



Milestone 1:

Qubic Bridge

How does it work?

User Interface (Frontend):

Users connect wallets (EVM-compatible or Qubic-compatible) via bridge interface. They select a swap type—Qubic to Ethereum/Arbitrum or vice versa—and follow pre-exchange conversion steps.

Origin Chain Operations:

If a user swaps Qubic to Ethereum/Arbitrum, the backend sends an order to Qubic's smart contracts to lock or transfer the tokens. For swaps from wQUBIC on Ethereum/Arbitrum to Qubic, the backend sends an order to smart contracts on Ethereum/Arbitrum to lock wQUBIC in a FIFO queue, preparing it for the destination chain.

Backend Operations (Vottun):

Acting as an intermediary, the backend validates that orders on each network are executed correctly. It then triggers the inverse transaction on the destination chain to complete the swap, managing security checks, error handling, and communication with the frontend.

Destination Chain Operations:

The backend finalizes the transfer by instructing smart contracts on the destination chain. If converting from wQUBIC to Qubic, Qubic smart contracts release the tokens. If converting from Qubic to wQUBIC, Ethereum/Arbitrum smart contracts complete the transaction and update the order status with the transaction hash.

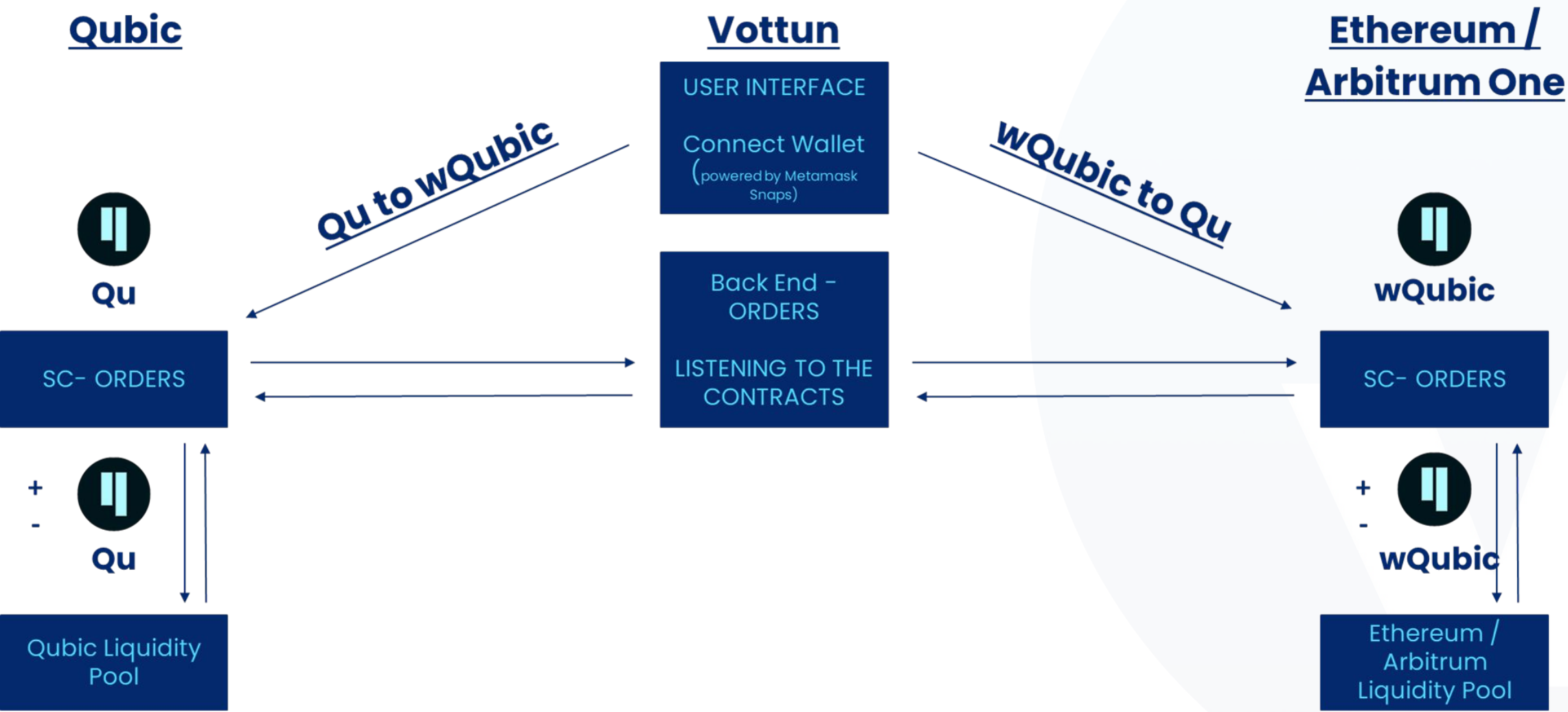
Process Summary:

Qu to wQUBIC: Qubic and Ethereum/Arbitrum smart contracts execute swap orders, updating token balances on each network.

wQUBIC to Qu: Reverse swap orders ensure tokens are returned to their original form.

Vottun: Manages and validates each step, maintaining a seamless, error-free bridge between both networks.

Bridge Architecture



Project Scope & Solution Requirements

Project Scope

Scope of Work:

- Frontend Development Interface for connecting wallets (pending to confirm by Qubic which wallet solution to apply).
- Backend Development Manage communication between Qubic and Ethereum/Arbitrum One smart contracts. Monitor transaction status and ensure accurate execution of swaps.
- Smart Contract Development & Integration.
- Testing and Security: Test and audit to ensure secure and reliable swaps.

Deliverables:

- Functional bridge between Qubic and Ethereum/Arbitrum One.
- Frontend for user interaction.
- Backend to manage swaps and smart contract execution.

Out of Scope:

- Any Extra development not considered in the proposal.

Front End Requirements

1. **Development Environment Setup:** Setup environment to support frontend development, testing, and debugging.
2. **Connect EVM Wallet:** Connection of Ethereum Virtual Machine (EVM)-compatible wallets
3. **Connect Qubic Wallet:** Connection of Qubic-compatible wallets, for users to interact with the Qubic network. For the Connection with a Qubic wallet, Metamask Snaps will be used. The UX will be the same as with Metamask.
4. **Pre-exchange Qubic to wQubic on Ethereum / Arbitrum:** Set up user interface for converting native Qubic tokens to wrapped Qubic tokens (wQubic) on the Ethereum or Arbitrum networks before the exchange process.
5. **Pre-exchange wQubic on Ethereum / Arbitrum to Qubic:** Set up interface for converting wQubic tokens back to native Qubic tokens on the Qubic network before the exchange process.
6. **UI Layout:** Design the frontend interface layout for the entire exchange and wallet interaction process.
7. **Create Order for Exchange from Qubic to wQubic on Ethereum / Arbitrum:** Allow users to initiate an exchange order to convert Qubic tokens to wQubic on Ethereum or Arbitrum networks.
8. **Create Order for Exchange from wQubic on Ethereum / Arbitrum to Qubic:** Enable users to initiate an exchange order to convert wQubic tokens back to native Qubic tokens on the Qubic network.
9. **Error Management:** Handle and display errors on the frontend, to inform users of any issues.

Back End Requirements

1. **Development Environment Setup:** Set up a environment for the backend for server-side development and testing.
2. **Order Listener on Qubic:** Set up a service to monitor and listen for incoming orders on the Qubic network, triggering the necessary backend processes.
3. **Order Listener on EVM:** Create a service to monitor and listen for incoming orders on Ethereum Virtual Machine-compatible networks, facilitating interactions between Qubic and Ethereum.
4. **Endpoints for Rating and Fees:** Develop backend endpoints to manage rating and fee calculations related to the token exchange process.
5. **Order Processor:** Create an order processing system to manage end-to-end execution of token exchanges and interactions between Qubic and Ethereum.
6. **Transaction Component:** Create a component to manage and store transaction records (history and details).
7. **Audit Component (Operations Ledger):** Create an audit component to maintain a record of all operations for token exchanges.
8. **Error Handling with Frontend:** Implement backend logic to handle errors and communicate them to the frontend with error messages.

Solidity Requirements

Orders Smart Contract: Develop a smart contract to manage and record the exchange orders on Ethereum-based networks.

1. **Push Order Function (FIFO):** Function that adds new orders to the order queue in a First-In-First-Out (FIFO) manner.
2. **Pull Order Function (FIFO):** Function that retrieves and processes orders from the order queue following the FIFO principle.
3. **Update Order Function:** Function that allows updates to existing orders, such as modifying the status or details of an order.
4. **Refund Order Function:** Function to handle refunds in cases where an order cannot be fulfilled, ensuring users receive their assets back.
5. **Security Verification Functions:** Implement functions to validate and secure each transaction, ensuring that only authorized and secure exchanges occur.

Qubic Smart Contracts Requirements

Orders Smart Contract: Develop a smart contract to manage and record exchange orders on the Qubic network.

1. **Push Order Function (FIFO):** Function that adds new orders to the order queue in a FIFO manner on the Qubic network.
2. **Pull Order Function (FIFO):** Function to retrieve and process orders from the order queue on the Qubic network, following FIFO.
3. **Update Order Function:** Function to update order details or status on Qubic as needed.
4. **Refund Order Function:** Function to manage refunds for unfulfilled orders on the Qubic network.
5. **Security Verification Functions:** Functions to ensure each transaction security, adding an extra layer of validation on the Qubic network.

Liquidity Pools – To Confirm with Qubic

1. **LP Initialization and Configuration:** Smart contract to create the Liquidity Pool on both Ethereum Virtual Machine (EVM)-compatible networks and Qubic, based on parameters such as deposit limits, governance settings, and initial liquidity rules.
2. **Deposit and Withdrawal Functions:** Functions to allow liquidity providers to deposit and withdraw assets according to the governance rules (lock periods, early withdrawals...)
3. **Fee Calculation and Distribution:** Functions to calculate transaction fees for each liquidity action.
4. **Risk Management Mechanisms:** Risk management functions to detect liquidity levels & market volatility and activate response.
5. **Monitoring and Alert Triggers:** Monitoring functions to track liquidity levels in real-time. Include alert triggers that activate notifications when liquidity falls below thresholds / unusual activity occurs.
6. **Bridge Integration Functions:** Functions to allow the Liquidity Pool to interact with the created bridge system, supporting asset transfers and managing liquidity across both EVM and Qubic networks.

***THIS REQUIREMENTS NEED TO BE CONFIRMED AND VERIFIED WITH QUBIC**

Other Requirements

1. **Error Management:** Error handling across the frontend, backend, and smart contracts.
2. **Unit Testing:** Perform unit testing for all individual components.
3. **Integration Testing:** Conduct integration tests to verify that components work together.
4. **End-to-End Testing:** End-to-end testing to simulate real user scenarios and ensure the entire process flows from start to finish.

User Journey

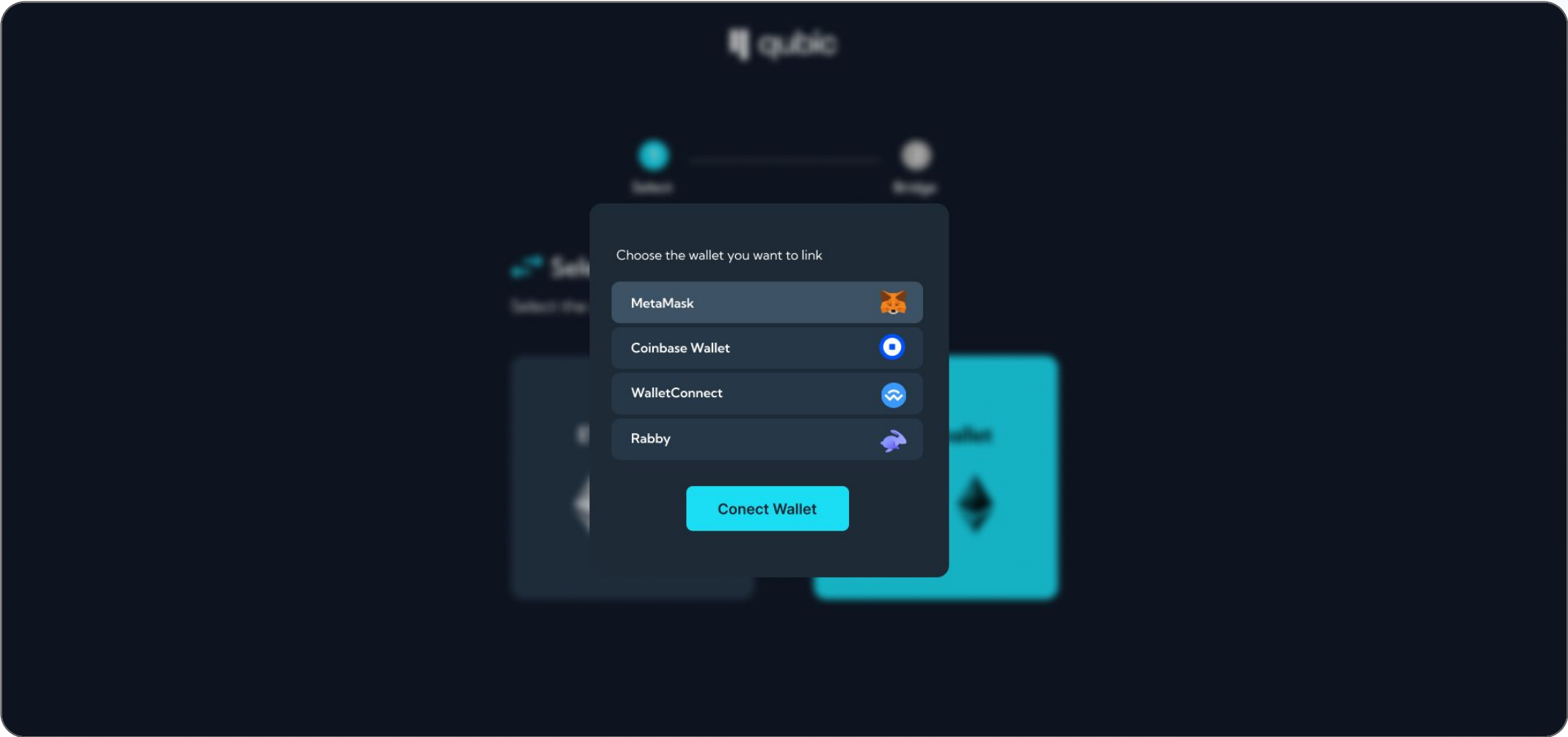
*The design is indicative and may be subject to change.

Design & User Journey



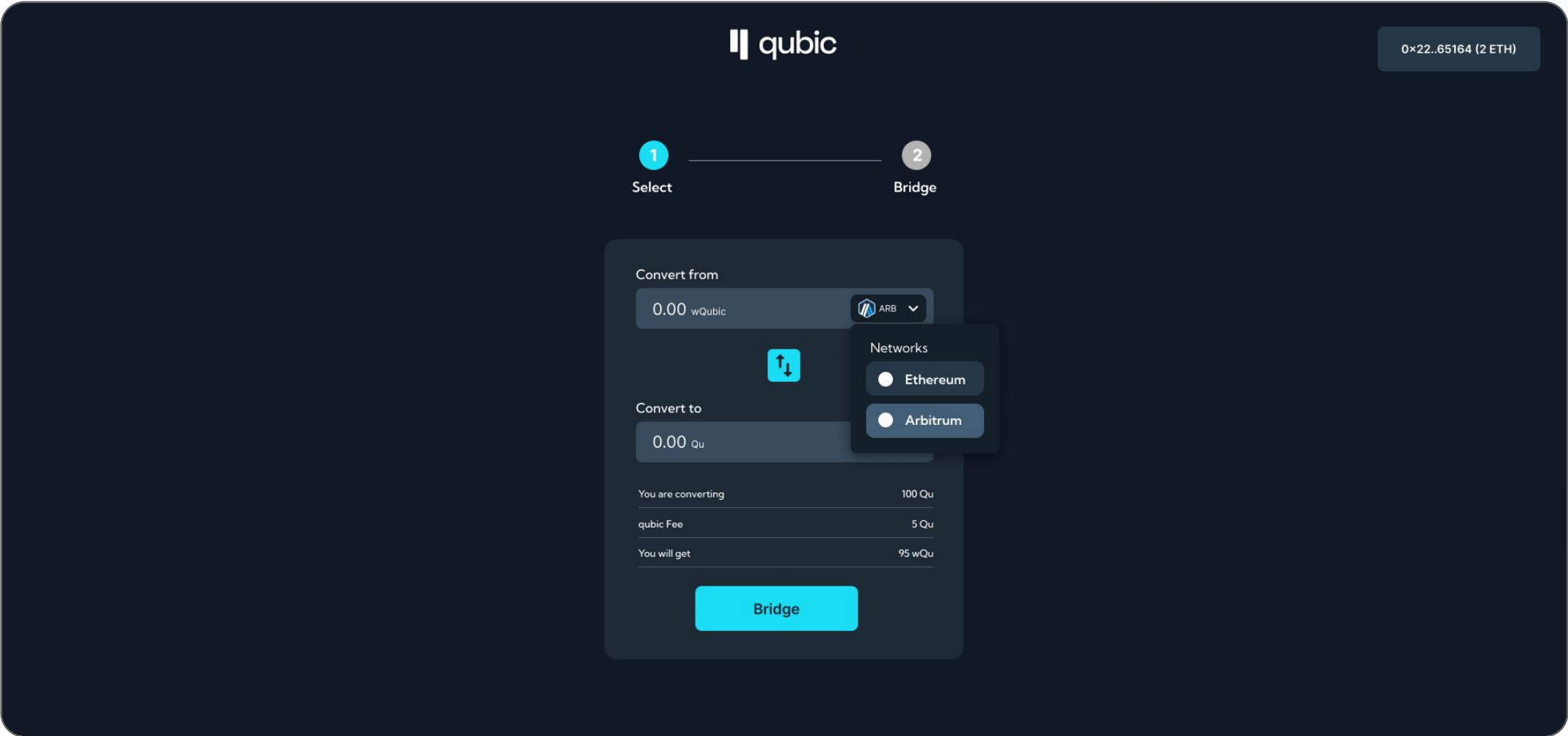
Bridge Type Selection (EVM to qubic / qubic to EVM)

Design & User Journey



Connect Wallet (EVM)

Design & User Journey



Introduce Import + Select the Network

Risks and Dependencies

This section outline potential risks and dependencies for the development of the Qubic Bridge, including mitigation strategies and responsible parties. This section is 'live' and will be updated in the coming days as new dependencies or risks are discovered in the shared Github (<https://github.com/vottundev/vottun-qubic-bridge-docs/blob/main/RISKS.md>)

Risks

Risk ID	Risk Description	Impact Level	Probability	Mitigation Strategy	Responsible
R1	EVM Smart contract vulnerabilities leading to token loss or theft	High	Medium	Conduct comprehensive audits with third-party security experts before launch; perform extensive testing.	ceseshi
R2	Qubic Smart contract vulnerabilities leading to token loss or theft	High	Medium	Conduct comprehensive audits with third-party security experts before launch; perform extensive testing.	anarojoagusti
R3	Cross-chain bridging communication can fail in the middleware, resulting in delayed or lost token transactions.	High	Medium	Implement a retry mechanism for failed transactions as well as a refund strategy	alexlopezt
R4	Rising gas prices make bridges economically unviable to use	Medium	High	Implementation of an adaptive fee structure; consideration of batching of transactions in order to reduce costs.	alexlopezt, rasito99
R5	Network congestion affecting transaction confirmation times	Medium	Medium	Notify users of potential delays and provide flexible transaction timeout settings.	alexlopezt
R6	Poor user experience due to complex transaction processes	Medium	Medium	Design a user-friendly interface; conduct usability testing and gather feedback.	bmora-vottun, vicargo
R7	Inadequate liquidity for token swaps across the bridge	Medium	Low	Establish partnerships with liquidity providers and incentivize early liquidity provision.	qubic team
R8	Loss of trust as a result of security incidents or miscommunication with users	High	Medium	Develop a robust incident response plan and maintain clear, transparent communication with users.	TBD
R9	Llack of attention from Qubic's stakeholders to the development team	High	Medium	Have the necessary communication channels to be able to resolve the development team's doubts	Qubic Team

The rows highlighted in light blue (R7) pertain to the Qubic team's responsibilities and may represent a Risk/Dependency for Vottun.

Dependencies

Dependency ID	Dependency Description	Impact Level	Mitigation Strategy	Responsible
D1	Lack of deep knowledge about Qubic ecosystem	Medium	the whole team needs support from Qubic with any question about how to implement and use the smart contracts as well as communication with the chain.	Qubic team, Vottun team
D2	Dependency on Redis pubsub strategy to be implemented and released to get smart contract events	High	Release the strategy at the development stage as soon as possible, and later on the main network before releasing the bridge.	qubic team
D3	Integration with third-party wallet providers	Medium	Ensure compatibility with leading wallet providers; test integrations for stability on a regular basis.	vicargo – alexlopezt
D4	We will need to work with liquidity providers and exchanges for token swaps, assuming liquidity pools are used.	High	Maintain active communication with partners early in the development process.	qubic Team
D5	Reliance on the stability of blockchain networks and APIs (e.g. Ethereum, Binance Smart Chain)	High	Maintain communication with the blockchain providers; establish a fallback mechanism for API failures.	vottun team, qubic team
D6	Security audits and review processes by 3rd party	High	Schedule audits well in advance.	rasito99
D7	Dependency on reliable server infrastructure for bridge operations	High	Use scalable, decentralized infrastructure options where possible; have disaster recovery plans.	vottun Team
D8	Secure and compliant storage for user transaction data	Medium	Implement data encryption and access control policies; work with compliance to align on data retention.	alexlopezt

The rows highlighted in light blue (D1, D3 & D4) pertain to the Qubic team's responsibilities and may represent a Risk/Dependency for Vottun.



CONNECT WITH US!

 www.vottun.com

 [@vottun](https://twitter.com/vottun)

 Email: luis@vottun.com