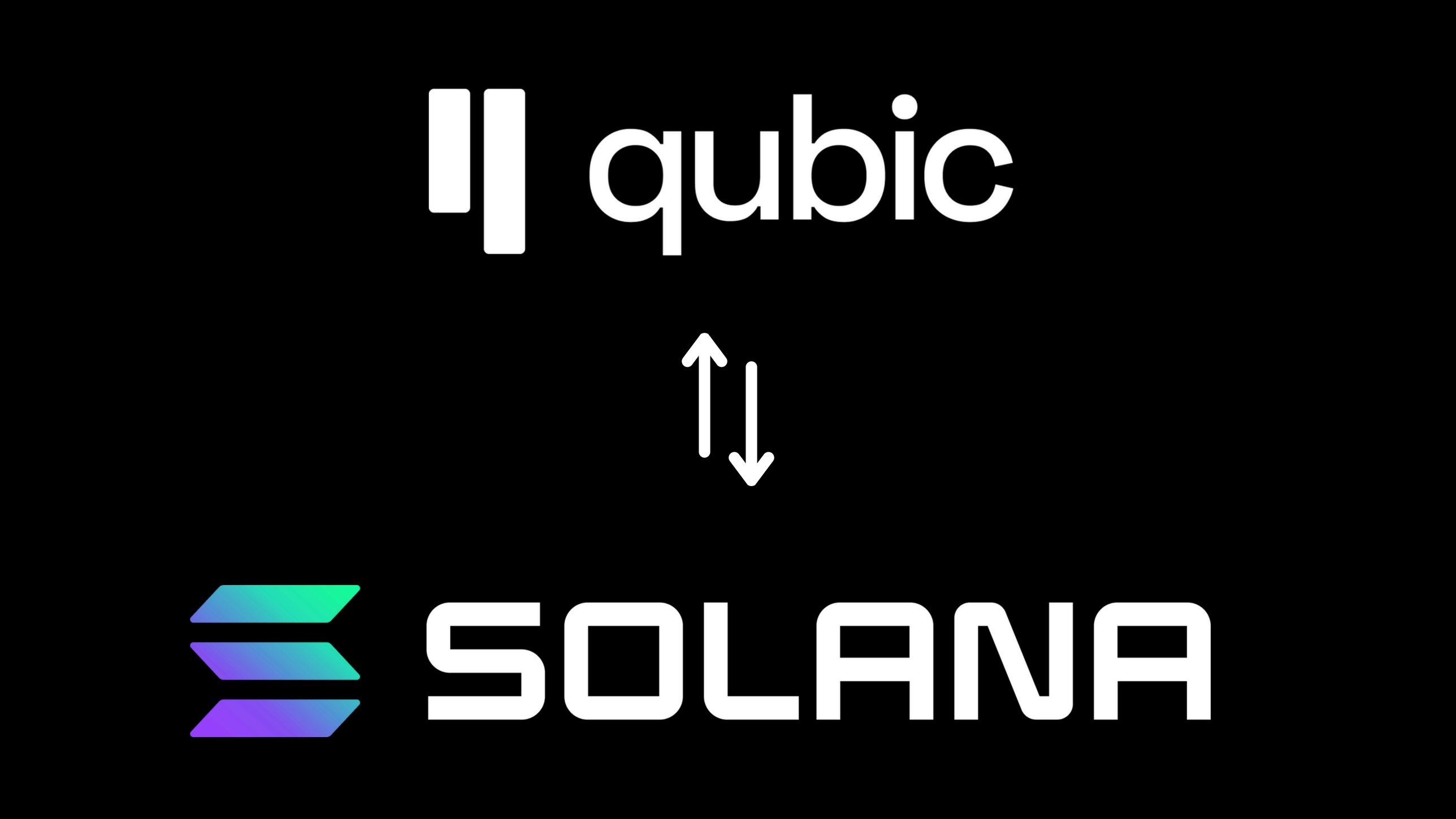
qubic
SOLANA

Overview and Requirements

The goal is to build a decentralized bridge between Solana and Qubic . The bridge must support token transfers (e.g. QUBIC coins and potentially SOL or SPL tokens) as well as arbitrary cross-chain messages, all while maintaining security and decentralization. Key requirements include:

- Bi-Directional Token Transfers: Allow users to move tokens from Qubic to Solana and vice versa. For example, moving QUBIC (the native coin on Qubic) to Solana as a wrapped SPL token, and returning it back to Qubic, as well as potentially moving Solana assets into Qubic as wrapped tokens.
- Cross-Chain Message Passing: Beyond tokens, the bridge should carry generic messages or data (e.g. signals or contract calls) from one chain to the other, enabling richer interoperability.
- Security and Trust Model: This one is a no-brainer. Aiming for a trust-minimized design.
- Production Readiness: The design should emphasize robustness, handling high throughput and edge cases, with upgradability and auditability in mind. The solution will use well-tested patterns (e.g. lock-mint and burn-unlock for tokens) and include thorough validation, error handling, and role-based access control to prevent misuse.

Bridge Architecture

At a high level, the Solana–Qubic bridge consists of on-chain smart contracts on each network and an off-chain decentralized relayer network that connects them. The major components are:

- Qubic Smart Contract (Bridge Contract): This contract allows Qubic users to initiate outgoing transfers (locking tokens and creating a bridge “order”) and also handles incoming transfers or messages from Solana (releasing tokens or processing messages). It maintains state tracking all pending and completed cross-chain orders. For the hackathon, this will be a single contract managing the bridge logic.
- Solana Bridge Program: A Rust-based Solana program deployed on the Solana blockchain. This program will manage a wrapped QUBIC token mint on Solana (an SPL token representing QUBIC coins) and possibly vault accounts for any Solana assets to be sent to Qubic. It exposes instructions to mint or burn wrapped tokens and to send or receive generic messages.
- Relayer/Guardian Network (Off-Chain Oracles): A decentralized network of bridge nodes that observe events on both chains and cryptographically attest to those events. These could be independent validators or trusted nodes chosen from the community. When a user initiates a transfer on one chain, the guardians observe that event, verify it, and collectively sign a message attestation. Once a threshold of signatures is reached (quorum), this signed attestation (often called a VAA – Verifiable Action Approval – in Wormhole terminology) can be submitted to the target chain’s contract to execute the corresponding action. This approach spreads trust across multiple parties, greatly reducing risk compared to a centralized bridge.

Process Flow:

Workflow Summary: The bridging process follows a lock-and-mint / burn-and-unlock modelb:

- Qubic → Solana (Lock and Mint): A Qubic user locks some QUBIC tokens in the Qubic bridge contract by calling a function (e.g. createOrder) with the target Solana address and amount. The contract records a new bridge order and holds the tokens in escrow. The relayer network detects this event and, after Qubic finality, signs an attestation. A transaction is then sent to the Solana program with the attestation. The Solana program verifies the guardians' signatures and, if valid, mints the equivalent amount of wrapped QUBIC (wQUBIC) tokens to the specified Solana address. The bridge order on Qubic can be marked completed (with an order status update and a record of the Solana transaction or signature) to finalize the process. The real QUBIC coins remain locked in the Qubic contract as collateral backing the wQUBIC on Solana.
- Solana → Qubic (Burn and Unlock): A user on Solana initiates a transfer back to Qubic. They call an instruction on the Solana program (e.g. burnAndBridge) providing the amount and their Qubic address. This burns the specified wQUBIC tokens on Solana (or locks the SOL/SPL asset in the program's custody if bridging a Solana asset) and emits an event or log containing the details. The guardian nodes observe the Solana transaction, and once it's confirmed final on Solana, they sign an attestation of the burn event. That attestation is then submitted in a transaction to the Qubic bridge contract (calling a function like completeOrder or releaseTokens). The Qubic contract verifies the attestation (or checks that the call is authorized by a quorum of managers) and then releases the corresponding QUBIC tokens from its escrow to the intended user's Qubic address, completing the bridge. In the Qubic contract's state, the previously locked tokens are unlocked and the order marked as completed (or a new inbound order is recorded for audit purposes.

Thank You