Implementation Strategy for USD-Collateralized Stablecoin on Ethereum Layer 1 & Linea

Leveraging Consensys Products for Secure and Scalable Solutions

TheNewAutonomy
August 2024

Objectives...

... extend the availability of USD-collateralized stablecoin from Ethereum to Linea

... provide a seamless user experience between networks

... ensure security of the stablecoin on and between Ethereum and Linea

... provide future options to issue a new Linea native token

... provide good liquidity management between chains

... leverage the tools and technologies of Consensys

Questions

1. How quickly do I want to be available on Linea?
2. How much code and infrastructure do I want to create and maintain?
3. Is my token on Linea a plain ERC20 or is there custom functionality?
i.e. Address blocking, Regulatory (KYC, AML features), reporting, liquidity balancing functions.
4. Liquidity management, compliance, integrating with wider ecosystem and community

Bridge Implementation - Phases

Bridge Phases

Phased implementation approach

1

Initial Deployment on Linea via Bridging

- Canonical Token Bridge to Linea
- Linea Message Service for secure communication between networks
- Focus on security and liquidity during bridging

2

Seamless User Experience

- Integrate with MetaMask for a simple wallet experience
- Utilise Infura for reliable data availability and APIs

3

Native issuance on Linea

- Issue a new token on Linea
- Explore the role of Consensys Rollups and zxEVM to enhance scalability and security

1

Initial Deployment on Linea via Bridging

Bridge options

Linea Native Bridge

Flexible bridging options.

Canonical token on Linea.

- Default
- Custom Token
- Custom Bridge & Token

Third-Party Bridge

LI.FI and Socket aggregators

Bridge providers supported

- Hop
- Connext
- Squid (Axelar)
- Across

MetaMask Bridge

Simple User Experience

Fast and Cheap for user

Bridge & Swap

Phase 1

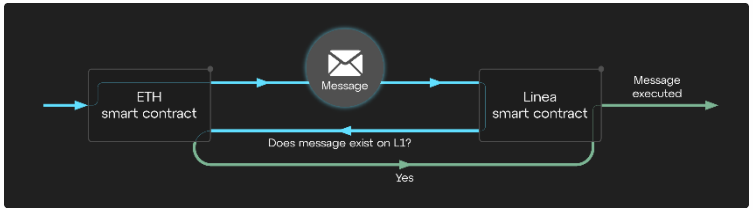
Initial Deployment on Linea via Bridging – How it works

Canonical message service

1

Components
- Smart Contract on Ethereum
- Smart Contract on Linea
- Postbots messaging service

Process (birds eye view)
- Call sendMessage(...) method in source chain MessageService.sol contract
  Args (_to, _fee (optional), _calldata (created with abi.encode(...))
- Postbot listens for call then delivers message to contract on target network
- Contract executes
- If fee paid, claimMessage(...) triggers automatically, otherwise call manually to pay fee
- Asset is delivered to _to address.
- Extra check to ensure call came from MessageService *onlyMessagingService*
- *Optionally, use onlyAuthorizedRemoteSender to ensure trusted contract sent message, such as from a dApp contract on source chain*



Phase 1

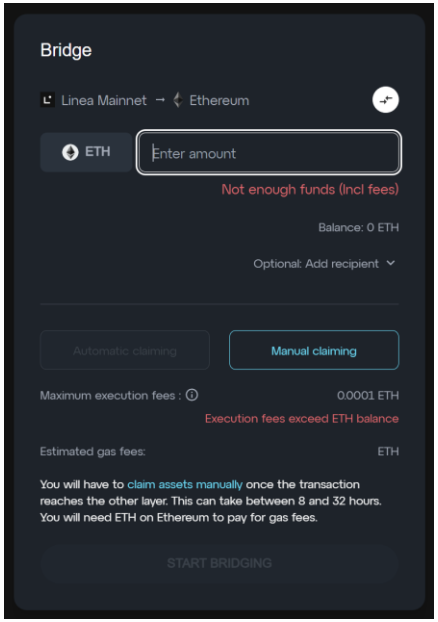
Initial Deployment on Linea via Bridging – Canonical Token Creation

Default Linea bridged token

1

Benefits
- Codeless
- Tried & Tested bridging
- Standard bridge URL and interface for users
- Operate alongside other popular bridged tokens

Limitations
- Bridged token is a basic wrapped ERC20 token
- No custom logic in bridged token



Phase 1

Initial Deployment on Linea via Bridging – Canonical Token Creation

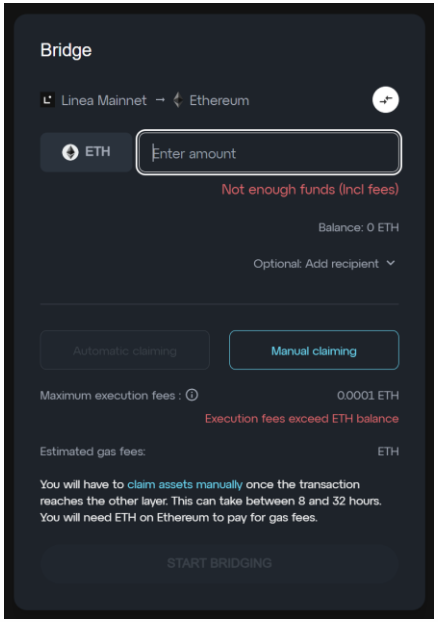
1

Custom bridged token

Benefits
- Add custom logic to bridged token
- Tried & Tested bridging
- Standard bridge URL and interface for users
- Operate alongside other popular bridged tokens

Limitations
- Need to write, test and audit custom ERC20 smart contract for Linea
- Need to work with bridge team to set up bridged token
- Can't be carried out if already bridged to Linea

Considerations
- When providing a custom bridge UI, use a dedicated Infura node for maximum data access reliability and performance



Phase 1

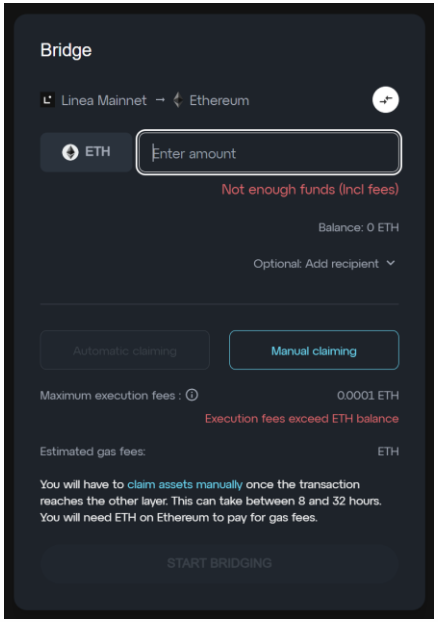
Initial Deployment on Linea via Bridging – Canonical Token Creation

1

Custom bridge and token

Benefits
- Leverage tried, tested and audited bridge code base
- Write custom ERC20 smart contract for Linea
- Adjust bridging logic and functionality as needed to support bridged token
- Freedom to design a unique user experience through a new UI

Limitations
- Need to write, test and audit custom ERC20 smart contract for Linea
- Need to modify, audit and maintain bridge code and infrastructure
- Need to work with bridge ensure token can't be bridged by default bridge
- Increased responsibility, legal and financial risk maintaining custom solution
- Token no longer found in default bridge alongside other popular tokens



Phase 1

Initial Deployment on Linea via Bridging - MetaMask

MetaMask bridge

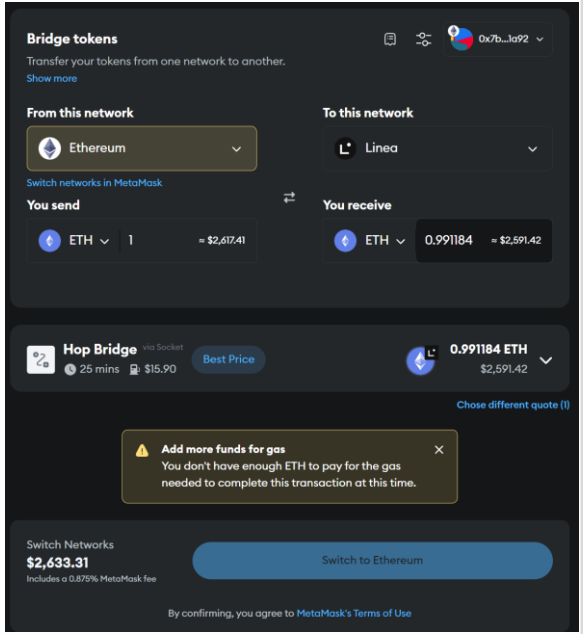
1

Uses the Li.FI and Socket aggregators to find the cheapest bridge route.
Bridge providers supported by LI.FI and Socket that support Linea.

- Hop
- Connext
- Squid (Axelar)
- Across

Benefits

- Very simple user experience
- Finds the cheapest route to bridge via multiple bridges
- Can swap and bridge as a single transaction
- Faster than the Linea Native Bridge
- Doesn't require a manual claim action on the receiving network



Phase 1

Initial Deployment on Linea via Bridging – Third Party

1

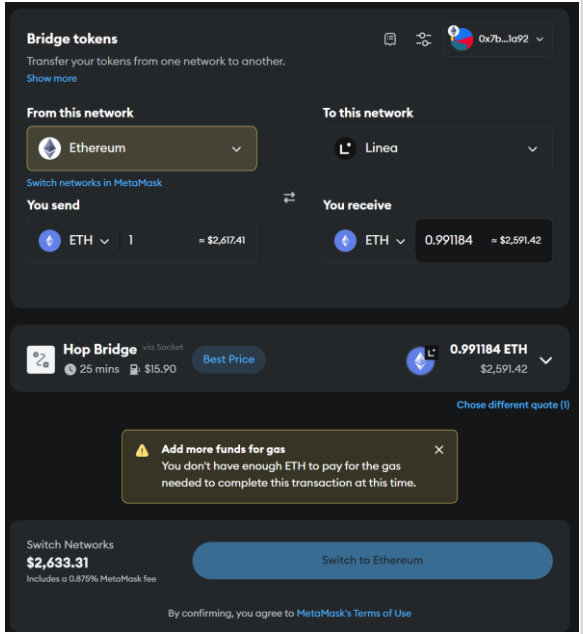
Third-Party bridges

Hop bridge
- Hop DAO to get token added to bridge
- Uses a non-canonical Hop bridge token to bridge
- Uses AMM's on each chain to balance price and availability
- Arbitrage between networks maintains price

Connext
- Messages are aggregated to Ethereum L1
- Fraud protection with fallback to Linea bridge
- Routers front run the transfer using canonical tokens

Squid (Axelar)
- Squid runs on Axelar's infrastructure
- 1 week turn-around to whitelist a token via online form
- Canonical token

Across
- Funds escrowed in a settlement protocol
- Request for quote, relayers fulfil quote.



2

User Experience & Compliance

Phase 2

User Experience

2

dApp UI – Best Practice

- Build and dApp to be user-centric. Consider user flows.
- Independent UI evaluation (i.e. Bunnyfoot) for an intuitive customer experience
- Mobile and desktop experience
- Accessibility such as color blindness, visually impaired, mobility restricted users
- Multilingual support in dApp
- Integration into popular tools (PDF, Excel etc)
- Minimize number of user actions, including transactions to sign
- Documentation, user videos, user guides, onboarding guides
- Integrate broadly. Consider wider ecosystem and their user base.
- Optimize contracts to be cheap and fast
- User support

Phase 2

User Experience

2

Metamask

- Popular / familiar interface
- Snaps – Option to create a custom snap utilizing features of the token
- MetaMask bridging
- Audited and trusted
- Support for hardware wallets

Infura

- Reliable node with high availability for a better customer experience
- Different node types including full Archive Nodes
- Low congestion with a dedicated node
- Scalable infrastructure
- Consistent performance

Phase 2

User Experience

2

Dilligence

- Customer confidence from a reputable auditor and professional testing tools
- Smart contract audits by highly experienced and qualified code auditors
- Tools for analysing smart contracts for common vulnerabilities
- Automated scanning tools
- Continuous monitoring and verification for security vulnerabilities
- Professional testing techniques such as fuzzing

Protect the user

- Bug bounty programs
- For any tokens held, use multisig or smart wallet
- Provide education to users regarding online safety such as phishing scams and benefits of hardware wallets
- Ensure KYC/AML compliance requirements met where needed. Bake into token? Whitelisting or blacklisting?
- Ensure any required licensing is in place
- Consider regional regulatory and legal requirements such as GDPR in Europe and Standard Contractual Clauses for international data transfers
- Work with regulators (i.e. SEC in the USA, FCA in the UK, DNB in the Netherlands)
- Financial reporting, particularly related to collateral
- Defence in Depth for infrastructure, pen-testing, fuzzing, DDoS mitigation, strict WAF rules.

3

Native Issuance on Linea

Phase 3

Native Issuance on Linea

3

Is it required?

- Was a custom contract deployed in phase 1?
- What is the motivation behind a new native token?

The token

- A new ERC20 token deployed to Linea
- Original canonical token remains on Linea and bridge (two tokens). Not all tokens will be bridged and converted
- Burn the native token on Ethereum and issue a new token on Linea
    - Bridge original token back to Ethereum
    - Send to contract to burn on Ethereum
    - Canonical Message Service communicates to Linea the quantity of tokens received and sender address
    - Contract on Linea mints and sends new tokens to owner's wallet
- Consider underlying collateral, whether it's combined or separate and how to manage

Liquidity Management

Rebalancing

Liquidity challenge

- Bridges need Liquidity on Linea to fund transfers
- Liquidity fragments between chains, reducing availability
- Reduced availability increases risk of slippage and volatility

Tools and Strategies

- Seed liquidity on Linea via AMMs
- Incentivize Liquidity Providers with rewards via yield farming and pool staking
- Automate liquidity rebalancing between networks
- Partnerships with Market Makers
- Continuous monitoring and review
- Support bridges that use the canonical token

AMMs

Supporting liquidity on Linea via AMMs

1

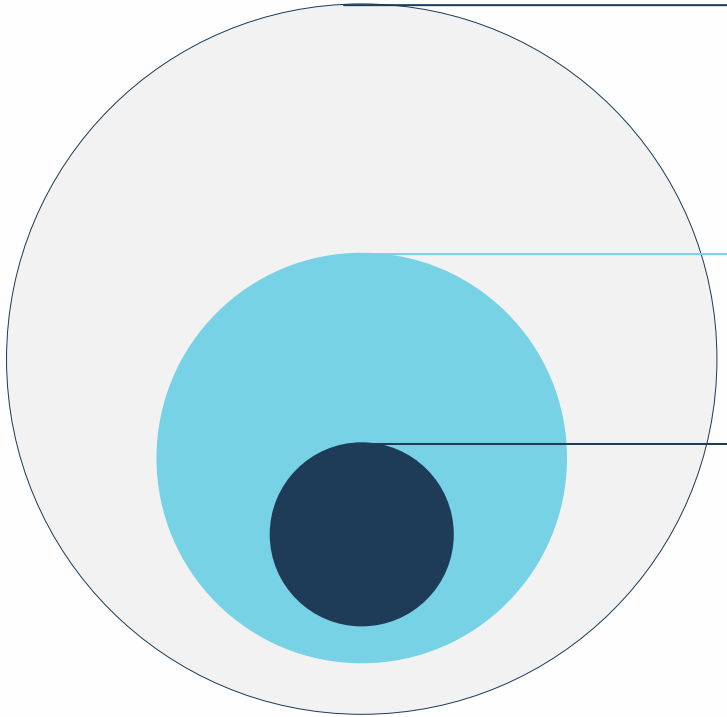
Why?

- Deep pools on AMMs ensure liquidity is available to fund transfers
- Hop Bridge requires Hop tokens paired with Canonical tokens
- Pools support arbitrage to balance prices
- Deep pools reduce volatility

How?

- Partner with AMM providers to maximize benefit
- Create small number of deep pools pairing token with quality assets such as ETH
- Incentivize Liquidity Providers with rewards such as ETH and additional stables
- Build partnerships to jointly fund pools and offer joint rewards

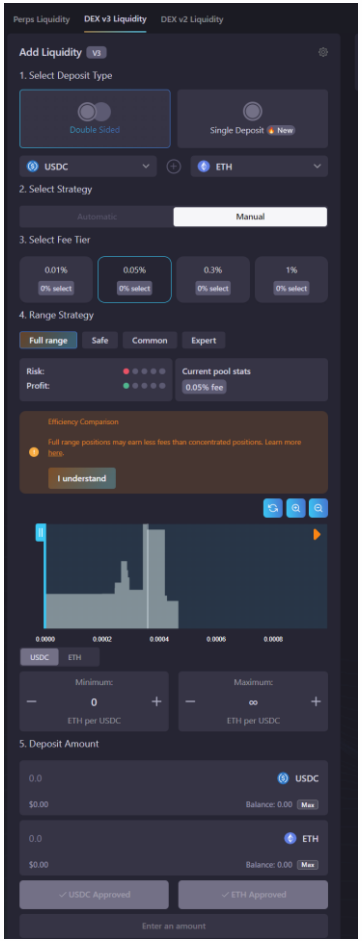
Program of Liquidity Provisioning



Market takes over, expanding pools and rebalancing through arbitrage between networks

Increase pool size through investment

Create pools with seed liquidity on AMMs



Actively work with AMMs

Connect with AMM teams
Identify listing requirements and partnership programs

Create Liquidity pools
Small number of deep pools of quality assets paired with token

Incentivize Liquidity Providers
Work on reward programs for LPs and farms such as Liquidity Mining

Build out a healthy ecosystem
A healthy ecosystem underpins bridges and the token itself

Automated Rebalancing

Use tools and strategies to rebalance between networks

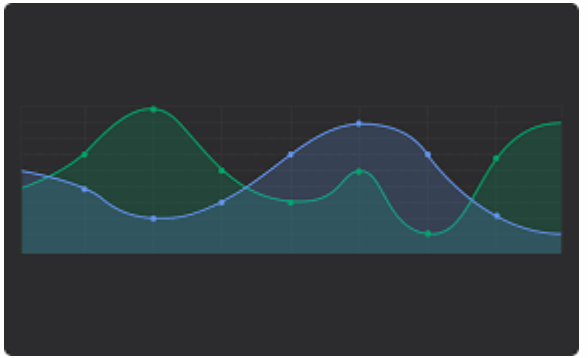
1

Why?

- Risk of fragmentation can reduce availability of tokens on Linea
- Could disrupt bridging if liquidity is unavailable
- Markets can change quickly and so the need to detect, and response must be fast

How?

- Design strategy including liquidity levels on Linea to be maintained
- Smart contract on Linea and Ethereum monitors availability, price or volume
  - Triggers when liquidity is low or on a schedule
  - Tx volume or slippage via oracles is high
  - Complex algorithmic monitoring and balancing or predictive
- Consider gas fees, performance and benefits of each approach
- Can be as simple as a liquidity threshold or as complex as off-chain ML
- A powerful tool for building a secure ecosystem



Market Makers

Partnering with market makers to support the economy

1

Why?

- Market Makers build out deep order books, ensuring buy and sell orders are available for the market and supporting larger transfers to balance prices
- Reduces volatility and slippage
- Encourages users to trust, trade and use a new coin

How?

- Partner with Professional Market Making businesses and/or DeFi markets
- Reward market makers either through fees or revenue sharing
- For professional businesses, check contracts and responsibilities, set targets
- Work with Market Makers on CEXs and DEXs which also opens up to trading bots

Monitor & Review

Have a program of monitoring and review in place

1

Why?

- Markets change quickly, stables risk getting unpegged
- New regulations
- Create feedback loops to adjust liquidity

How?

- Develop internal program of monitoring and reporting
- Work with Market Makers and have frequent two-way communication
- Have plans in place for conditions. Planning mid-event is the worst time.