The Surplus Protocol v1.0

Kevin Liu

August 30, 2023

Abstract

The Surplus Protocol is a decentralized and non-custodial approach that empowers users to leverage market discrepancies in token price across different blockchains through the use of Hyperlane and leveraging single sided staked asset pools on destination chains to eliminate and overcome the constraint of the time taken to bridge.

1 Introduction

Unique arbitrage opportunities consistently present themselves to the market all the time, some examples include: general market arbitrage opportunities, stablecoin depegs, protocol hacks and fundamental market moving news. The Surplus Protocol focuses on apprehending potential openings in the decentralized market exclusive of opportunities which present themselves in the centralized market (i.e., CEXs). The central focus of the Surplus Protocol revolves around the existence of parallel markets across different blockchains. E.g. TOKEN/USDT (Optimism), TOKEN/USDT (Arbitrum).

The undeniable problem associated with taking advantage of these situations is the time taken to transfer funds from one blockchain to the other (This problem also occurs in a CeFi based arbitrage scenario where deposit/withdrawal times are a huge issue). The Surplus Protocol aims to solve this.

2 Overview

A high level overwiew can be described as such:

1. We suppose that $TOKEN price on chain A is at 1.00
2. We suppose that $TOKEN price on chain B is at 1.50
3. Alice sees this as an arbitrage opportunity $\Rightarrow$ she buys $TOKEN on Chain A
4. She immediately sends her amount of $TOKEN across Hyperlane to Chain B
5. $TOKEN from the pool on Chain B is sold

6. Once $TOKEN arrives on the other side of the bridge, this is used to replenish the pool on Chain B

7. Vice versa if considering Chain B to Chain A.

3 Spotting an opportunity

The Surplus Protocol itself is not tooling nor is it software to detect such opportunities. It is merely a means of acting upon already realized market opportunities.

4 Asset Pools

The Surplus Protocol will offer single sided staked asset pools to DeFi users. The return on such pools will be variable rather than constant; dependent on the magnitude and frequency of trades which the users of Surplus make. 5% of the profits from an arbitrage performed by a Surplus user will be redistributed back to stakers.

5 Smart Contracts

For the protocol to work. Two smart contracts must exist on each chain. One on chain A, and one on Chain B. We take advantage of Hyperlane's autmatatic relayers for activities with EVM chains, and specialized relayers when delaying with non-EVM enviroments such as Solana or Sui.

6 Finality

We run into a huge issue of finality with the protocol. If we want to detect Alice's transaction as soon as possible, and then act on it, how can we be sure that Alice's bridge transaction does not revert and then the liquidity pool is short, and Alice has all of her assets back?

We solve this buy asking for a 1:1 backed collateral deposit before the bridge transaction, so that if the bridge transaction reverts/cancelled/maliciously is manipulated to not go forward, that deposit is kept.

7 Oracles

A lot of thought and consideration has been put into thinking about what the fastest way of notifying chain B about the incoming transaction is. A consideration might be using Hyperlane's messaging protocol, but still, that would still require to pass along the guardians even if you set the requirements to non-finality signage.

So the solution is in fact, not very programmatic, but does the job. We use a restAPI backend to listen for events such as: collateralDeposted (bool), transactionSent(bool), amount-Tokens(num), contractAddressTokens(address), once these events are emitted because of the

function calls. Ethers automatically detects these events and POSTs them to the API. The API is wrapped inside a chainlink oracle, and the chainlink oracle is within a function in chainB. The user must call the function on chainB for it to detect it though. Which is a huge disadvantage (in terms of time), but no feasible workaround at present. As it involves, switching the chain in their wallet, and also signing another transaction. Once the function on chainB is called, the funds from the liquidity pool can start to be distributed to perform the arbitrage. This solution not only is slow and a bad UX for the user, but also now incites a centralized and off-chain component to the protocol.

8 Composability

The Surplus Protocol has huge opportunities as a plugin module for other protocols. An example of a use case is a multi-chain future/derivatives protocol whereby Bob might see that the futures price on chainB is better than chainA so he decides he needs to have liquidity on chainB as soon as possible.