
Received: 2024-09-09 Revised: 2025-10-19 Accepted: 2025-11-01 Published: : 2026-02-15

WebAssembly (Wasm) and Blockchain Technology Provide a Feasible Solution for Dependable and High-performance Frontend Systems

Horbenko Rollen

National Transport University, Ukraine

ABSTRACT

WebAssembly (Wasm) with blockchain technology provides a feasible alternative for dependable and high-performance frontend systems. Wasm delivers rapid execution rates by integrating code from high-level languages to enhance performance constraints. Its sandboxed execution approach improves security by mitigating memory-related vulnerabilities. Likewise, blockchain enhances security with decentralized, tamper-proof data structures and smart contracts. Traditional blockchain frameworks frequently have computational inefficiencies; nevertheless, Wasm-based execution platforms such as Polkadot and EOS enhance resource utilization and augment interoperability. This integration enables rapid, dependable interactions in decentralized applications (dApps). Potential advantages encompass rapid and safe off-chain computations, hence alleviating blockchain congestion in front-end frameworks. Nonetheless, obstacles persist in ensuring the security of Wasm execution within decentralized systems and enhancing the interoperability between blockchain and Wasm. A possible approach is to utilize Just-In-Time (JIT) and Ahead-of-Time (AOT) compilation methods in conjunction with zero-knowledge proofs to improve efficiency and security attributes. By integrating the efficiency of WebAssembly with the security of blockchain, scalable and decentralized front-end systems are developing to address demanding online scenarios.

Keywords: Blockchain, Cryptography, WebAssembly.

INTRODUCTION

WebAssembly is a binary instruction format engineered for secure and efficient execution across diverse platforms. Its linguistically prophetic nature permits developers to compile code from languages such as C, C++, and Rust into a compact binary format, facilitating competitive performance for web browser applications. Blockchain technology is defined by decentralized and immutable ledgers that promote data integrity, transparency, and security across various applications. Wasm significantly enhances blockchain performance through the implementation of smart contracts in high-level languages. The current emphasis in blockchain technology is increasingly on security considerations. To address these vulnerabilities, other studies concentrated on memory attacks and related issues. A thorough examination examined 121 articles on WebAssembly security, classifying them into seven specific categories (Waseem et al., 2024). This study emphasized the capacity of Wasm to improve system security while also acknowledging the dangers of its exploitation for malevolent vulnerabilities. Advanced integration of WebAssembly and blockchain technology for front-end systems is documented in the literature. Certain academics investigated the implementation of Wasm to develop safe and decentralized application architectures. Utilizing Wasm's ability to promote tamper-resistant structures. Furthermore, progress was achieved in enhancing Wasm's functionality to directly engage with hardware interfaces such as I2C and USB for IoT-based embedded systems (Kaluva & Hossain, 2020). This resulted in better security and efficient interactions between blockchain apps and hardware components for front-end systems. A research gap exists in fully harnessing the promise of merging WebAssembly and blockchain for secure frontend systems to tackle challenges such as scalability, energy consumption, and integration complexities.

Efficient execution environments are required to analyze substantial data volumes in intricate machine learning for real-time analytics. Traditional web technologies such as JavaScript experience performance limitations in complex computational environments. Wasm, owing to its tiny binary structure, enables efficient execution of code written in languages such as C, C++, and Rust in both web browsers and server-side environments. Wasm offers speeds comparable to native programs due to its efficient compilation method. Wasm

Received: 2024-09-09 Revised: 2025-10-19 Accepted: 2025-11-01 Published: : 2026-02-15

enables applications to operate uniformly across several platforms without modification and ensures effective memory management with linear memory to prevent unnecessary latencies.

Table 1. A performance inter-comparison between Javascript and Wasm

Feature	Javascript	Wasm
Execution Speed	Moderate	High
Memory Efficiency	Low	High
Multi- Threading Support	Limited	Experimental
Portability	High	High
Security	Moderate	High
Deterministics Execution	Low	High
Binary size	Large	Compact

RESEARCH ELABORATIONS

Wasm utilizes a manual memory scheme wherein values are stored in densely packed array forms to enhance cache locality. The computational expense is additionally diminished by storing boolean values as distinct bytes. Similarly, the utilization of hash tables in Wasm enhances memory optimization, implemented via memory buffers and open addressing techniques (Zhang et al., 2023). An additional sophisticated data structure utilized for efficient memory management in Wasm is the arena allocator. An arena allocator preallocates a substantial memory block and oversees object lives within that block, instead of dynamically creating and deallocating individual objects. This reduces allocation overhead and accelerates memory allocation and deallocation processes. This method is optimal for frequent allocation and deallocation scenarios, such as game engines and WebAssembly virtual machines. Compression-oriented data structures, such as sparse matrices and compressed trees, mitigate unnecessary storage in applications including machine learning, graph processing, and scientific computing (Baradaran et al., 2024). Run-length encoding, Huffman encoding, and delta compression facilitate the computation of extensive datasets with reduced space requirements. Moreover, the multi-threading capability facilitates the concurrent execution of

Received: 2024-09-09 Revised: 2025-10-19 Accepted: 2025-11-01 Published: : 2026-02-15

data streams across many CPUs. The shared array buffering enables several instances to allocate memory efficiently and utilize lock-free techniques. Consequently, selecting the appropriate data format is contingent upon the particular application needs and access patterns.

Single Instruction, Multiple Data (SIMD) is employed to execute a singular instruction across numerous data pieces concurrently. SIMD utilizes data-level parallelism, allowing Wasm systems to perform vectorized computations effectively, particularly in multimedia processing, cryptography, and machine learning applications (Ketonen, 2022). SIMD was utilized via a collection of vector instructions that function on 128-bit registers, partitioned into various smaller data lanes, including 16 lanes of 8-bit integers, 8 lanes of 16-bit integers, 4 lanes of 32-bit integers, or 2 lanes of 64-bit integers. The vectorized operations enable Wasm modules to execute arithmetic, logical, and bitwise operations on many values concurrently, hence enhancing memory throughput. SIMD in Wasm is executed via platform-independent vector instructions that translate the fundamental SIMD instructions into contemporary processors. This allows Wasm to fully exploit the capabilities of the host system without necessitating platform-specific code optimizations. The Wasm embedded SIMD framework offers a comprehensive array of instructions for data shuffling and permutation tasks. The use of SIMD in Wasm is fostering real-time applications addressing significant data challenges, including gaming, simulations, and AI applications.

AOT compilation entails converting Wasm bytecode into native machine code prior to execution, rather than during runtime. This precompilation reduces startup latency for sophisticated compiler optimizations by enabling the compiler to execute advanced techniques, like function inlining, loop unrolling, and register allocation, which are restricted to JIT compilers. This is relevant for embedded devices, server-side applications, and real-time applications that necessitate immediate responsiveness. As AOT-generated code is precompiled and validated, it diminishes the attack surface linked to JIT compilation, including JIT spraying attacks, hence assuring more secure execution environments. AOT compilers are designed for certain CPU architectures and instruction sets, generating highly efficient binaries customized for the underlying hardware. Optimizations for speculative execution to enhance cross-platform compatibility.

Received: 2024-09-09 Revised: 2025-10-19 Accepted: 2025-11-01 Published: : 2026-02-15

Security considerations encompass memory limitations to prevent buffer overflows and minimize attack vectors. It is essential to ensure that only authenticated modules are executed, as WebAssembly lacks direct access to system APIs. However, WebAssembly can utilize WebCrypto APIs for secure encryption and decryption of data streams. The debugging process and trash collection system in WebAssembly code are more intricate than in JavaScript due to its binary structure. Wasm must engage with JavaScript to utilize imperative browser-specific functionalities, and the standardization of multi-threading support remains notably difficult. Consequently, enhanced multi-threading capabilities, integration with WebGPU, zero-copy data sharing, and expanded language support represent significant research avenues for future endeavors.

RESULTS AND DISCUSSION

Amid escalating cyber threats, data breaches, and privacy infringements, the necessity for data protection has intensified. Conventional centralized storage systems exhibit a singular point of failure, rendering them vulnerable to attacks; however, blockchain has surfaced as a viable security solution by utilizing cryptographic hashing, decentralization, and consensus methods to guarantee data integrity. The blockchain is a decentralized ledger system, consisting of a continuously evolving sequence of blocks that hold transaction records. Each block is cryptographically linked to its predecessor, creating an immutable chain that precludes unauthorized alterations. The ledger is meticulously maintained throughout a network of nodes, with each node retaining a complete copy of the blockchain. This facilitates redundancy and fortitude against nefarious assaults. A method known as cryptographic hashing is employed for data security and integrity, with each block possessing a distinct cryptographic hash function produced by SHA-256 algorithms. Any alteration to a block would modify its hash and disrupt the chain, signifying fraudulent activity. Smart contracts facilitate blockchain capabilities by establishing automated and self-executing agreements. These programmable scripts perform specified conditions upon activation and ensure transparency in digital transactions.

The implementation of smart contracts reduces operational overhead automatically. Decentralization is a fundamental characteristic of blockchain, wherein data is disseminated among several parties rather than being subject to centralized authority. In a decentralized network, transactions and data are collectively vetted by nodes, enhancing the system's resilience to cyberattacks. The tokenization function facilitates the digital representation of assets on the blockchain. In this view, Bitcoin and Ethereum are intrinsic tokens of their own blockchains. The interoperability of any blockchain emphasizes easy communication between various blockchain networks. In this context, interoperability solutions like Polkadot and Cosmos provide interaction among many blockchains, thereby realizing the full potential inherent in blockchain technology. Scalability in blockchains is attained through sharding, off-chain processing, and layer-two protocols. Sharding divides the blockchain into smaller, manageable portions, facilitating the concurrent processing of many transactions. Likewise, off-chain solutions alleviate the computational load by processing transactions externally while maintaining data integrity. The governance structures in blockchain networks dictate decision-making processes and the implementation of protocol upgrades.

Consensus techniques guarantee that blockchain distributed networks maintain a singular version of the truth without dependence on centralized authority. These techniques provide dependable transactions to uphold the integrity of the blockchain by authenticating fresh blocks to the ledger. A summary of common items is provided below. Proof of Work (PoW) is the inaugural consensus mechanism developed by Bitcoin and is being extensively utilized across other blockchain networks. In Proof of Work (PoW), participants, known as miners, resolve intricate cryptographic challenges to authenticate transactions and incorporate new blocks into the blockchain through data mining. Miners vie to discover a hash value that satisfies a certain difficulty threshold by brute-force computation. The initial miner to resolve the puzzle disseminates the new block to the network for validation. Additional nodes authenticate the block and refresh their version of the blockchain. The successful miner obtains a block reward and transaction fees. Although Proof of Work (PoW) offers adequate security and decentralization, it is energy-intensive.

Proof of Stake (PoS) emerged as an energy-efficient substitute for Proof of Work (PoW), wherein validators confirm new blocks based on the quantity of tokens they possess, rather than solving computational puzzles. Validators are selected in a deterministic manner

Received: 2024-09-09 Revised: 2025-10-19 Accepted: 2025-11-01 Published: : 2026-02-15

according to their stake size and by randomization. The designated validator proposes the subsequent block and disseminates it to the network, where other validators authenticate the block. Upon reaching consensus, the information is included in the blockchain, and validators are compensated with rewards instead of newly minted currencies. Owing to its enhanced transaction throughput, it is utilized in networks like Ethereum 2.0, Cardano, and Polkadot. Nonetheless, it is susceptible to centralization problems due to its bias favoring rich users, who possess a greater likelihood of being selected to validate blocks. Delegated Proof of Stake (DPoS) is a version of Proof of Stake (PoS) wherein token holders utilize a vote method to elect delegates responsible for validating transactions. The elected delegates alternate in validating transactions and provide incentives to the people that supported them. Delegated Proof of Stake (DPoS) improves scalability; yet, its dependence on a limited number of validators may result in centralization vulnerabilities.

Wasm addresses various applications, including decentralized finance, fraud detection, and secure payment processing. Blockchain networks are utilizing its ability to build and implement smart contracts with greater efficiency. Wasm-based smart contracts exhibit enhanced flexibility and scalability, rendering DeFi platforms more responsive to user requests. In addition to trade and decentralized applications, Wasm-driven fraud detection algorithms are safeguarding financial transactions, benefiting key payment providers such as PayPal. Moreover, Wasm is streamlining cross-border payment processes by diminishing dependence on numerous intermediaries for expedited transactions. Wasm has become a potent instrument for medical imaging and diagnosis. In comparison to baseline models, Wasm applications optimize time by conducting health-related tasks directly in a web browser with superior efficiency. An exemplary case is the Open Health Imaging Foundation (OHIF), which has included Wasm in its web-based imaging viewer. This enables radiologists to access and interpret medical images immediately without severe software prerequisites. Telemedicine facilitates seamless, high-quality video consultations between physicians and patients while safeguarding sensitive medical information. In genomics and personalized medicine, Wasm accelerates DNA sequencing and other intricate bioinformatics procedures. In addition to specific applications, Wasm is facilitating enhanced communication and data exchange throughout the global healthcare ecosystem.

Received: 2024-09-09 Revised: 2025-10-19 Accepted: 2025-11-01 Published: : 2026-02-15

Blockchain is transforming banking technology by accelerating transactions and reducing expenses. This is prompting corporations such as MoneyGram and Visa to begin incorporating blockchain into their systems. Platforms such as Uniswap enable users to exchange cryptocurrencies directly from their wallets, representing a progression towards more financial autonomy. Owing to the intrinsic security attributes of blockchain, Visa's B2B Connect facilitates businesses in managing international transactions more securely and on a larger scale. Insurance businesses are increasingly utilizing blockchain-based smart contracts. These digital contracts provide immediate disbursements contingent upon established criteria. Similarly, blockchain possesses significant potential for regulatory compliance frameworks and anti-money laundering tactics.

Utilizing a decentralized approach, healthcare providers can securely encrypt and store medical data with greater autonomy through blockchain networks. Examples include companies such as MediBloc and Medicalchain, which empower people with greater ownership over their health information and facilitate the secure sharing of records with experts. Blockchain networks facilitate the detection of counterfeit pharmaceuticals, and prominent technology companies are developing innovative methods to guarantee the authenticity of prescriptions. Likewise, clinical trials benefit from blockchain's capacity to provide a tamper-resistant system for the storage of research data. Blockchain-based solutions have demonstrated their efficacy in combating fraudulent claims and have resulted in savings of billions of dollars by reducing administrative costs. Furthermore, blockchain networks are facilitating the secure exchange of data between patients and physicians in remote environments.

CONCLUSION

The integration of Wasm and blockchain transforms the FinTech and healthcare sectors by establishing novel opportunities for security, efficiency, and interoperability. The banking sector is transformed by blockchain, which facilitates frictionless and secure cross-border payments, while Wasm improves computational efficiency. In healthcare, blockchain enhances

Received: 2024-09-09 Revised: 2025-10-19 Accepted: 2025-11-01 Published: : 2026-02-15

the security and transparency of telemedicine while preserving the integrity of patient records and conversations. The future of FinTech and healthcare will hinge on the rational integration of blockchain's dependability with Wasm's high-performance attributes.

REFERENCES

- Baradaran, S., Huang, L., Raghothaman, M., & Wang, W. (2024). Reusing Legacy Code in WebAssembly: Key Challenges of Cross-Compilation and Code Semantics Preservation. *ArXiv Preprint ArXiv:2412.20258*.
- Kaluva, L. V. S., & Hossain, A. (2020). *WebAssembly for Video Analysis: An Explorative Multi-method study*.
- Ketonen, T. (2022). *Examining performance benefits of real-world WebAssembly applications: a quantitative multiple-case study*.
- Waseem, M., Das, T., Ahmad, A., Liang, P., & Mikkonen, T. (2024). Issues and their causes in WebAssembly applications: an empirical study. *Proceedings of the 28th International Conference on Evaluation and Assessment in Software Engineering*, 170–180.
- Zhang, Y., Cao, S., Wang, H., Chen, Z., Luo, X., Mu, D., Ma, Y., Huang, G., & Liu, X. (2023). Characterizing and detecting webassembly runtime bugs. *ACM Transactions on Software Engineering and Methodology*, 33(2), 1–29.