

Sergei S.

Senior Blockchain Engineer

SUMMARY

- Blockchain Developer with 5 years of experience building decentralized applications (dApps) using Solidity and Rust across EVM-compatible chains (Polygon, BSC, Base) and Solana.
- Proficient in frameworks such as HardHat and Anchor for developing, testing, and deploying smart contracts, with hands-on experience integrating wallets like MetaMask and working with DeFi protocols including Uniswap, Aave, and Alpaca Finance.
- Experienced in implementing and customizing token standards such as ERC-20, ERC-721, ERC-721A, and ERC-1155, as well as secure signing mechanisms via EIP-712.
- Skilled in smart contract auditing and fuzz testing using tools like Slither, Mythril, and RustSec, ensuring robust and secure codebases.
- Adept in version control and collaboration using Git and GitHub, with a focus on maintainable and testable code.
- Experienced in blockchain data indexing and integration using The Graph/ Subgraphs, enabling efficient access to on-chain data across applications.
- Well-versed in cross-chain development and decentralized identity (DID), contributing to secure, scalable, and user-centric Web3 ecosystems.

TECHNICAL SKILLS

Main Technical Skills	Solana (1 yr.), Solidity (4 yr.)
Programming Languages	JavaScript, Python, Rust, Solidity (4 yr.), TypeScript
Rust Libraries and Tools	Cargo
JavaScript Libraries and Tools	Ethers.js, Web3.js
JavaScript Frameworks	React
Python Libraries and Tools	Web3.py
BlockChain and Decentralized Software	Aave, Anchor, BSC, DAO, ERC-1155, ERC-20, ERC-721, Ethers.js, Hardhat, MetaMask, OpenZeppelin, Polygon, Solana (1 yr.), Uniswap, Web3.py
Version Control	Git, GitHub

Other Technical Skills	Alpaca Finance, Base, DeFi Metachain, EIP-712, ERC-721A, EVM networks, Fuzz, HTML/CSS, Mythril, PancakeSwap, Pyth, RustSec, Slither, Solhint, SPL, Subgraphs, v2/v3, WAGMI, Wallets integration, Wormhole
-------------------------------	---

EXPERIENCE

DEFI ECOSYSTEM HUB

04.2023 - till now

Description: DeFi Ecosystem Hub offers next-generation decentralized finance (DeFi) products built on DeFiChain, a Bitcoin-based blockchain that incorporates native DeFi capabilities.

Responsibilities:

- Blockchain development;
- Design project architecture;
- Prepared reports summarizing audit findings and recommendations for improvement;
- Integration with Wormhole;
- Integrated with new blockchains and side-chains to increase the multichain infrastructure;
- Implemented buy and burn algorithms;
- Optimized performance of blockchain applications and smart contracts;
- Developed smart contracts for various DeFi products;
- Integrated with decentralized exchanges and other DeFi protocols;
- Updating and maintaining smart contract code and associated documentation as required to ensure accuracy and adherence to evolving standards;
- Maintained up-to-date knowledge of emerging blockchain and Python technologies, and led the adoption of best practices within the team;
- Contributed to high-level architectural decisions for blockchain infrastructure and software systems, ensuring alignment with business goals;
- Implemented EIP-712 for secure and efficient signatures;
- Implemented robust security measures for all smart contracts to safeguard users' assets;
- Created comprehensive documentation for smart contracts, integration processes, and architecture design;
- Analyzed and refined transaction processing to improve throughput and user experience;
- Code review.

Environment: Python, Solidity, Hardhat, Solhint, Wormhole, Base, Slither, Mythril, ERC-20, ERC-1155, ERC-721, ERC-721A, EIP-712, Web3.py, Pyth, Subgraphs, Aave, DAO, DeFi Metachain, Uniswap v2/v3, GitHub.

INVESTMENT PLATFORM

06.2021 - 04.2023



Description: Platform for investment in digital assets. The De-Fi platform combines a variety of broad functionality that makes it easier to work, the automation of many functions makes the user's work easier.

Responsibilities:

- Blockchain development using appropriate technologies and frameworks;
- Blockchain integration into applications, developing APIs and ensuring compatibility;
- Smart contract development to automate processes, manage assets, and enforce business rules;
- Custom algorithm development to address specific blockchain requirements;
- ERC-721 implementation for NFTs;
- ERC-20 token development;
- Wallet integration for storing and managing digital assets;
- Build wallet connection UI using Wagmi hooks within React components;
- Smart contract development for blockchain migration;
- Mentoring new developers on blockchain concepts and best practices;
- Code review to ensure quality, identify issues, and enforce standards.

Environment: Solidity, JavaScript, TypeScript, ethers.js, Wagmi, React, HTML/CSS, web3.js, Wallets integration(Metamask), EVM networks, Polygon, Binance Smart Chain, PancakeSwap, Uniswap v2, OpenZeppelin, ProxyContract, Alpaca Finance, EIP-721, ERC-20, GitHub.

PLAY2EARN GAME

04.2020 - 06.2021

Description: Web3.0 application with play2earn mechanics, which is on movement by car. Users owning a car earn standing in traffic, being behind the wheel while traveling, moving a car around the city. Each NFT car owner earns in-game currency, which can be used in-game or cashed in for profit.

Responsibilities:

- Blockchain development;
- Conducting security audits using RustSec to identify vulnerabilities and potential risks within dependencies and code;
- Writing and testing smart contracts using SPL;
- Development of dApp applications by integrating blockchain technology with gaming mechanics, including tokenomics and NFT assets;
- Specify and implement consensus mechanism;
- Developing and optimizing system components for transmitting and receiving payments;
- Customize transaction execution;
- Optimize smart contracts for gas efficiency to minimize transaction costs for users;
- Conduct performance testing and benchmarking of blockchain transactions to ensure scalability;
- Build custom runtime upgrade mechanisms;
- Implementing utility tokens;
- Integrate off-chain data services to enhance gameplay experiences and provide real-time data for in-game events;
- Building custom metrics and monitoring systems;
- Functional test of blockchain;



- Solve security issues;
- Mentoring new developers;
- Code review.

Environment: Rust, JavaScript, TypeScript, RustSec, Cargo, AuditRust, Fuzz, Solana, Anchor, SPL, Raydium, GitHub.

EDUCATION

Computer Science and Software Engineering

