Overview of The Proposed DeFi Staking & Lending Protocol(codename Solar Finance, SFX):

LENDING:

- P2P lending on Solar Blockchain
- Relies on collateralized loans and a lending pool

Lenders deposit funds into the DeFi network, and borrowers can access the funds. Borrowers are required to lock an amount of collateral of a supported asset(SXP for a start, and SXP20 tokens when they launch). The valuation of the collateral is based on USD with prices fetched via an oracle network or determined using an AMM.

The protocol will have some of the funds in the lending pool go towards reserves to serve as insurance to lenders, when they need to retrieve funds from liquidity pools.

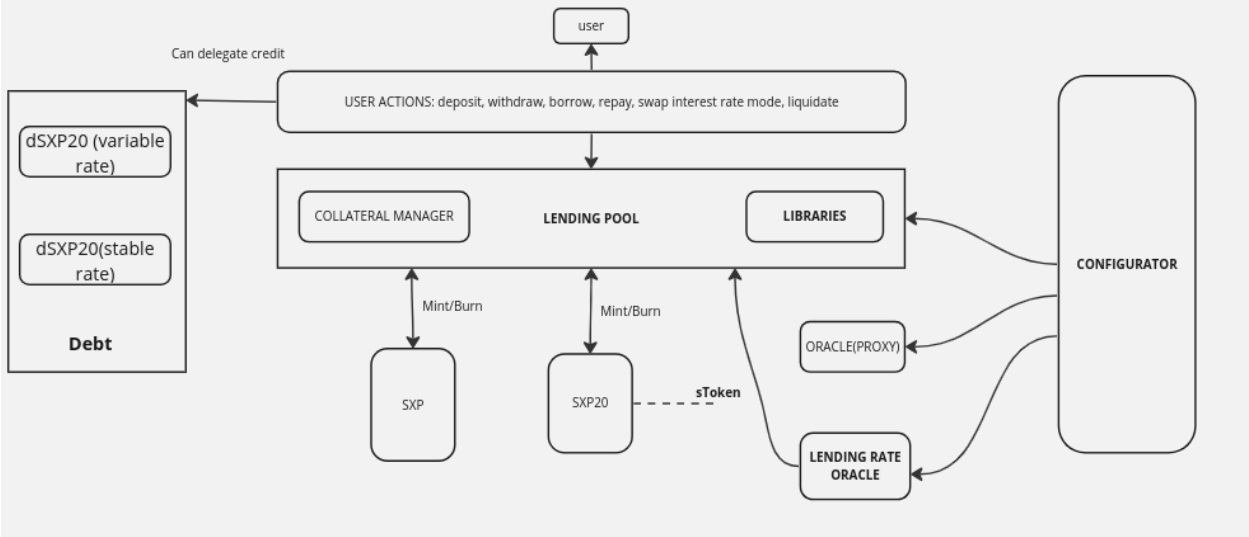
Interest rates for the loans are either stable or variable and users can switch between either rate. Stable rates are to be based on an interest average for the asset over a length of time e.g 30 days. The length of time is adjustable as one of the protocol's parameters.

When the user deposits in a lending pool, they receive an interest-bearing token whose value is pegged 1:1 to the value of the underlying asset [sToken]. The token is transferable, and can be redeemed for the underlying asset + accrued interest at time of withdrawal.

Interest rates are determined based on the amount of money available in the pool. The more funds are borrowed from a pool, the less amount is available for lending which raises the interest rate.

Proposed Protocol architecture:

AAVE v2 is proposed as the model architecture Solar Finance is to emulate.



LENDING POOL: most of the user interactions with the protocol will happen through the lending pool as indicated in the diagram.

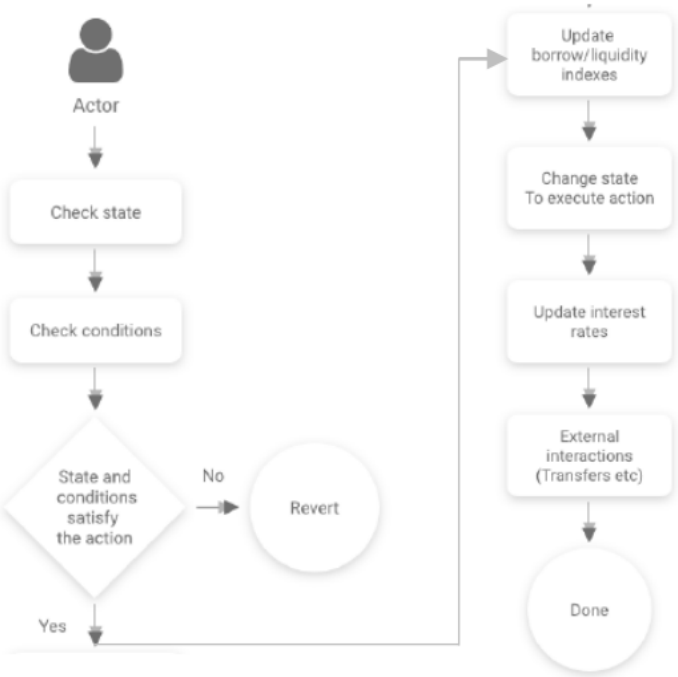
sTOKENS: when a lender deposits an asset into the lending pool, a corresponding amount of s[Token]s is minted and deposited into their wallet. These tokens are interest bearing and will implement most of the ERC20 token methods with modifications such as scaled balance, scaled total supply and scaled user balance.

DEBT TOKENS: these are tokenized debt positions used across the protocol. They will be implemented as transferable so that it is possible to delegate credit to a cold storage for proper debt management.

ORACLE: asset prices will be fetched using an oracle, to safeguard the security of deposits and prevent bad debts. Asset pricing will enable determination of the proper collateral ratio, liquidation threshold and so on.

CONFIGURATOR: various libraries will be used such as upgradability, logic, and math libraries.

The lending Pool Process:



1. **Deposit:** User deposits an amount of an asset into the protocol, minting the same amount of corresponding sTokens and transferring them to the caller's address.
2. **Withdraw:** user withdraws an amount of the underlying asset and burns the corresponding sToken.
3. **Borrow:** this transfers to the borrower a specific amount of an underlying asset, while the user transfers an amount of collateral that remains locked. The two values are as per the collateral ratio.
4. **Repay:** the user can partially or completely repay the borrowed amount, inclusive of the accrued interest.
5. **Swap interest rate mode:** the user can choose to use either the stable or variable interest rate mode.
6. **Liquidation Call:** when the collateral to loan ratio is above liquidation threshold, the loan is flagged for liquidation. Liquidators repay a percentage of the outstanding amount borrowed on behalf of the borrower, while receiving a liquidation bonus (a discounted amount of the collateral). The bonus is subtracted from the borrower's collateral as a liquidation fee. As such, the max collateral to loan ratio is kept at a healthy percentage and below 1 to cover for the risk of liquidation. Each lending pool will have independent parameters for this, dependent on the prices and amounts of assets available.

7. **Delegate Credit:** Borrowers can transfer their credit to another address by transferring the respective debt token.

Tokenization:

sTokens: these are interest-bearing tokens, minted and burned upon deposit and withdrawal respectively. The value is pegged to the corresponding asset at a 1:1. Interests collected by the

sToken holders are distributed to their wallets directly by increasing the wallet balance. The interest rate model can be either variable or stable. The minimum length of time before withdrawal can be determined as one of the lending pool's parameters.

User scaled Balance: the user balance is algorithmically stored together with the user index as a ratio, the scaled balance ScB. Every action leading to the minting or burning of sTokens automatically updates the balance of the user via calculation. This is derived from the aave tokenization model and strategy.

When a user deposits an m amount in the protocol, the scaled balance updates as per the algorithm:

$$ScB_t(x) = ScB_{t-1}(x) + \frac{m}{NI_t}$$

Where NIt is the reserve normalized income.

When the user makes a withdrawal the balance is updated as per the algorithm:

$$ScB_t(x) = ScB_{t-1}(x) - \frac{m}{NI_t}$$

The sToken balance of a user at any time can be derived as:

$$aB_t(x) = ScB_t(x)NI_t$$

The details for the mathematical implementation of the lending pool tokenization strategy can be found here(aave v2 whitepaper). The proposed algorithms can be revised to give a more suitable tokenization model for the solar blockchain.

Debt tokenization: these are interest-accruing tokens minted and burned on borrow and repay, and are indicative of the debt owed by the token holder. They are of two types:

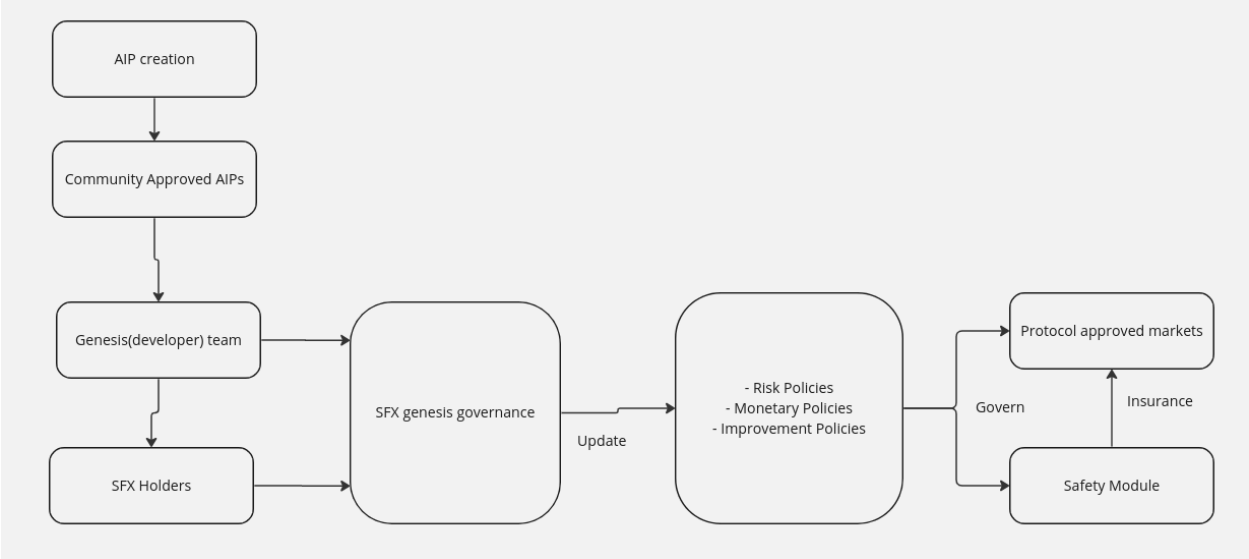
- **Stable debt tokens:** represent a debt to the protocol with a stable interest rate.
- **Variable debt tokens:** represent a debt to the protocol with a variable interest rate.

These tokens implement the ERC20 token standard.

Governance:
Governance will provide policies on key parameters which are to be determined in the protocol. Protocol policies will govern the overall behavior of the protocol and its entities such as safety, expansion to accommodate new assets and overall economics. Market policies would be defined in the context of each market within the ecosystem. Market policies would have to confine to the limits of the protocol policies.

Proposed governance architecture at launch:
Protocol Governance:

Market Specific Governance:



Holders of the SFX(Solar Finance) token will have the ability to vote on proposals and collectively act as governors of the protocol. The AIPs are on-chain governance proposal that will automatically execute once the votes are cast and validated.

AIPs can be created by any community actor that meets set requirements. These can be proposals to add a new token to the protocol, update risk parameters or add a new liquidity market. This enhances decentralization of the protocol.

SFX holders can also delegate their proposal and voting powers, making governance scalable.

Protocol Policies include: risk policies, improvement policies, incentives policies, safety incentives, ecosystem incentives, staking incentives

Market Policies include: supported assets to provide liquidity and borrow from, supported assets to use as collateral, enable/disable borrowing modes of an asset, market-specific component updates, risk configuration per asset, interest rate models per asset

STAKING

The staking module will allow users to stake the network while maintaining the utility of their capital. When a user stakes SXP, the SFX token is minted and transferred to their wallet, which will maintain a 1:1 peg of the underlying SXP token. The SFX token will implement the ERC20 utility token standard, unlocking the user's staked capital value. The SFX token will further accrue the staking rewards or penalties associated with the respective validator. Upon withdrawal of the staked SXP the user burns the corresponding amount of SFX and receives the initial amount, alongside with the amount earned in rewards or penalties.

The main components of the SFX staking protocol will thus comprise:

1. **Staking Pool:** protocol that will manage deposits, staking rewards and withdrawals. The staking pool will consist of:
    - Validator registry
    - Withdrawal credentials for security
    - Updates Oracle integration
    - Rewards integration module
2. **SFX:** liquid staking token that will maintain a 1:1 peg of the deposited SXP native token.

Staking Pool: this will be the core responsible for:

- SXP deposits and withdrawals
- Delegation of funds to validators and block producers
- Allocating a fee to staking rewards
- Receiving updates from the oracle

Node validators will be selected as part of the protocol governance measures. This function can be separated from the main protocol in future updates.

The validators will provide their keys to the main smart contract, via which funds would be allocated to them. The logic for distributing funds will be contained in this smart contract.