International Journal of Current Science Research and Review
ISSN: 2581-8341
Volume 08 Issue 05 May 2025
DOI: 10.47191/ijcsrr/V8-i5-36, Impact Factor: 8.048
IJCSRR @ 2025
www.ijcsrr.org

# Web Assembly and Blockchain for High-Performance Secure Front-End Systems

**Yuliia Horbenko**
Master's degree, Senior Front-End Developer at Smartbarrel
National Transport University, 1 Mykhaila Omelianovycha-Pavlenka Str., Kyiv, 01010, Ukraine
ORCID: https://orcid.org/0009-0004-5451-4530

**ABSTRACT:** Web Assembly (Wasm) and blockchain technology offer a viable solution for reliable and high-performance front-end systems. Wasm provides high execution speeds by incorporating code from high-level languages to improvise on performance limitations. Its sand-boxed execution model enhances security by extenuating memory-related weaknesses. Similarly, blockchain reinforces security with decentralized, tamper-resistant data structures and smart contracts. Conventional blockchain frameworks often suffer from computational overhead, but Wasm-based execution platforms like Polkadot and EOS optimize resource utilization and improve interoperability. This integration facilitates high-speed, reliable interactions in decentralized applications (dApps). Potential benefits include fast and secure off-chain computations, hence reducing blockchain congestion in front-end frameworks. However, challenges remain in securing Wasm execution in decentralized environments and optimizing blockchain and Wasm interoperability. A promising direction is to exploit Just-In-Time (JIT), Ahead-of-Time (AOT) compilation schemes along with zero-knowledge proofs to further enhance performance and security characteristics. By coupling Wasm's efficiency with blockchain's security, scalable and decentralized front-end systems are evolving to meet challenging web demand scenarios.

**KEYWORDS:** Blockchain, cryptography, dApps, Just in time, Web assembly, Zero knowledge proofs.

## INTRODUCTION

Web Assembly is a binary instruction format designed for safe and efficient execution across various platforms [1]. Its language-prescient nature allows developers to compile code from languages including C, C++, and Rust into a compact binary format, enabling competitive performance for web browser applications [2]. Blockchain technology is characterized by decentralized and immutable ledgers to provide enhanced data integrity, transparency, and security features across multifarious applications [3, 4]. Wasm is specifically helpful in improving the performance of blockchain in the form of smart contracts via implementations in high level languages [5- 9]. The recent focus in block-chain technology is more towards the security aspects [10-13]. To mitigate such vulnerabilities [14], different researches focused on memory attacks and associated problems [15-17].

A comprehensive review by Perrone and Romano in 2024 analyzed 121 papers on Web Assembly security, categorizing them into seven distinct areas [18]. This study highlighted both the potential of Wasm to enhance system security and the risks of its exploitation for malicious vulnerabilities. Sophisticated integration of Wasm and blockchain technologies for front-end systems are reported in the literature [19- 22]. Some researchers explored the use of Wasm to create secure and decentralized application architectures by . leveraging Wasm's capabilities to foster tamper-proof architectures [23-25]. Additionally, advancements were made in extending Wasm's capabilities to interact directly with hardware interfaces including I2C and USB for IoT based embedded systems [26, 27]. This led to more secure and efficient interactions between blockchain applications and hardware components for front-end systems. There exists a research gap in fully realizing the potential of integrating Wasm and blockchain for secure front-end systems to address issues like scalability, energy consumption, and integration complexities [28- 30].

## I. WASM FOR HIGH-PERFORMANCE DATA PROCESSING

To process large amounts of data in complex machine learning for real-time analytics requires efficient execution environments. Conventional web technologies like JavaScript suffer from performance bottlenecks in challenging computations scenarios. Wasm due to its compact binary format allows code written in languages like C, C++, and Rust to run efficiently in web browsers and server-side settings. Wasm provide speeds comparable to indigenous applications due its optimized compilation process. Wasm

International Journal of Current Science Research and Review
ISSN: 2581-8341
Volume 08 Issue 05 May 2025
DOI: 10.47191/ijcsrr/V8-i5-36, Impact Factor: 8.048
IJCSRR @ 2025
www.ijcsrr.org

allows applications to run seamlessly across different platforms without any modification and ensure efficient memory management via linear memory to avoid unneccesary latencies. Table 1 is incorporated to compare the performance characteristics of JavaScript with Wasm.

**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 | Mopderate | High |
| Deterministics Execution | Low | High |
| Binary size | Large | Compact |

***A. Efficient Data Structures for memory optimization***

Wasm exploits manual memory scheme in which values are stored in tightly packed arrays formats for improving cache locality. The computational cost is further reduced by storing boolean values in the form of separate bytes. Likewise, the use of hash tables in Wasm also contributes to memory optimization and they are employed through memory buffers and open addressing schemes. Another advanced data structure used for efficient memory management in Wasm is the arena allocator. Rather than dynamically allocating and freeing individual objects, an arena allocator preallocates a large block of memory and manages object lifetimes within that block.

This helps in decreasing the allocation overhead and speeds up memory allocation and deallocation activities. This strategy is optimum for frequent allocation and deallocation considerations like game engines and Wasm virtual machines. Compression-based data structures including sparse matrices and compressed trees help reduce redundant storage in applications such as machine learning, graph processing, and scientific computing. The run- length and Huffman encoding, and delta compression helps in computing large datasets with minimal space requirements. Furthermore, the multi-threading support allows parallel execution of data streams across different CPU's. The shared array buffering allows multiple instances to allocate memory efficiently and lock-free algorithms. Therefore choosing the right data structure depends on the specific application requirements and access patterns.

***1) SIMD (Single Instruction Multiple Data): Accelerating Vectorized Computations***

To realize a single instruction to process multiple data elements simultaneously, single instruction multiple data (SIMD) is used. SIMD leverages data level parallelism, enabling Wasm applications to execute vectorized computations efficiently and are well multimedia processing, cryptography, and machine learning applications. Wasm exploited SIMD through a set of vector instructions that operate on 128-bit registers, divided into multiple smaller data lanes, such as 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. These vectorized operations allow Wasm modules to efficiently perform arithmetic, logical, and bitwise operations on multiple values simultaneously, hence improving memory throughput. SIMD in Wasm is implemented through platform-agnostic vector instructions that map the baseline SIMD instructions into modern processors. This enables Wasm to unleash the full potential of the host system without requiring platform specific code optimizations. The Wasm embedded SIMD scheme provides an extensive set of instructions for data shuffling and permutation operations. The use of SIMD in Wasm is nurturing real-time applications big data problems including gaming, simulations and AI applications.

***2) Ahead-of-Time (AOT) Compilation***

AOT compilation involves translating Wasm bytecode into native machine code before execution rather than during runtime. This precompilation helps in mitigating startup latency for advanced compiler optimizations by allowing the compiler to perform advanced optimizations, such as function inlining, loop unrolling, and register allocation, which are limited to JIT compilations. This is pertinent for embedded systems, server-side and real-time applications that require immediate responsiveness. Since AOT-generated code is precompiled and verified, it reduces the attack surface associated with JIT compilation, including the JIT spraying attacks hence ensuring safer execution environments. AOT compilers target specific CPU architectures and instruction sets,

International Journal of Current Science Research and Review
ISSN: 2581-8341
Volume 08 Issue 05 May 2025
DOI: 10.47191/ijcsrr/V8-i5-36, Impact Factor: 8.048
IJCSRR @ 2025
www.ijcsrr.org

producing highly optimized binaries tailored to the underlying hardware, which allows. Speculative execution optimizations for cross-platform compatibility.

*3) Security Considerations & Challenges*

Security aspects involve memory bounds to restrict buffer overflows and reducing attack vectors. It must be ascertained that only verified modules are executed as Wasm does not have direct access to system APIs. But Wasm can make use of WebCrypto APIs for secure encryption and decryption data flow. On the downside the debugging process and garbage collection framework in Wasm code is more complex than JavaScript due to its binary format. Wasm needs to interact with JavaScript to access imperative browser-specific features and standardization of multi- threading support is yet quite challenging. Therefore, improved multi- threading capabilities, integration with WebGPU, zero copy data sharing and extended language support are some key research directions for future work.

## II. BLOCKCHAIN FOR SECURE DATA STORAGE AND TRANSMISSION

With rising cyber threats, data breaches, and privacy violations, data security is needed more than ever. Traditional centralized storage systems present a single point of failure, making them susceptible to attacks but blockchain has emerged as a promising security solution leveraging cryptographic hashing, decentralization, and consensus mechanisms to ensure data integrity.

### *A. Key Components of Blockchain*

The blockchain is a distributed ledger system, which is an incessantly developing chain of blocks containing dealing records. Each block is cryptographically connected to the previous one, resulting in a changeless chain that prevents illegitimate changes. The ledger is well-kept across a network of nodes, with each node storing a copy of the entire blockchain. This helps ensure redundancy and resilience against malicious attacks. A technique called cryptographic hashing is used for data security and integrity, and each block maintains a unique cryptographic hash function generated by SHA-256 algorithms. Any modification to a block would change its hash and break the chain, indicating fraudulent activity. Smart contracts enable blockchain functionality to create automated and self-executing agreements. These programmable scripts execute predefined conditions when initiated and are responsible for maintaining transparency in digital transactions. The use of smart contracts mitigates the operational overhead in an automatic fashion. Decentralization is another key feature of blockchain in which data is distributed that distributes among different participants rather than centralized control. In a decentralized network, transactions and data are validated collectively by nodes, which makes the system more resilient to cyberattacks. The tokenization function enables the representation of assets in a digital form within the blockchain. In this context, cryptocurrencies such as Bitcoin and Ethereum are native tokens of their respective blockchains. The interoperability of any blockchain focuses on seamless communication amongst different blockchain networks. In this regard, interoperability solutions such as Polkadot and Cosmos enable different blockchains to interact to unleash the true potential associated with a blockchain. Scalability is achieved in blockchains by sharding, off-chain processing, and layer-two protocols. Sharding partitions the blockchain into smaller, manageable segments, allowing parallel processing of multiple transactions. Similarly, off-chain solutions reduce the computational burden by handling transactions externally while ensuring data integrity. Further, the governance models within blockchain networks guide how decisions are made, and protocol updates are implemented.

### *B. Consensus Mechanisms for Secure Data Processing*

Consensus mechanisms ensure that blockchain distributed networks adhere a single version of the truth without relying on centralized control. These mechanisms enable reliable transactions to maintain the integrity of the blockchain by validating new blocks to the ledger. A list of common are summarized as below.

*1) Proof of Work (PoW)*

PoW is the first consensus mechanism pioneered by Bitcoin and is now widely deployed in many blockchain networks. In PoW, participants (miners) solve complex cryptographic puzzles to validate transactions and add new blocks to the blockchain via data mining. Miners compete to find a hash value that meets a predefined difficulty target using brute-force computation. The first miner to solve the puzzle broadcasts the new block to the network for verification. Other nodes validate the block and update their copy of the blockchain. The winning miner receives a block reward and transaction fees.While PoW provides reasonable security and decentralization but are energy extensive.

International Journal of Current Science Research and Review
ISSN: 2581-8341
Volume 08 Issue 05 May 2025
DOI: 10.47191/ijcsrr/V8-i5-36, Impact Factor: 8.048
IJCSRR @ 2025
www.ijcsrr.org

*2) **Proof of Stake (PoS)***

The PoS evolved as an energy-efficient alternative to PoW in which miners, instead of solving computational puzzles, validate new blocks based on the number of tokens they hold. Validators are chosen deterministically based on their stake size and randomization. The selected validator proposes the next block and publicizes it to the network, where other validators verify the block. If consensus is reached, it is added to the blockchain, and validators receive rewards rather than new coins. Due to its increased transaction throughput, it is used in networks like Ethereum 2.0, Cardano, and Polkadot. However, it suffers from centralization risks as it is biased towards affluent participants who have a higher chance of being chosen to validate blocks.

*3) **3. Delegated Proof of Stake (DPoS)***

DPoS) was pitched as a variant of PoS in which voting system is used by the token holders elect a number of delegates to validate transactions. The elected delegates take turns validating transactions and delegates share rewards with the users who voted for them. DPoS enhances scalability but its reliance on a small group of validators can lead to centralization risks.

*4) **Proof of Authority (PoA)***

In PoA a limited number of trusted nodes, known as validators, are pre- confirmed to validate transactions and create blocks. Unlike PoW and PoS, PoA is not contingent upon computational power or staking but emphasize on the credibility of validators. PoA lacks decentralization and is therefore considered a viable choice for private or consortium blockchains including VeChain and certain enterprises.

*5) **Proof of Burn (PoB)***

PoB, as its name suggests, is an alternative consensus scheme where participants burn cryptocurrency by sending it to an irretrievable address. This process demonstrates commitment to the network and allows the participant to validate transactions. The amount burned determines the participant's likelihood of being selected as the next block validator. Hence, participants must continue burning tokens to maintain influence in the network. PoB provides a balance between network security and resource efficiency.

*6) **Proof of Elapsed Time (PoET)***

PoET is designed by Intel to ensure fair leader election without extensive computational load. In PoET, each participant requests a randomly assigned waiting time from a trusted execution environment (TEE) and the participant with the shortest wait time gains the right to create the next block. It is primarily used in enterprise blockchain networks including Hyperledger Sawtooth.

*7) **Hybrid Consensus Mechanisms***

Hybrid consensus models that combine multiple mechanisms to balance security, scalability and efficiency are being evolved. For instance, Ethereum 2.0 employs PoS with sharding for scalability, while Bitcoin uses PoW with layer-2 solutions to enhance transaction throughput. Decred (DCR) is a common hybrid blockchain configuration that combines PoW and PoS. Similarly, Horizen (ZEN) uses a hybrid model to enhance network governance and security. However, hybrid consensus mechanisms face challenges in terms of architectural complexity, governance and higher development expenses.

### *B. Secure Data Transmission with Blockchain*

Ensuring reliable data transmission in blockchain is pertinent for maintaining confidentiality, integrity, and authenticity in a decentralized environment. The blockchain relies on advanced cryptographic techniques and secure communication protocols as discussed under to prevent interception. Some of the imperative strategies are also discussed in this section.

#### *1) Asymmetric Encryption for Confidentiality and Authentication*

Blockchain uses asymmetric encryption which uses a pair of keys (public and private) to ensure both confidentiality and authentication. The public key is used to encrypt data the corresponding private key can decrypt it. The idea is to ensure that the confidentiality remains intact and only the intended recipient can access the information.

#### *2) Digital Signatures for Authentication*

Digital signatures confirm the integrity of transmitted data in a blockchain. It ensures that the information originated from the source is intact and any trivial change invalidates the whole signature. The non-repudiation guarantees that a party cannot negate their involvement in an action.

#### *3) Homomorphic Encryption (HE) & Quantum Cryptography*

HE is the conversion of data into ciphertext that can be analyzed and worked with as if it were still in its original form. HE enables complex mathematical operations to be performed on encrypted data without compromising the encryption. It differs from typical encryption methods because it enables mathematical computations to be performed directly on the encrypted data, which can make

International Journal of Current Science Research and Review
ISSN: 2581-8341
Volume 08 Issue 05 May 2025
DOI: 10.47191/ijcsrr/V8-i5-36, Impact Factor: 8.048
IJCSRR @ 2025
www.ijcsrr.org

the handling of user data by third parties safer. Quantum-resistant cryptographic techniques include: lattice- based cryptography, post-quantum digital signatures, and quantum key distribution.

## III. EXAMPLES OF REAL-WORLD USE CASES IN FINTECH AND MEDICINE

The integration of Wasm and blockchain is bringing in security and performance advancements in the realms of FinTech and healthcare, which are succinctly discussed below.

### *A. Wasm in FinTech: Powering the Future of Financial Services*

Wasm helps in solving a range of applications from decentralized financing, fraud detection, and secure payment processing. Blockchain networks are leveraging its capabilities to develop and execute smart contracts more efficiently. Wasm based smart contracts are more flexible and acalable in terms of making DeFi platforms quite responsive to user demands. Beyond trading and decentralized applications, Wasm driven fraud detection models are serving protected financial transactions and leading payment providers like PayPal are reaping its benefits. Furthermore, Wasm is simplifying cross-border payments procedures by reducing reliance on multiple mediators for faster transactions.

### *B. Wasm in Medicine: Transforming Healthcare*

Wasm has emerged as a powerful tool for medical imaging and diagnostics. As compared to baseline models, Wasm applications conserve time by directly executing health-related tasks in a web browser with premium efficiency. A great example is the Open Health Imaging Foundation (OHIF), which has integrated Wasm into its web-based imaging viewer. This allows radiologists to access and analyze medical images instantly without any stringent software requirements. Telemedicine ensures smooth, high-quality video consultations between doctors and patients while ensuring that sensitive medical information remains protected. In the field of genomics and personalized medicine, Wasm helps in speeding up DNA sequencing and other complex bioinformatics processes. Beyond individual applications, Wasm is also helping streamline communication and data sharing across the global healthcare ecosystem.

### *C. Blockchain in FinTech*

Blockchain is reshaping financial technology in terms of speeding up transactions while cutting costs. This is encouraging companies like MoneyGram and Visa to start integrating blockchain into their systems. Platforms like Uniswap are allowing users to trade cryptocurrencies straight from their own wallets, which is a step towards more financial freedom. Due to blockchain's inherent security features, Visa's B2B Connect helps businesses handle international transactions in a safer way at a broader scale. Insurance companies are making more use of blockchain-based smart contracts than ever. These digital agreements allow instant payouts based on predefined conditions. Likewise, blockchain holds immense potential for regulation compliance framework and anti-money laundering strategies.

### *D. Blockchain in Medicine*

By using a decentralized system, healthcare providers can encrypt and store medical data securely and with more freedom using blockchain networks. Typical examples are companies like MediBloc and Medicalchain, which give patients more control over their health information and allow them to safely share records with consultants. Blockchain networks help in the identification of counterfeit drugs, and leading tech firms are evolving novel solutions to ensure the authenticity of medications. Similarly, clinical trials also benefit from blockchain’s ability to create a tamper-proof system for storing research data. Block chain-based solutions have proven their competence against fraudulent claims and have saved billions of dollars by minimizing administrative overhead. Additionally, blockchain networks are making an impact by allowing secure dissemination of data between patients and doctors in remote settings.

## CONCLUSIONS

The combination of Wasm and blockchain redefines the FinTech and healthcare industries by creating new security, efficiency, and interoperability possibilities. In the financial sector, blockchain revolutionizes transactions by enabling seamless and secure cross-border payments, while Wasm enhances computational performance. In healthcare, blockchain makes telemedicine safer and more transparent while maintaining patient records and communications integrity. It can be deduced that the future of FinTech and healthcare will depend on the logical integration of blockchain’s reliability with Wasm’s high-performance capabilities.

International Journal of Current Science Research and Review
ISSN: 2581-8341
Volume 08 Issue 05 May 2025
DOI: 10.47191/ijcsrr/V8-i5-36, Impact Factor: 8.048
IJCSRR @ 2025
www.ijcsrr.org

**REFERENCES**

1. Haas, A., Rossberg, A., Schuff, D., Titzer, B. L., Holman, M., Gohman, D. & Wagner, L. 2017. Bringing the web up to speed with WebAssembly. *Proc. 38th ACM SIGPLAN Conf. Prog. Lang. Des. Implement.*, 185-200.
2. Jangda, A., Powers, B., Berger, E. D., & Guha, A. 2019. Not so fast: Analyzing the performance of WebAssembly vs. native code. *Proc. 2019 USENIX Annu. Tech. Conf.*, 107-120.
3. Nakamoto, S. 2008. Bitcoin: A peer-to-peer electronic cash system.
4. Wood, G. 2014. Ethereum: A secure decentralised generalised transaction ledger. *Ethereum Project Yellow Paper* 151, 1-32.
5. Parno, B., Howell, J., Gentry, C., & Raykova, M. 2013. Pinocchio: Nearly practical verifiable computation. *Proc. 2013 IEEE Symp. Secur. Priv.*, 238-252.
6. Kosba, A., Miller, A., Shi, E., Wen, Z., & Papamanthou, C. 2016. Hawk: The blockchain model of cryptography and privacy-preserving smart contracts. *Proc. 2016 IEEE Symp. Secur. Priv.*, 839-858.
7. Luu, L., Chu, D. H., Olickel, H., Saxena, P., & Hobor, A. 2016. Making smart contracts smarter. *Proc. 2016 ACM SIGSAC Conf. Comput. Commun. Secur.*, 254-269.
8. Bhargavan, K., Delignat-Lavaud, A., Fournet, C., Gordon, A. D., Pironti, A., Strub, P. Y., & Swamy, N. 2016. Formal verification of smart contracts: Short paper. *Proc. 2016 ACM Workshop Prog. Lang. Anal. Secur.*, 91-96.
9. Grigg, I. 2017. EOS.IO Technical White Paper v2. *block.one*.
10. Wood, G. 2016. Polkadot: Vision for a heterogeneous multi-chain framework.
11. Schwarz, A., & Schanzenbach, M. 2019. Substrate: A next-generation blockchain framework. *Parity Technologies*.
12. Lehmann, D., Kinder, J., & Pradel, M. 2020. Everything old is new again: Binary security of WebAssembly. *Proc. 29th USENIX Secur. Symp.*, 217-234.
13. Stievenart, Q., De Roover, C., & Ghafari, M. 2021. The security risk of lacking compiler protection in WebAssembly. *arXiv preprint arXiv:2111.01421*.
14. Brito, T., Lopes, P., Santos, N., & Santos, J. F. 2022. Wasmati: An efficient static vulnerability scanner for WebAssembly. *arXiv preprint arXiv:2204.12575*.
15. Harnes, H., & Morrison, D. 2024. SoK: Analysis techniques for WebAssembly. *arXiv preprint arXiv:2401.05943*.
16. Perrone, G., & Romano, S. P. 2024. WebAssembly and security: A review. *arXiv preprint arXiv:2407.12297*.
17. Jangda, A., Kelley, K., & Guha, A. 2021. A study of bugs in WebAssembly programs. *Proc. 2021 ACM SIGPLAN Int. Symp. New Ideas, New Paradigms, and Reflections on Programming and Software*, 1-16.
18. Perrone, G., & Romano, S. P. 2024. WebAssembly and security: A review. *arXiv preprint arXiv:2407.12297*.
19. Fischer, A., Fuchs, A., & Kossmann, D. 2020. Integrating WebAssembly into serverless computing platforms: A lightweight virtualization approach. *Proc. 15th ACM Eur. Conf. Comput. Syst.*, 1-16.
20. Zafar, H., Zafar, J., & Sharif, F. 2022. Automated clinical decision support for coronary plaques characterization from optical coherence tomography imaging with fused neural networks. *Optics*, 3(1), 8-18.
21. Zafar, H., Zafar, J., & Sharif, F. 2023. GANs-based intracoronary optical coherence tomography image augmentation for improved plaques characterization using deep neural networks. *Optics*, 4, 288-299.
22. Fazzini, M., & Wang, Y. 2021. WASPy: Detecting malicious WebAssembly modules in the wild. *Proc. 30th USENIX Secur. Symp.*, 1-18.
23. Kumar, D., Fischer, M., & Pradel, M. 2020. Towards finding bugs in WebAssembly modules: A case study with WasmFuzz. *Proc. 29th ACM SIGSOFT Int. Symp. Softw. Test. Anal.*, 1-12.
24. Brenner, S., & Buchmann, J. 2020. WebAssembly module isolation for secure web embedding. *Proc. 15th Int. Conf. Avail., Reliab., and Secur.*, 1-10.
25. Gadepalli, S., & Porter, D. E. 2020. Formal verification of WebAssembly sandboxing guarantees. *Proc. 31st IEEE Symp. Comput. Secur. Foundations*, 123-139.
26. Yan, C., Gao, J., & Xu, S. 2021. WASMEdge: Secure and efficient WebAssembly execution for IoT applications. *Proc. 20th ACM Int. Conf. Embed. Softw.*, 1-15.

International Journal of Current Science Research and Review
ISSN: 2581-8341
Volume 08 Issue 05 May 2025
DOI: 10.47191/ijcsrr/V8-i5-36, Impact Factor: 8.048
IJCSRR @ 2025
www.ijcsrr.org

27. Li, Y., Xu, C., & Wu, J. 2022. Enhancing WebAssembly execution efficiency on resource-constrained IoT devices. *Proc. 27th IEEE Int. Symp. Real-Time Embed. Technol. Appl.*, 56-69.
28. Zhang, H., & Wang, X. 2023. Exploring the integration of WebAssembly and blockchain for IoT security. *Proc. 11th Int. Conf. Comput. Secur. Appl.*, 187-203.
29. Sun, Z., & He, L. 2023. A survey on WebAssembly for blockchain security. *J. Comput. Secur.* 41, 3, 301-320.
30. Chen, P., & Liu, Y. 2024. The future of WebAssembly in decentralized applications. *J. Decent. Comput.* 18, 1, 55-73.

*Cite this Article: Horbenko, Y. (2025). Web Assembly and Blockchain for High-Performance Secure Front-End Systems. International Journal of Current Science Research and Review, 8(5), pp. 2279-2285. DOI: https://doi.org/10.47191/ijcsrr/V8-i5-36*