Openhedge

A unified smart contract standard for decentralised structured finance

Intro

At the time of writing, nearly USD 9bn is locked into decentralised finance applications. The introduction of collateralised lending protocols has created a solid set of incentives for digital asset holders to take active part in decentralised finance. Unlike many previous attempts, the current lending primitives have succeeded in adding undeniable value to the decentralised economy by increasing market liquidity, enabling short selling and providing a way to earn interest.

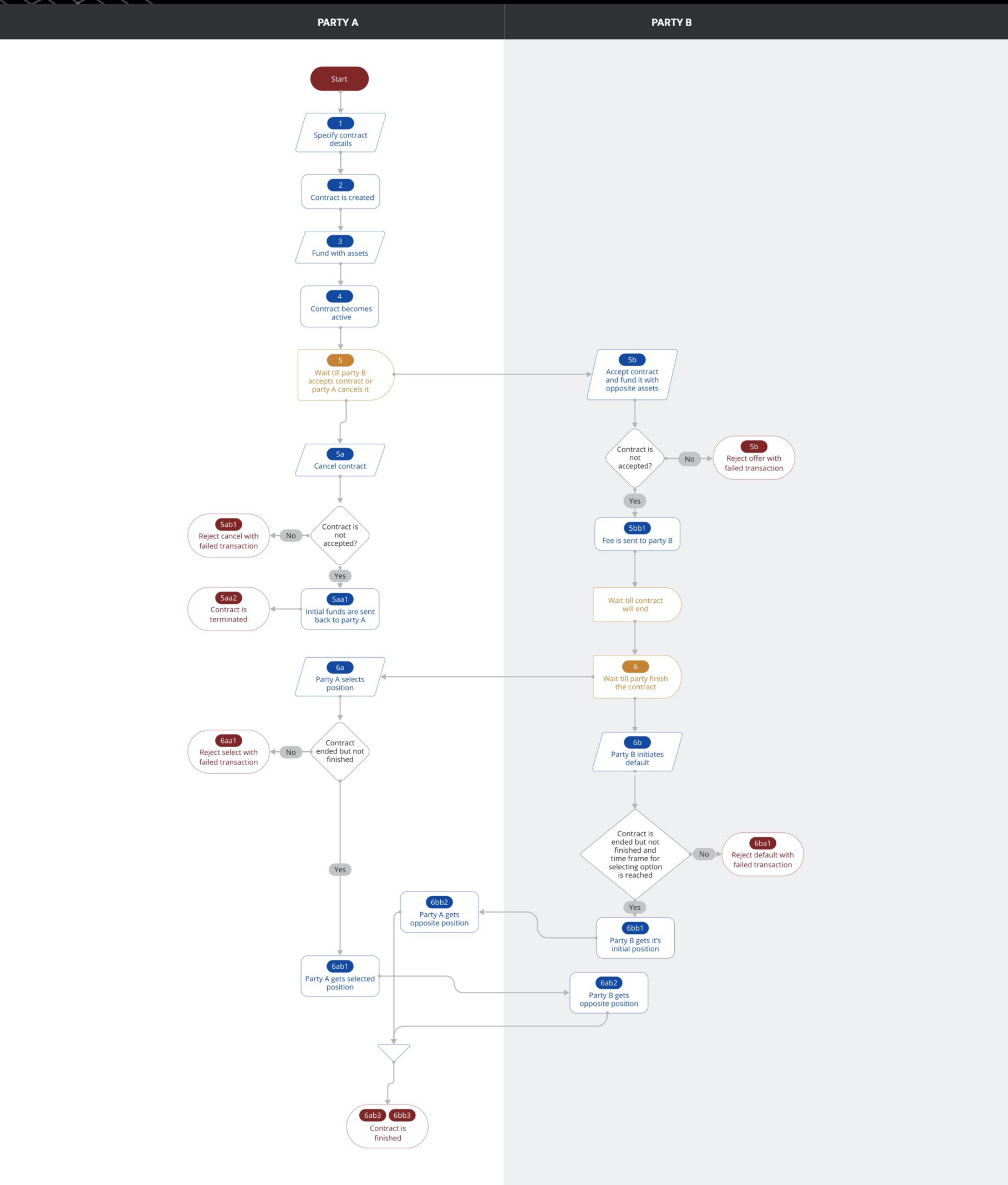
At openhedge, our mission is to contribute towards the advancement of decentralised open source finance, by proposing a new DeFi primitive aimed at further market infrastructure maturation for risk management. More specifically, we propose the openhedge smart contract standard for structured finance product creation, that is the next evolutionary step after lending primitives.

The openhedge smart contract standard enables replication of option payoff structures across any 2 assets on the ethereum network for both long and short positions, without the use of any intermediate voucher tokens. Openhedge will expand this architecture to other protocols in the future.

In this whitepaper we outline the financial and technical mechanics behind the concept, and demonstrate conditions under which any rational agent will engage in either long or short openhedge contract.

Openhedge is an independently governed non-profit organisation, funded by Proswap.com. Subject to reaching systemic importance, openhedge stands ready to implement a decentralised governance model, should that facilitate accelerated advancement of DeFi.

Technical mechanics



Workflow:

1. Party A specifies contract params and initiates contact creation. (Signs transaction with private key)
2. Contract is created. (Data is written to the blockchain)
3. Party A funds contract with tokens. Those tokens involve fee paid to party B. (Signs transaction with private key and this action involves asset sending in it too)
4. Contract becomes active. (Data is written to blockchain, funds arrive to contract)
5. When contract is active, it can be funded by party B or canceled by party A so we wait for one of these to happen.
   A. Party A initiate's contract cancelation (Signs transaction with private key)

      Contract checks if it's active and is not accepted

      A. Yes scenario
         1. initial funds are sent back
         2. **✓ Contract becomes terminated (finished) (Data is written to blockchain)**

      B. No scenario
         1. Blockchain transaction is rejected (Nothing happens on contract)

   B. Party B accept contract and fund it with opposite assets (Signs transaction with private key and this action involves asset sending in it too)

      Contract checks if it's active and is not accepted

      A. No scenario
         1. Blockchain transaction is rejected (Nothing happens on contract and funds are not moved to contract)

      B. Yes scenario
         1. Fee is sent to party B (on same time party B assets are moved to smart contract and Fee assets goes to party B wallet)

6. When contract is ended party A has select it's position and if party A doesn't do that in 24h party B can initiate default (24h time frame in future can be configured on contract creation)
    A. Party A selects position (Signs transaction with private key)

        Contract checks if it's ended but not finished

        A. No scenario
            1. Blockchain transaction is rejected (Nothing happens on contract)
        B. Yes scenario
            1. Party A gets selected position (asset sent to party A wallet)
            2. Party B gets opposite position (asset sent to party B wallet)
            3. **✓ Contract becomes finished (Data is written to blockchain)**
    B. Party B initiates default (Signs transaction with private key)

        Contract checks if it's ended but not finished and timeframe not allowing to initiate default has passed

        A. No scenario
            1. Blockchain transaction is rejected (Nothing happens on contract)
        B. Yes scenario
            1. Party B gets its initial position (asset sent to party B wallet)
            2. Party A gets its initial position (asset sent to party A wallet)
            3. **✓ Contract becomes finished (Data is written to blockchain)**

Smart contract input parameters:

1. Tokens that are going to be hedged - underlying asset
2. Period for which tokes are going to be hedged - maturity
3. Fee in % that is going to be paid to other party who will fill the hedge
4. Tokens that other party need to fill - hedge asset
5. Price for token in opposite tokens
6. Time frame in which party will have to select part after contract finishes before allowing other party to default contract (in future) - settlement period (in v1 release this will be set to default)

Smart contract will allow you to:

While contract is not filled by other party it can be canceled by contact creator

Once contract is filled filling party instantly gets fee

If contract is filled it will not accept any more parties to fill the hedge (in initial release, hedge asset can be provided from just a single transaction - no pooled hedging)

After contract reaches maturity, initiating party can choose which asset they recover - their initial asset or the hedge asset.

If after contract is finished, the initiating party hasn't selected a position, other party can initiate contract closing and receive back their original assets.

Financial mechanics

Buyer payoff structure

Let **S** = price of underlying asset, with respect to the hedge asset. (example: price of ETH denominated in USDT)

S_0 = current price of underlying asset, with respect to the hedge asset

T_0 = time of hedge contract initiation

T_m = time of hedge contract maturity

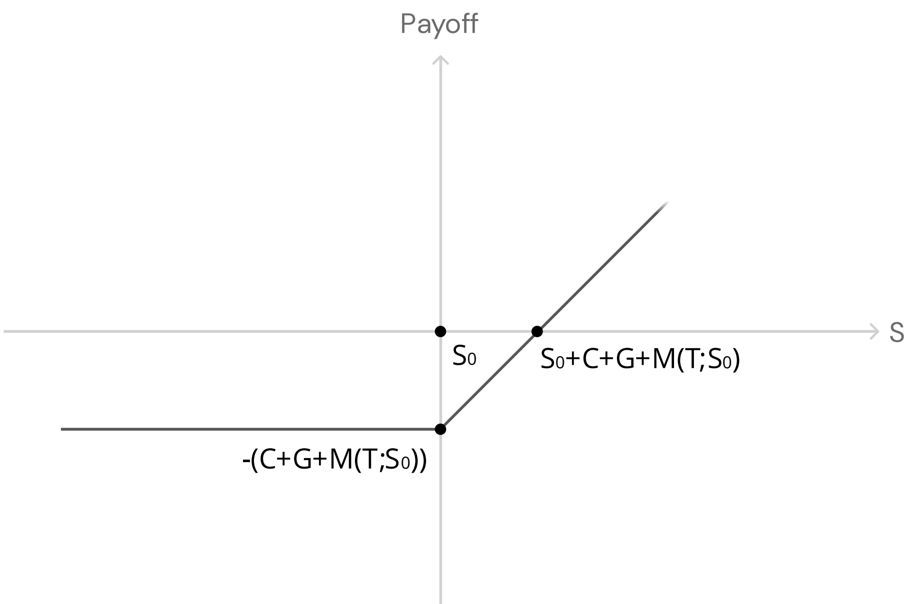
T = T_m-T_0

S_0 = price of underlying asset at T_0

C = fee paid for position creation (premium)

G = Network fees paid per transaction

M(T ; S) = **M((T_m-T_0); S_0)** = opportunity time value cost of S (for simplicity, we disregard the discussion about availability of risk free interest rates and associated transaction costs)



$$\textbf{Buyer payoff}\ (S_0;S;C;G;M(T;S)) = \begin{cases} -(C+G+M(T;S_0)) & \text{for } S\in[0;S_0] \\ -(C+G+M(T;S_0))+(S-S_0) & \text{for } S\in(S_0;S_0+C+G+M(T;S_0)] \\ S-(S_0+C+G+M(T;S_0)) & \text{for } S\in((S_0+C+G+M(T;S_0);+\infty) \end{cases}$$

Seller payoff structure

Let **S** = price of underlying asset, with respect to the hedge asset. (example: price of ETH denominated in USDT)

$\mathbf{S_0}$ = current price of underlying asset, with respect to the hedge asset

$\mathbf{T_0}$ = time of hedge contract initiation

$\mathbf{T_m}$ = time of hedge contract maturity

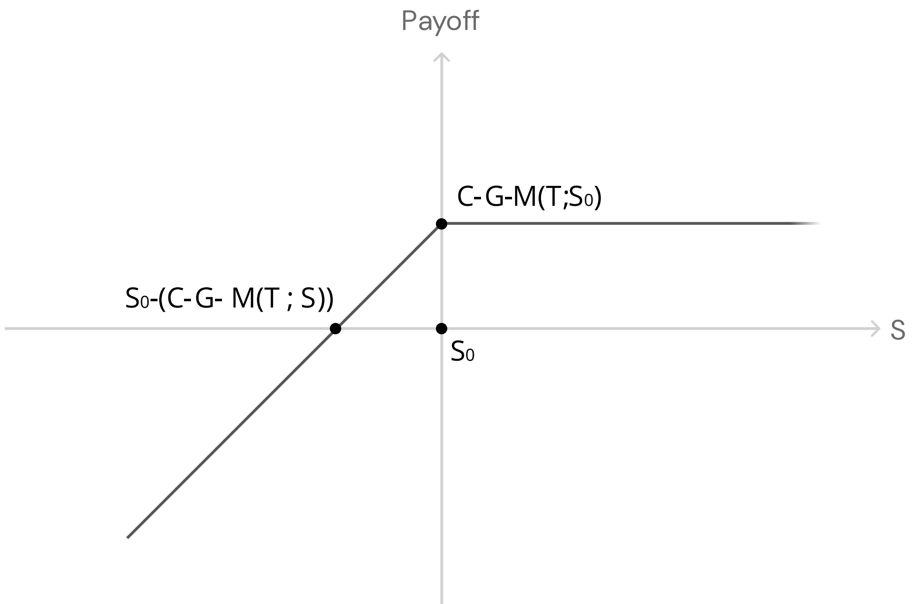
T = T_m-T_0

$\mathbf{S_0}$ = price of underlying asset at T_0

C = fee paid for position creation (premium)

G = Network fees paid per transaction

M(T ; S) = $\mathbf{M((T_m\text{-}T_0); S_0)}$ = opportunity time value cost of S (for simplicity, we disregard the discussion about availability of risk free interest rates and associated transaction costs)



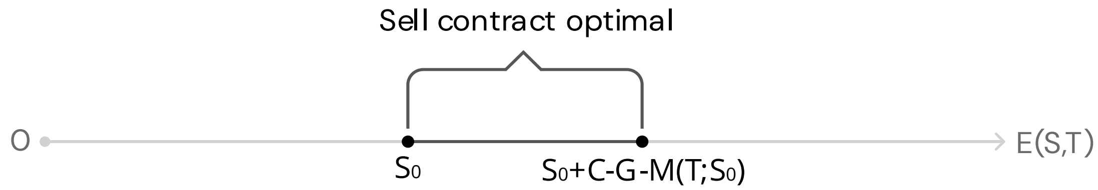
$$\textbf{Seller payoff}\ (S_0;S;C;G;M(T;S)) = \begin{cases} -S+(S_0-(C-G-M(T;S_0))) & \text{for } S\in[0;(S_0-(C-G-M(T;S_0))] \\ (C-G-M(T;S_0))-(S_0-S) & \text{for } S\in((S_0-(C-G-M(T;S_0));S_0] \\ C-G-M(T;S_0) & \text{for } S\in(S_0;+\infty) \end{cases}$$

Optimal conditions behind engaging in an openhedge contract

Let **S** = price of underlying asset, with respect to the hedge asset. (example: price of ETH denominated in USDT)
$\mathbf{S_0}$ = current price of underlying asset, with respect to the hedge asset
$\mathbf{T_0}$ = time of hedge contract initiation
$\mathbf{T_m}$ = time of hedge contract maturity
T = T_m-T_0
$\mathbf{S_0}$ = price of underlying asset at T_0
C = fee paid for position creation (premium)
G = Network fees paid per transaction
$\mathbf{M(T ; S)}$ = $\mathbf{M((T_m-T_0); S_0)}$ = opportunity time value cost of S (for simplicity, we disregard the discussion about availability of risk free interest rates and associated transaction costs)
E(S;T) = expected price of the underlying asset at the end of maturity period

To understand the relevance of the OpenHedge smart contract standard, we need to explore the conditions, under which the agent will find it rational to engage in either side of the position.

$Assumption_1$: Hedge asset price fixed for period T
$Assumption_2$: Standard utility function assumptions
$Assumption_3$: The asset universe is limited to 2 assets - underlying asset, hedge asset



Within a 2 asset universe, it is rather straightforward to see that if $S_0 < E(S,T) < (S_0 + C - G - M(T; S_0))$, the optimal strategy for a rational agent is to sell a hedge position and earn the associated premium minus the transaction costs.

While the selling conditions are straight forward, we need to look at the conditions under which buying a hedge is optimal, and this gets a bit more tricky. Under the simplified assumptions of price expectations alone, we can not demonstrate the case for buying a hedge being the optimal condition, so new dimensions must be added to the model. One of the following statements must be true, in order for there to exist a range of values for E(S,T), such that buying a hedge becomes the optimal strategy:

- There are equilibrium altering transaction costs associated with underlying asset liquidation
- The agent perceives likelihood of $S > (S_0 + C + G + M(T; S_0))$ at any $T \geq T_m$, to be of a level, s.t. the cost of hedge is less or equal to the value of continued exposure to the underlying asset, according to the model of their risk preferences and expectations

Either or both of the above two conditions can be true in order to demonstrate that the optimal action for the agent is to buy a hedge should they expect $S < (S_0 - C - G - M(T; S_0))$. There are many other possible considerations, not demonstrated above, that would yield the same conclusions, but are left out from this basic model.