posgrad.ufsc.br

Jean E. Martina
Computer Security Laboratory,
Department of Informatics and Statistics
Federal University of Santa Catarina (UFSC)
Florianópolis, Brazil
jean.martina@ufsc.br

***Abstract*—This paper presents a pilot implementation of a Blockchain Emission Trading System (BETS) tailored for regulated environments. Building on a theoretical BETS model, our design integrates privacy-preserving auctions executed within Trusted Execution Environments (TEEs) using Microsoft Confidential Containers, leverages Hyperledger Fabric's modular and permissioned blockchain for secure carbon credit lifecycle management, and employs Hyperledger Cacti for cross-chain interoperability. Our approach balances transparency and privacy by combining public state data, private collections, and confidential off-chain computations. We validate our implementation through extensive functional testing, ensuring trustworthiness via hardware attestation and cryptographic verification. This work addresses key challenges such as empirical validation, privacy, and regulatory compliance, providing a foundation for interoperable, transparent, and privacy-aware carbon markets.**

***Index Terms*—Emission Trading Systems, Hyperledger, Microsoft Confidential Containers, AMD SEV-SNP, Privacy, Interoperability,**

I. Introduction

The Kyoto Protocol [1] and the Paris Agreement [2] recognize the need for market-based technology in cooperation with stakeholders to reduce Greenhouse Gas (GHG) emissions. However, authors point out that the traditional markets face several challenges regarding transparency, trust, corruption, and double-counting [3, 4]. To address these issues, blockchain's inherent properties, such as decentralization, immutability, transparency, and automation, can play a role in the design of the carbon market [5, 6].

Even though blockchain technology is theoretically considered a promising solution for carbon markets, authors still point out research gaps and challenges to be resolved. The mentioned gaps include empirical validations, scalability, interoperability, novel privacy preservation methods, and data quality assurance [5, 7, 8, 9]. Regarding legal compatibility, the implementations must align with regulatory frameworks [5, 10]. Also, frameworks should consider geographical, climatic, and industry variability [7, 8]. As an example of regulatory legislation, the recent Brazilian carbon market law can be used as a reference for identifying relevant stakeholders and processes in a regulated environment [11].

To bring Blockchain Emission Trading System (BETS) to a real implementation, privacy, data validation, and interoperability are relevant challenges to be addressed. Trusted Execution Environments (TEEs) are suggested as a privacy-preserving technology for processing sensitive data for future carbon markets [7]. Generally, TEEs in the blockchain environment already help processes such as authentication, confidential execution, and integrity [12]. Since the on-chain data help determine credits minting and burning amounts, it is crucial to ensure its trustworthiness. Thus, they may come from government databases, verified providers, or via oracle mechanisms [13]. Interoperation enables heterogeneous blockchains to communicate and exchange data, which is essential for international carbon markets [10, 14].

In this work, we present a model for a blockchain-based carbon market in a regulated environment, including the stakeholders most commonly found in the literature. This model mints, burns, and allows trading credits based on georeferenced data to promote effective carbon sequestration. The data comes from trusted sources, preferably with legal endorsement. It also considers parties' privacy concerns. After that, we focus on the initial implementation design and address how the state-of-the-art technologies can be used for the implementation. Mainly, our implementation involves technologies such as Microsoft Confidential Containers (MCC) (as TEE technology), Hyperledger Fabric (as the blockchain platform), and Hyperledger Cacti (as the cross-chain interoperability framework) [15, 16, 17]. It foresees the integration with government databases, verified data providers, or oracle mechanisms for data validation [13, 18].

This work was supported by the Brazilian Federal Agency for Support and Evaluation of Graduate Education (CAPES - ID ROR:00x0ma614).

The rest of this paper is organized as follows: Section II presents the base theoretical model and the technology stack rationale. Section III details the core parts of the implementation and its validation. Finally, Section IV presents the conclusions and future work.

II. Model, Architecture, and Technology Decisions

In this section, we present an overview of the architectural model and the rationale behind our choice of technology stack.

A. Model

Our proposed model, displayed in Fig. 1, introduces a modular BETS tailored for regulated carbon markets. It operates on a consortium blockchain, granting governments primary control while involving other actors like sinkers (e.g., farmers), emitting companies, and auditors to ensure transparency and data availability. Buyers interact with the system using pseudonymous identities, supported by anonymous credentials to prevent identity inference from transaction patterns. On the other hand, sellers transact with their credits tied to their location and identity. Smart contracts enforce the system's logic, from participant registration to credit management, creating a secure and auditable environment for all stakeholders.

The model's functionality is anchored in trusted data and adaptable policies. Data from government databases and verified providers acts as oracles, feeding information about carbon-sinking and emitting activities into the system. This data is managed with strong considerations for privacy, trustworthiness, and scalability, using techniques like data commitments and off-chain storage. A technical committee defines methodologies and dynamic policies that use this data to calculate a "multiplier." This multiplier adjusts the value of carbon credits based on factors like data reliability or the relationship between a carbon sink and an emitter. When the policy considers this relationship, we call it a *coupled policy*, while the ones that only consider either one are the *independent policies*. Credits are minted periodically to ensure sustained environmental commitment and are burned by companies to offset their emissions, all governed by these on-chain policies.

Market operations are centered around a sophisticated and privacy-preserving auction module. The auction mechanism adapts to the complexity of the market's policies, handling both simple fungible credit exchanges and complex one-to-one matchings for non-fungible credits derived from coupled policies. To protect sensitive financial and operational data, the auction is executed off-chain using trusted technologies like TEE or Zero-Knowledge Proof (ZKP), with only the results and an execution proof being published on-chain. Transaction settlement can be handled either through a trusted entity issuing a virtual token or via decentralized Hash Time-locked Contract (HTLC) on public blockchains. Finally, the model supports cross-chain interoperability, enabling secure credit transfers between different national markets through a relayer system, ensuring global compatibility and preventing double-spending.

B. Technology Stack Rationale

Our technological choices were based on the identification of the model's requirements that fit the state-of-the-art solutions' use cases. After some evaluations, we selected Hyperledger Fabric as a blockchain, Hyperledger Cacti as an interoperability tool, and MCC as TEE technology.

Hyperledger Fabric [19] is a permissioned blockchain with consortium governance, X.509-based access control, private data collections for confidentiality, pseudonymous identities for buyers, modular architecture with customizable consensus, and rich smart contract support in Go. Hyperledger Cacti [17] provides interoperability across Fabric, Ethereum, and Corda, supports HTLC-based cross-chain settlements [14], and offers examples in Go, JavaScript, and Solidity—enabling government-to-government credit trading. Microsoft Confidential Containers (MCC) [20] leverage AMD Secure Encrypted Virtualization (SEV)-Secure Nested Paging (SNP) as a TEE, run containerized applications with hardware attestation, and integrate easily with Fabric chaincode through documented Golang support.

III. Core Implementation

Our implementation [21] counts on two chaincodes and an off-chain MCC service for running auctions with integrity and privacy. The *Carbon* chaincode manages the carbon credit lifecycle, performing operations such as minting, burning, bid creation, sensor data publication, vegetation properties publication, setting policies, TEE configuration, and auction result processing. The *Interop* chaincode handles cross-chain transactions and allows relays to have access to specific functions for locking and unlocking credits to enable HTLC-based atomic swaps. At last, an MCC service expects auction data (including public and private data) to run the auction algorithm privately, returning results with hardware attestation proof. Once ready, the results are submitted to the *Carbon* chaincode for verification against the data commitments and the AMD SEV-SNP certificate chain.

Because the goal is providing public transparency while preserving necessary privacy, we segment the visibility of data types based on the stakeholder's role defined in their X.509 certificate attributes. Function invoking authorization follows the same logic. Furthermore, we enable a segmented per-organization private data storage in different private data collections [16]. Buyers' (companies') activities are protected via Idemix pseudonyms [16], with data access organized into three tiers: Tier 1 (Public State) covers minting, burning, and credits; Tier 2 (Private Collections) stores financial attributes, multipliers (potentially revealing geolocation), and pseudonym–identity mappings; and Tier 3 (TEE Computation) executes auctions with hardware attestation, ensuring confidentiality and integrity.

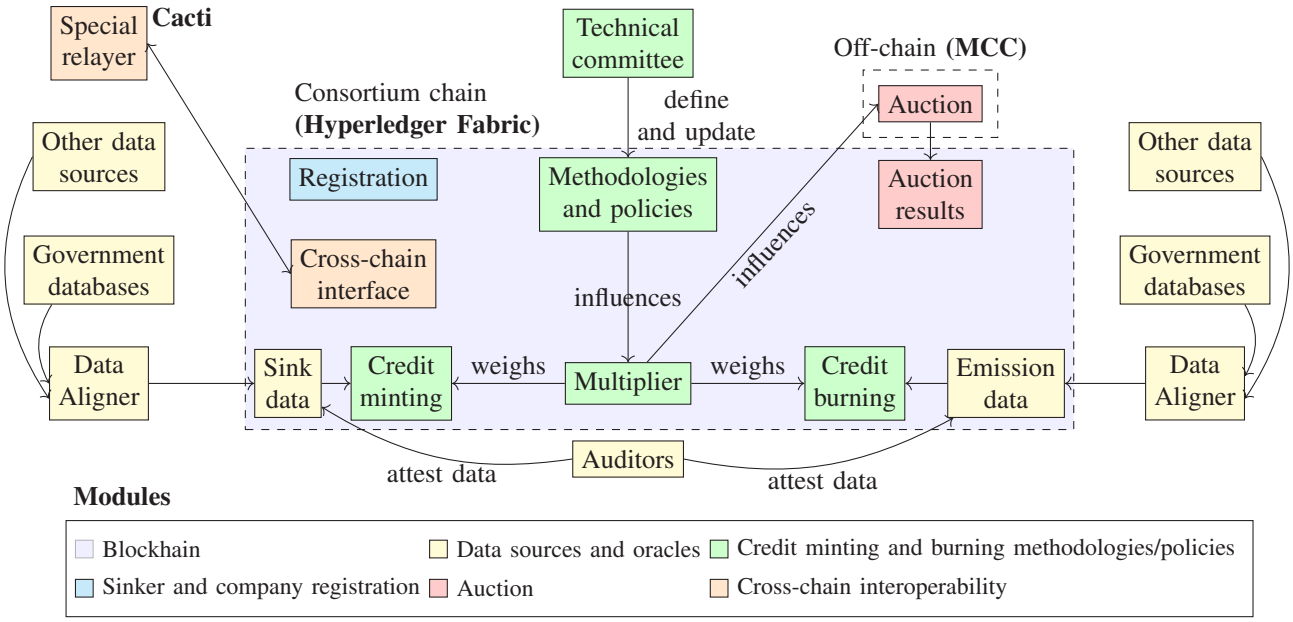

Fig. 1. High-level representation of our proposed model.

A. MCC as TEE for Auctions

MCC are containerized applications that interface with the AMD hardware through the */dev/sev-guest* Unix device. Beyond enabling integrity, MCC make their memory pages inaccessible to the hypervisor. The examples provided by the Microsoft library contain two relevant pieces: the *FetchAttestationReportByte* method and the *SNPAttestationReport* struct [20]. The first, when called inside the container, generates a signed serialized *SNPAttestationReport* including a 64-byte arbitrary argument called *ReportData*. Another important field of the *SNPAttestationReport* is the *HostData*, which enables verification of the Docker image running inside the container.

Before launching the MCC, a policy must be generated using the Azure *confcom* extension [22]. This policy contains permission and integrity configurations, including the expected container file system layers' hashes. The AMD hardware reads this policy at launch time and enforces it. For verification, the *HostData* field in the attestation report contains a hash of the provided policy. This way, the chaincode can verify that the expected container is running in a genuine AMD SEV-SNP environment. All this setup can be understood by reading and executing the Makefile in the *tee_auction* folder in our repository [21].

As the first operation, the MCC service initializes itself by generating a self-signed certificate and private key to establish a secure identity for receiving Hypertext Transfer Protocol Secure (HTTPS) requests. It then generates a hardware attestation report, embedding a SHA-512 hash of the certificate into the report's user data field. Finally, it launches an HTTPS server using these credentials, enabling clients to securely connect.

Two HTTPS routes represent how the MCC behaves. The "/report" GET route provides clients with a deserialized AMD SEV-SNP attestation report generated at startup. The "/auction" POST route is the main entry point for running a confidential auction. It expects a *SerializedAuctionData* payload originated from the *Carbon* chaincode. This function executes the core auction logic securely within the TEE and returns a *SerializedAuctionResultTEE*, splitting the public and private results. A hash over the results is calculated (1). This hash is signed by the AMD SEV-SNP hardware and signed with the private key generated at startup, ensuring both integrity and authenticity of the auction results.

$$\text{Hash} = \text{SHA-512}(\text{ResultBytes}||\text{ReceivedDataHash}) \quad (1)$$

B. Carbon Chaincode

Our chaincode design contains separate packages that implement different parts of the BETS. For instance, our *Carbon* chaincode contains packages for the auction, bids, companies, credits, data, identities, payment, policies, properties, state, tee, and vegetation. Each package manages whether the visibility of its data types and functions is public or private.

1) State and Data Types: Fabric provides a key-value state database for storing data. For interacting with it, we implement a common package called *state* containing functions for saving, loading, and deleting. For instance, *PutStateWithCompositeKey* saves an object publicly with a composite key, while *PutPvtDataWithCompositeKey* saves an object in a private data collection.

The data structs must implement the *WorldStateManager* interface, which requires methods for saving to and loading from the world state. In the method implementations, the data structure controls which fields are saved publicly or privately.

Other aspects are also present in the state package, such as querying multiple objects by range and caching queries. For proper range queries using composite keys, we adapted the

Hyperledger Fabric Chaincode package in a forked repository [23].

2) Identity and Access Control: Fabric chaincodes provide a component to extract the invoker's identity from the transaction context. This enables asserting attributes in the X.509 certificate or Idemix credential through the function *AssertAttributeValue* [23]. With this information, we can enforce access control policies in the chaincode functions, limiting who can invoke them. For instance, only members with the "PriceViewer" attribute can see bid prices.

However, Fabric does not provide Idemix identity extraction. Thus, we implement a new functionality in the *identity* package to return a unique identifier from the pseudonym elliptic curve points. With this, we can map private attributes to the pseudonym, such as the company location, to enforce policies during the auction.

The Fabric Gateway module allows connections from external applications to the blockchain, but does not support Idemix identities [24]. To overcome this, we forked its repository and added support for Idemix. Furthermore, we adapted it, enabling applications to generate as many pseudonyms as needed, avoiding identification by the transaction amount or frequency [24].

3) TEE Integration: Once the *Carbon* chaincode is initialized, it allows the calling of the *PublishExpectedTEECCEPolicy* and *PublishInitialTEEReport* functions by an authorized identity. The first function saves the expected AMD SEV-SNP base64-encoded policy in the world state. With this, parties can verify that the policy enforces that the expected Docker image will run in the MCC. After that, the second function can be called to save the initial attestation report containing the hash of the self-signed certificate generated by the MCC. The *Carbon* chaincode verifies that the policy hash matches the one expected and that the signature is valid against the AMD certificate chain.

Once the TEE is properly set up, auctions can be executed. Periodically, a private data viewer party gathers the data needed for the auction and creates a commitment using the *CommitDataForTEEAuction* function. This commitment is verified by the TEE during auction execution and takes part in the final auction result, which is signed by the AMD SEV-SNP hardware and the MCC's generated private key. For receiving the results, the chaincode expects a call on the *PublishTEEAuctionResults* function, which verifies the attestation report, the signatures, and performs the necessary state updates. The result's private part comes from the transaction's transient map, while the public part is saved in the world state.

4) Auction: The auction package rules the auction process. It allows independent and coupled auctions. For more granular policies, the coupled one is preferred at the cost of complexity and more privacy concerns. Both types of auctions deal with bids defined in the *bids* package.

Coupled auctions require properties from both buyers and sellers, joined in a single struct *PolicyInput* [21]. To relate a buy bid to a *PolicyInput*, it is necessary to map the buyer's pseudonym to its real location, represented in the struct *Company*, in order to calculate the *multiplier*. Bids with higher multipliers are satisfied first because they represent higher carbon sinking efficiency. Since someone might infer the buyer's location from the multiplier and the seller's property, the multiplier is saved in a private data collection. The credits originated from the multiplier are only revealed after what we call the *privacy delay*, consisting of an interval to reduce the competitors' gain from knowing strategic information.

Independent auctions execute a simple clearing through a double auction mechanism. They do not require properties from buyers or sellers because the credits are fungible.

5) Credits Minting and Burning: Aside from the coupled policies mentioned in the auction Section III-B4, policies apply at the minting and burning of credits. Instead of relying on the relationship between buyer and sellers, these independent policies weigh the credits based on the reliability of the data, vegetation type, climate factors, credit age, and other aspects.

For privacy concerns, the company reveals its real identity after the privacy delay, proving its carbon offsetting by showing burning operations. An Idemix credential empowers the holder to disclose its desired attributes, including the enrollment ID, which usually stays hidden for common transactions but can be exposed to reveal the real identity [25].

6) Data And Policies: Policies rely on data to make decisions and represent a crucial part of the system. The data package handles the fetching, validation, and storage of data for the policies to be applied. On-chain and off-chain data compose the possible sources in our implementation. Diverse properties might impact the multiplier differently, such as the data measurement method, the data age, the data source, and agreement among multiple sources. In our implementation, the *DataFetcher* interface enables the *policies* package to fetch the necessary data for minting and burning policies.

7) Locking credits for cross-chain transactions: To enable cross-chain transactions, the *credits* package provides a function to lock the credits with a specified destination chain ID. After locking, the *Interop* chaincode can verify the lock status and proceed with the cross-chain transaction.

C. Interop Chaincode

Following examples from the Hyperledger Cacti project [17], our *Interop* chaincode limits itself to the operations regarding verifying locked credits, creating HTLCs for cross-chain transfers through cryptocurrency, and enabling settlements with trusted international payment companies. Currently, we do not directly use the Hyperledger Cacti source code. However, we follow the examples provided in its repository [17]. Similarly to them, we use the X.509 certificate attributes to identify and give the proper permissions to the relayers.

In a typical scenario, a buyer from another chain *A* indicates the intention to buy credits from chain *B*, and they negotiate the details off-chain. On chain *B*, the seller locks the credits using the *LockCredit* function in the *Carbon* chaincode. Using the relayer, the buyer can verify that the credits are locked and

check the HTLC parameters on chain *B* to match the payment HTLC on a public chain.

Alternatively, instead of using cryptocurrency, parties might employ a trusted international payment company that can trigger the release of locked credits upon receiving payment through traditional means. In this case, the payment company would have the necessary permissions to confirm the payment and trigger the relayer to release the credits on chain *B*.

D. Validation

To validate the implementation, we performed functional tests and deployments covering chaincode unit and integration tests for world state operations, bids, auctions (on-/off-chain, independent/coupled), and TEE attestation verification. Interop chaincode tests simulated cross-chain atomic swaps. Idemix identities were tested on a Hyperledger Fabric Test Network, confirming transactions through the modified Fabric Gateway. MCC integration was validated by deploying auction containers on Azure, checking HTTPS responses, and verifying chaincode functions with MCC-generated reports, confirming expected auction behavior end-to-end.

IV. Conclusion

Previous work identified limitations and research challenges in BETS, regarding privacy, interoperability, empirical evaluation, data quality, and alignment with regulatory frameworks [5, 7, 8, 10]. Based on a theoretical model, our implementation addresses many of these gaps by focusing on the empirical realization, the privacy features, and the interoperability aspects. To accomplish it, we selected a robust technology stack, including MCC, Hyperledger Fabric, and Hyperledger Cacti, and designed a modular architecture with two chaincodes and one MCC service.

To ensure privacy, we take advantage of the private data collections on Fabric, enabling access control of sensitive data only to necessary parties. Furthermore, to protect the private data while keeping the integrity and the verifiability of the auction process, we integrated MCC using AMD SEV-SNP as TEE, which provides hardware attestation over computations. Overall, we properly balance transparency and privacy, allowing the public to check that carbon emissions are being offset while protecting sensitive business information.

As for the interoperability, our initial design and implementation followed the examples available in the Hyperledger Cacti repository [17]. However, we did not explicitly import any package from Cacti, but our interoperability functions are based on its examples. These include locking and HTLC mechanisms. Another interoperability aspect that we foresee is the integration with governmental databases, such as SICAR (Brazil) and LPIS (Europe) [18].

Joining all these qualities, our implementation eases the adoption of BETS in a regulated environment. It provides transparency to the public (in the face of the claimed corruption limitations), privacy measures to protect business information (General Data Protection Regulation (GDPR) compliance), interoperability to connect with existing regulated systems and other countries' BETS. Additionally, due to Fabric working with X.509 certificates, governments can reuse their existing Public Key Infrastructure (PKI) infrastructure to issue identities to the network participants.

For now, we focused on testing the core functionalities of our implementation, like the TEE operations and auction process. However, many other aspects can be further explored and improved, including the actual integration with governmental databases; defining the organizations (parties) control over private data collections; using geolocation to shard subchains by region to improve scalability; developing voting mechanisms to allow proper stakeholders to decide policies; representing auction data in a Merkle tree to enable proof of inclusion for auction participants; and using Geohash or R-Trees to fetch georeferenced sensor data efficiently [26].

Other improvements, less related to our implementation but still relevant for the blockchain tools ecosystem, involve the Hyperledger Fabric ecosystem. Hyperledger Fabric Private Chaincode is a project that executes a chaincode inside the Intel SGX TEE, which operates in a process-based enclave model [27]. In contrast, AMD SEV-SNP works with a virtual machine-based enclave model, offering lower costs and higher practicality, requiring no application modifications, though with weaker isolation [28]. Extending Fabric Private Chaincode to support SEV-SNP would simplify and reduce the cost of running confidential chaincodes, benefiting the community. At last, the modifications on the Fabric Gateway to support Idemix identities could be offered as a pull request to the official repository, contributing to the community and enabling more privacy-preserving applications.

References

[1] United Nations, "Kyoto protocol to the united nations framework convention on climate change," Full Text, 1998. [Online]. Available: https://unfccc.int/resource/docs/convkp/kpeng.pdf

[2] U. Nations, "Paris agreement to the united nations framework convention on climate change," 2015. [Online]. Available: https://treaties.un.org/Pages/showDetails.aspx?objid=0800000280458f37&clang=_en

[3] L. Franke, M. Schletz, and S. Salomo, "Designing a blockchain model for the paris agreement's carbon market mechanism," *Sustainability*, vol. 12, no. 3, 2020. [Online]. Available: https://www.mdpi.com/2071-1050/12/3/1068

[4] G. Marchant, Z. Cooper, and V. Gough-Stone, P.J., "Bringing technological transparency to tenebrous markets: The case for using blockchain to validate carbon credit trading markets," *Natural Resources Journal*, vol. 62, no. 2, pp. 159–181, 2022. [Online]. Available: https://digitalrepository.unm.edu/nrj/vol62/iss2/2/

[5] A. L. Merlo, D. S. Mendonça, J. Santos, S. T. Carvalho, R. Guerra, and D. Brandão, "Blockchain for the carbon market: a literature review," *Discover Environment*, vol. 3, no. 1, pp. –, 2025. [Online]. Available: https://doi.org/10.1007/s44274-025-00260-4

[6] T. P. Abiodun, N. I. Nwulu, and P. O. Olukanmi, "Application of blockchain technology in carbon trading market: A systematic review," *IEEE Access*, vol. 13, pp. 5446 – 5470, 2025, cited by: 3; All Open Access; Gold Open Access. [Online]. Available: https://doi.org/10.1109/ACCESS.2024.3523672

[7] N. Adhikari, H. Li, and B. Gopalakrishnan, "A bibliometric and systematic review of carbon footprint tracking in cross-sector industries: Emerging tools and technologies," *Sustainability (Switzerland)*, vol. 17, no. 9, 2025, cited by: 0; All Open Access; Gold Open Access. [Online]. Available: https://doi.org/10.3390/su17094205

[8] B. Divya and H.-s. Byun, "Future-proofing co2 mitigation towards a circular economy: A systematic review on process integration and advanced tools," *Environmental Science and Ecotechnology*, vol. 26, 2025, cited by: 0. [Online]. Available: https://doi.org/10.1016/j.ese.2025.100587

[9] A. Vilkov and G. Tian, "Blockchain's scope and purpose in carbon markets: A systematic literature review," *Sustainability (Switzerland)*, vol. 15, no. 11, 2023, cited by: 29; All Open Access; Gold Open Access. [Online]. Available: https://doi.org/10.3390/su15118495

[10] C. Mulligan, S. Morsfield, and E. Cheikosman, "Blockchain for sustainability: A systematic literature review for policy impact," *Telecommunications Policy*, vol. 48, no. 2, 2024, cited by: 4; All Open Access, Hybrid Gold Open Access. [Online]. Available: https://doi.org/10.1016/j.telpol.2023.102676

[11] Brasil, "Lei n. 15042/2024," Brasília, 12 2024. [Online]. Available: https://www.planalto.gov.br/ccivil_03/_ato2023-2026/2024/lei/L15042.htm

[12] Z. Guo, H. Pan, A. He, Y. Dai, X. Huang, X. Si, C. Yuen, and Y. Zhang, "Trusted Execution Environments for Blockchain: Toward Robust, Private, and Scalable Distributed Ledgers," *IEEE Internet of Things Journal*, vol. 12, no. 18, pp. 38 736–38 754, 2025. [Online]. Available: https://doi.org/10.1109/JIOT.2025.3587023

[13] S. K. Ezzat, Y. N. Saleh, and A. A. Abdel-Hamid, "Blockchain Oracles: State-of-the-Art and Research Directions," *IEEE Access*, vol. 10, no. None, pp. 67 551–67 572, 2022. [Online]. Available: https://ieeexplore.ieee.org/document/9801856

[14] S. D. Kotey, E. T. Tchao, A.-R. Ahmed, A. S. Agbemenu, H. Nunoo-Mensah, A. Sikora, D. Welte, and E. Keelson, "Blockchain interoperability: the state of heterogenous blockchain-to-blockchain communication," *IET Communications*, vol. 17, no. 8, pp. 891–914, 2023. [Online]. Available: https://ietresearch.onlinelibrary.wiley.com/doi/abs/10.1049/cmu2.12594

[15] M. A. Johnson, S. Volos, K. Gordon, S. T. Allen, C. M. Wintersteiger, S. Clebsch, J. Starks, and M. Costa, "Confidential Container Groups," *Communications of the ACM*, vol. 67, no. 10, pp. 40–49, 2024. [Online]. Available: https://doi.org/10.1145/3686261

[16] Hyperledger, "A blockchain platform for the enterprise hyperledger fabric," 2023, online; accessed March-2024. [Online]. Available: https://hyperledger-fabric.readthedocs.io/en/release-2.5/

[17] Contributors to Hyperledger Cacti, "Hyperledger cacti," 2024, online; accessed May-2024. [Online]. Available: https://github.com/hyperledger/cacti

[18] S. Jung, L. V. Rasmussen, C. Watkins, P. Newton, and A. Agrawal, "Brazil's national environmental registry of rural properties: Implications for livelihoods," *Ecological Economics*, vol. 136, pp. 53–61, 2017. [Online]. Available: https://doi.org/10.1016/j.ecolecon.2017.02.004

[19] Contributors to Hyperledger Fabric, "Hyperleger fabric," WebPage, 2025. [Online]. Available: https://github.com/hyperledger/fabric

[20] Microsoft, "Confidential sidecar containers," WebPage, 2025. [Online]. Available: https://github.com/microsoft/confidential-sidecar-containers/tree/4814b442cf71de2b1317f00846f16727e40a3088

[21] J. Westphall, "A country-agnostic blockchain ets model with geographical and time references inspired by the brazilian ecosystem using hyperledger fabric, hyperledger cacti and microsoft confidential containers," WebPage, 2025. [Online]. Available: https://github.com/johannww/phd-impl

[22] Microsoft, "az confcom - Azure CLI," https://learn.microsoft.com/en-us/cli/azure/confcom?view=azure-cli-latest, Jul 2025, accessed: 2025-09-19.

[23] J. Westphall, "Hyperledger fabric packages for go chaincode fork," WebPage, 2025. [Online]. Available: https://github.com/johannww/fabric-chaincode-go

[24] ——, "Hyperledger fabric gateway fork," WebPage, 2025. [Online]. Available: https://github.com/johannww/fabric-gateway/

[25] IBM, "Idemix," 2025, online; accessed Sep-2025. [Online]. Available: https://github.com/IBM/idemix

[26] L. Sun and B. Jin, "Improving NoSQL Spatial-Query Processing with Server-Side In-Memory R*-Tree Indexes for Spatial Vector Data," *Sustainability (Switzerland)*, vol. 15, no. 3, pp. –, 2023. [Online]. Available: https://www.mdpi.com/2071-1050/15/3/2442

[27] C. C. Lew, C. F. Torres, S. Shinde, and M. Brandenburger, "Revisiting Rollbacks on Smart Contracts in TEE-protected Private Blockchains," *Proceedings - 9th IEEE European Symposium on Security and Privacy Workshops, Euro S and PW 2024*, pp. 217–224, 2024. [Online]. Available: https://doi.org/10.1109/EuroSPW61312.2024.00029

[28] L. Coppolino, S. D'Antonio, G. Mazzeo, and L. Romano, "An experimental evaluation of TEE technology: Benchmarking transparent approaches based on SGX, SEV, and TDX," *Computers and Security*, vol. 154, no. None, pp. –, 2025. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0167404825001464