



HECO



BSC



Fantom



Polygon



Avalanche



Gnosis



Telos



Fuse



Optimism



Arbitrum



Harmony



Hoo



KCC



Moonriver



Ethereum



OKEx



Elastos



Cronos



IoTeX

Elk.Finance

The DeFi Gateway



Elk Finance

Elk is a peer-to-peer network for cross-chain value transfers and blockchain interoperability.

ElkNet, the powerful cross-chain engine at the heart of the network, allows for secure and efficient transfers of value and information, reducing friction and asset fragmentation between blockchains.

Try ElkNet

Elk Finance

Elk Finance combines several innovative features:

- Farming-as-a-Service allows projects/users to create custom farms
- Transfer via ElkNet in seconds between all the chains Elk supports
- Cross-chain tokens for value transfer via Proxy Tokens & \$ELK
- ElkDEX deployment on all chains, including farm contracts
- Impermanent Loss Protection (ILP) for liquidity providers

Farming-as-a-Service (FaaS)

- Allows anyone to create a farm at app.elk.finance
- Allows up to 15 reward tokens at once
- A UI built from the ground up allows for a hassle-free deployment with no coding required
- Users can take advantage of Elk's on-chain Impermanent Loss Protection for any given farm on BTTC
- Bridging-as-a-Service will allow cross-chain interaction with FaaS

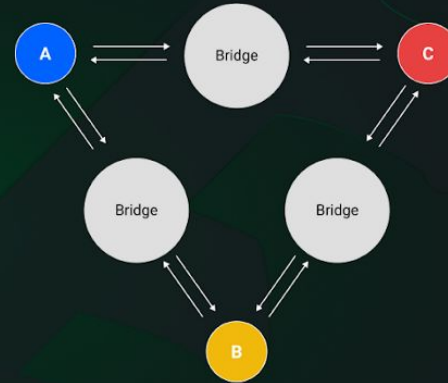
ElkNet v2



All reservoirs = total supply (42,424,242 ELK)

With ElkNet v2, the total ELK supply is deployed in advance into a "reservoir" (reserve pool) on each chain. Instead of minting + burning or locking + releasing tokens for cross-chain transfers, the ElkNet prompts tokens to move in and out of the reservoirs, resulting in fewer smart contract risks, greater transparency and lower fees.

Other cross-chain bridges



In regular cross-chain bridging, tokens are either locked into a contract on one chain and released as a wrapped token, or "burned" (permanently locked) and minted on the other side. Each method introduces additional complexity and security risks: "mint" functions can be exploited to create infinite tokens, while lock + release requires exit liquidity to operate.

Cross-chain SDK = Bridging-as-a-Service

- The Elk Software Development Kit (SDK) allows developers to create proxy tokens, utilising the ElkNet's lock and release reservoir system for cross-chain transfers.
- ElkNet can be utilized as the native bridge in the backend.
- Cross-chain bridging applications are possible using the SDK.
- Elk SDK can be considered synonymous with Bridging-as-a-Service (BaaS).

Note: Mint/burn functionality is also supported.

Elk SDK

- **Simple** – swiftly create a transparent bridging solution.
- **Secure** – built in protection of double spend and 51% attacks.
- **Decentralized** – post-alpha, the ElkNet will introduce decentralization via a proof-of-stake node framework.
- **Customizability**– Applications and bridges built on top of ElkNet can customize their bridging interface as they see fit.
- **Support** – Elk's development team will help create and manage the bridge.
- **Minimal Maintenance** – Exit liquidity problem eliminated with automated arbitrage.

Elk SDK

- **Interoperability** – no third-party integration limits.
- **Cost Effective** – BaaS applications use ElkNet, making them gas efficient and fast.
- **Reduced Fragmentation** – bridge directly to existing contract addresses instead of having to create new token contracts.
- **Compatibility** – Its unique design allows ElkNet to be compatible with all blockchains, whether they are EVM or not.

Elk SDK Alpha

ELK SDK Alpha will last until sufficient stress testing proves it reliable, it will then move over to the beta version once enough testing has taken place.

Comprehensive feedback and implemented fixes will allow a swift move forward to the beta phase.

The Elk SDK will remain in beta for approximately 1 year, depending on usage, development, and the depth and variety of testing completed. During the beta, continuous feature additions will enhance the functionality of the SDK.

Social Links



[Email](#)



[Twitter](#)



[Business Proposals](#)



[Documentation](#)



[dApp](#)

Thank you



Elk.finance