

Qubic Solana Bridge

Jamie Macleod

Vancouver, BC, Canada

07/09/2025

Overview

This project aims to develop a robust, secure, and fully decentralized cross-chain bridge connecting the **Qubic blockchain (a non-EVM platform)** with the high-performance **Solana network**. This bridge will enable seamless, trust-minimized bi-directional token transfers, empowering users to convert native Qubic tokens (**Qu**) into wrapped tokens (**wQUBIC**) on Solana and redeem them back on Qubic.

By implementing this interoperability layer, the bridge unlocks enhanced liquidity, broader DeFi integration opportunities, and cross-chain asset mobility for users of both ecosystems. The solution will comprise well-audited smart contracts on both blockchains, a reliable backend relayer service for secure message relay and validation, and an intuitive frontend interface facilitating smooth user interactions.

How It Works

1. User Interface(Frontend)

The frontend application, developed using **Next.js** and styled with **Tailwind CSS**, provides a responsive and intuitive user experience. It enables users to seamlessly connect their wallets—**Phantom** for Solana and **Metamask Snap** (currently experimental) for Qubic.

2. Smart Contract

- **Qubic** → **Solana**: When a user initiates a swap from Qubic to Solana, the Qubic smart contract securely locks the specified amount of Qu tokens. Upon successful lock confirmation, the backend relayer validates the event and triggers the minting of an equivalent amount of wrapped tokens ($wQUBIC$) on the Solana network.
- **Solana** → **Qubic**: Conversely, when transferring from Solana back to Qubic, the Solana program burns the user's $wQUBIC$ tokens. The backend confirms the burn event before instructing the Qubic smart contract to release the equivalent amount of Qu tokens to the user's wallet.

3.Relayer Backend(Node.js)


Swap Direction	Origin Chain Action	Backend Relayer Role	Destination Chain Action
Qu → wQUBIC	User locks Qu tokens in the Qubic smart contract	Detect and validate lock event, construct secure relay message	Mint equivalent wQUBIC tokens to user's Solana wallet
wQUBIC → Qu	User burns wQUBIC tokens in the Solana program	Detect and validate burn event, construct secure relay message	Unlock equivalent Qu tokens to user's Qubic wallet


Key Relayer Responsibilities:

- Real-time event monitoring on both chains
- Robust validation and signature verification
- Replay protection to prevent duplicate transactions
- Reliable message formatting and forwarding

User Interface

Qubic Solana Bridge

 GBIYW...DFPWF

 H3ut..UBKy

From Qubic

Qubic 0.00

To Solana

wQubic 0.00

Swap

Relayer(Validation and Mint wQubic)

```
const isValid = await confirmQubicTx(qubicTxHash);
if (!isValid) {
  return res.status(400).json({ error: "Invalid Qubic transaction" });
}
const { success, signature } = await createAndMintTokens(
  amount,
  solanaAddress
);
```



Minted Qubic




TOKEN

The WQubic Coin

Overview

 Refresh


Address

 Eseh4QUZhjKUjnLzjTnZkevVYVmse21LDfYPf7bcCSK


Current Supply

0.000023

Mint Authority

 H3ut97DhdFrq3AGA2LwqsYGeKt9hp461N5p9S6JPUBKy

Freeze Authority

 H3ut97DhdFrq3AGA2LwqsYGeKt9hp461N5p9S6JPUBKy

Decimals

9

Thanks