Intro

The following pages demonstrate the borrow/lending process end-to-end, starting with the lender depositing funds and ending with the borrower acquiring those funds.

Once this process is understood, the remaining processes (repayments, liquidations, flash loans) are trivial.

Agenda:
1. We start by looking at various state & oracle accounts that oversee borrow/lending and are needed for the protocol to function.
2. We then go through the actual borrow/lending process step by step.
3. We provide a snapshot of the entire process as a single page summary.

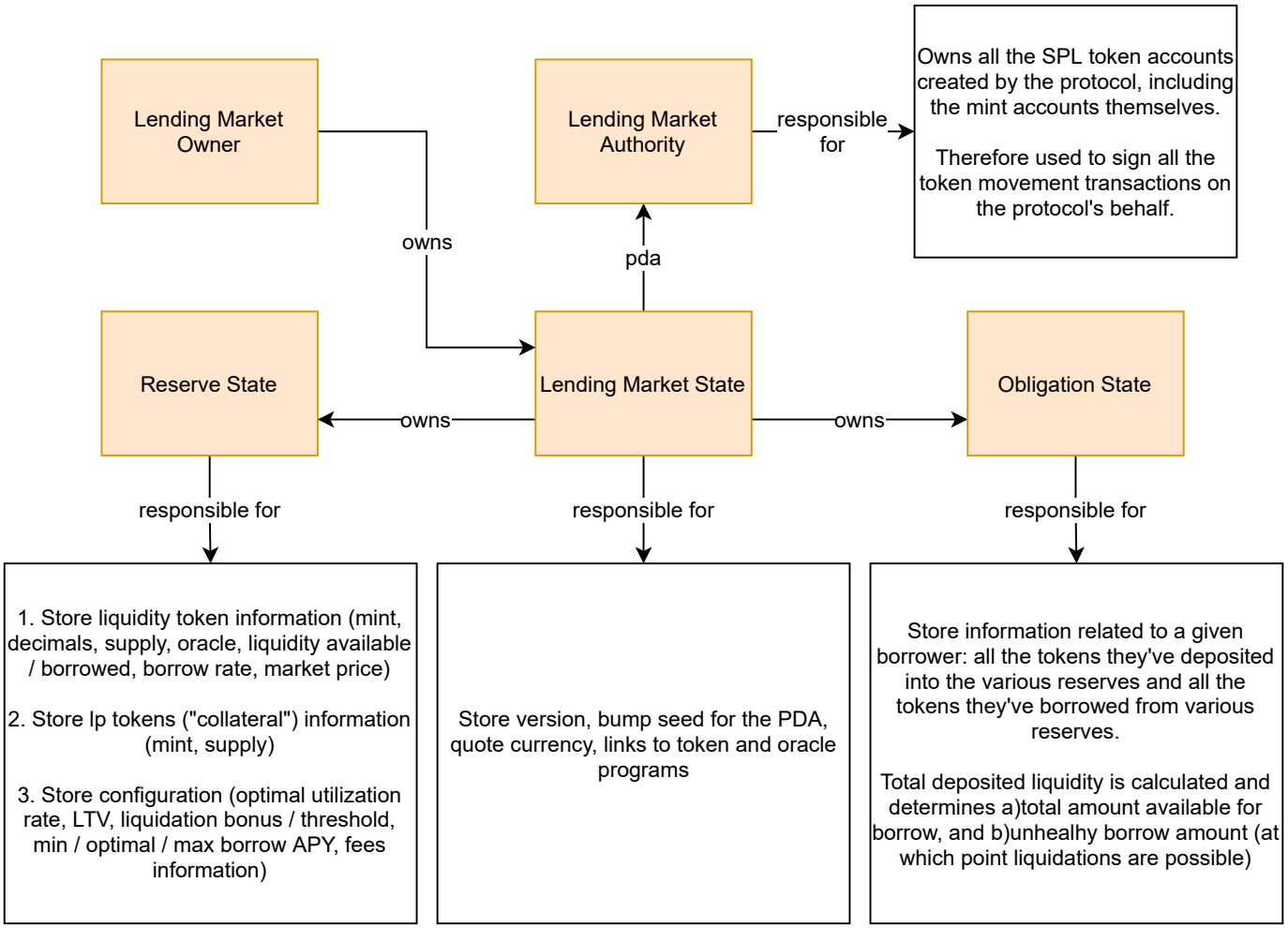
Throughout the manual 2 types of tokens are mentioned:
1. Liquidity tokens = tokens actually deposited / borrowed
2. Collateral tokens = think of them as "LP" tokens minted in return for depositing liquidity

0. State / oracle accounts

2 types of peripheral (non token holding) accounts are involved in borrow/lending:
1. State accounts
2. Oracle accounts

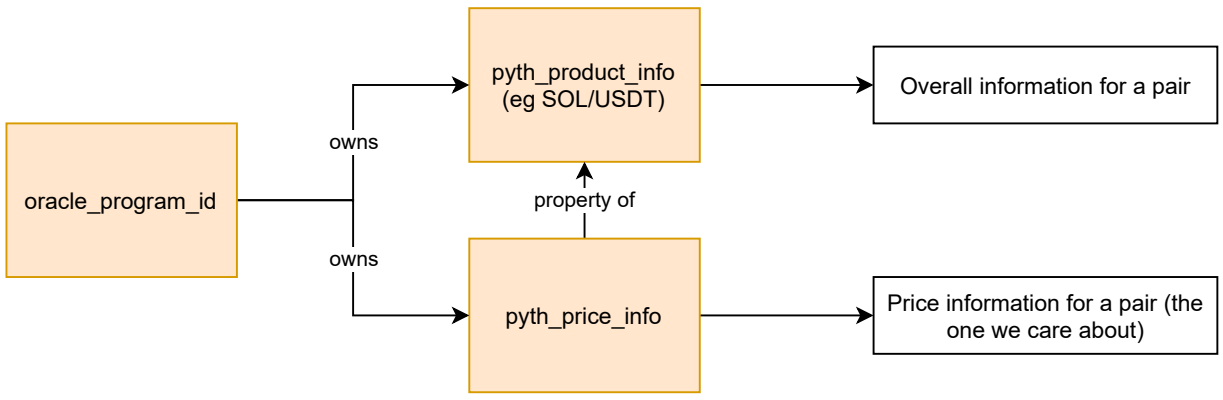
State accounts, like the name suggests, hold the state of the protocol. This includes the state of the protocol as a whole (Lending Market State), the state of each specific deposit pool, such as SOL or USDC (Reserve State) and the state of each individual borrower (Obligation State).

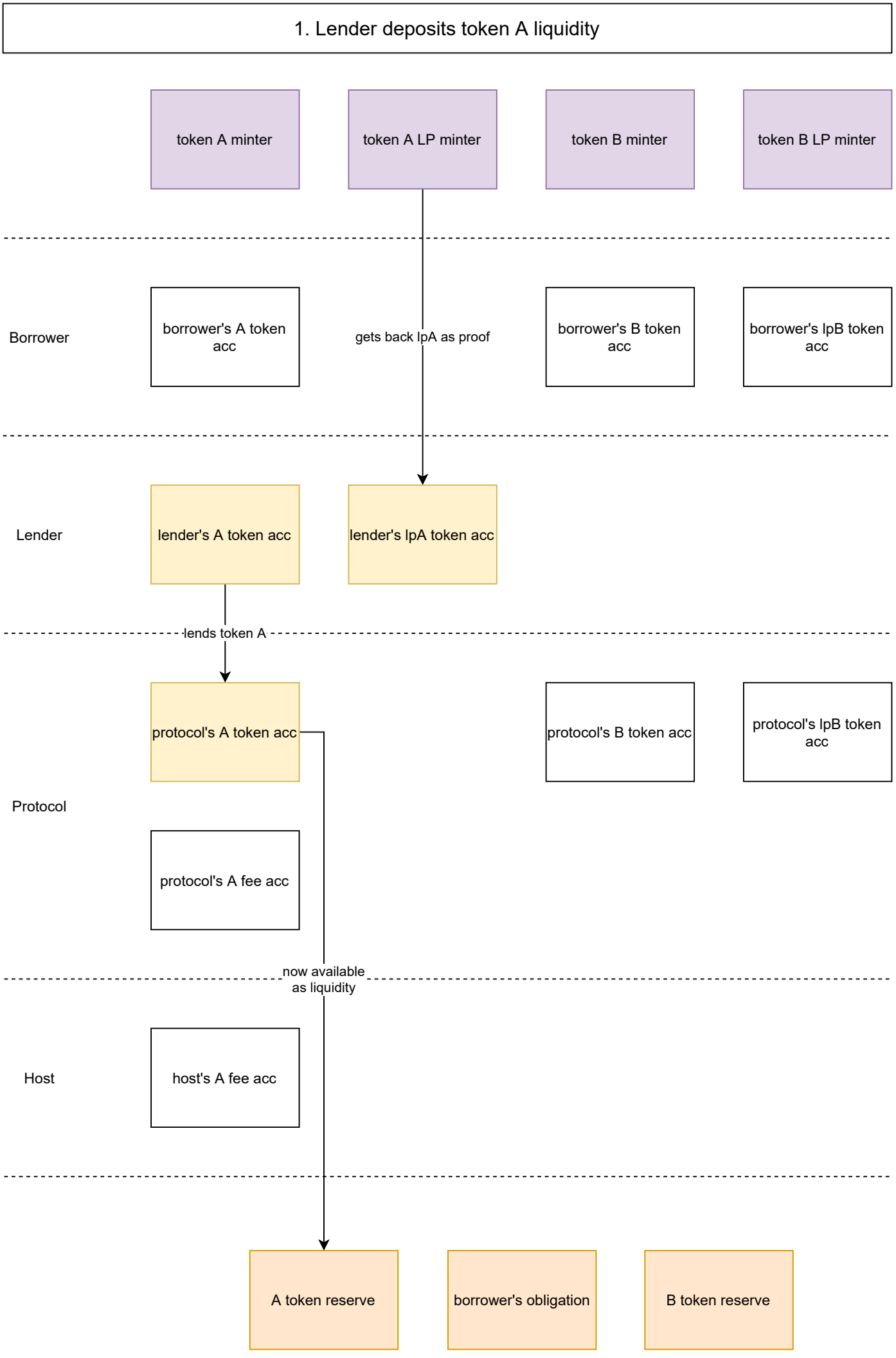
The relationship between the 3, and functions performed by each are shown below:

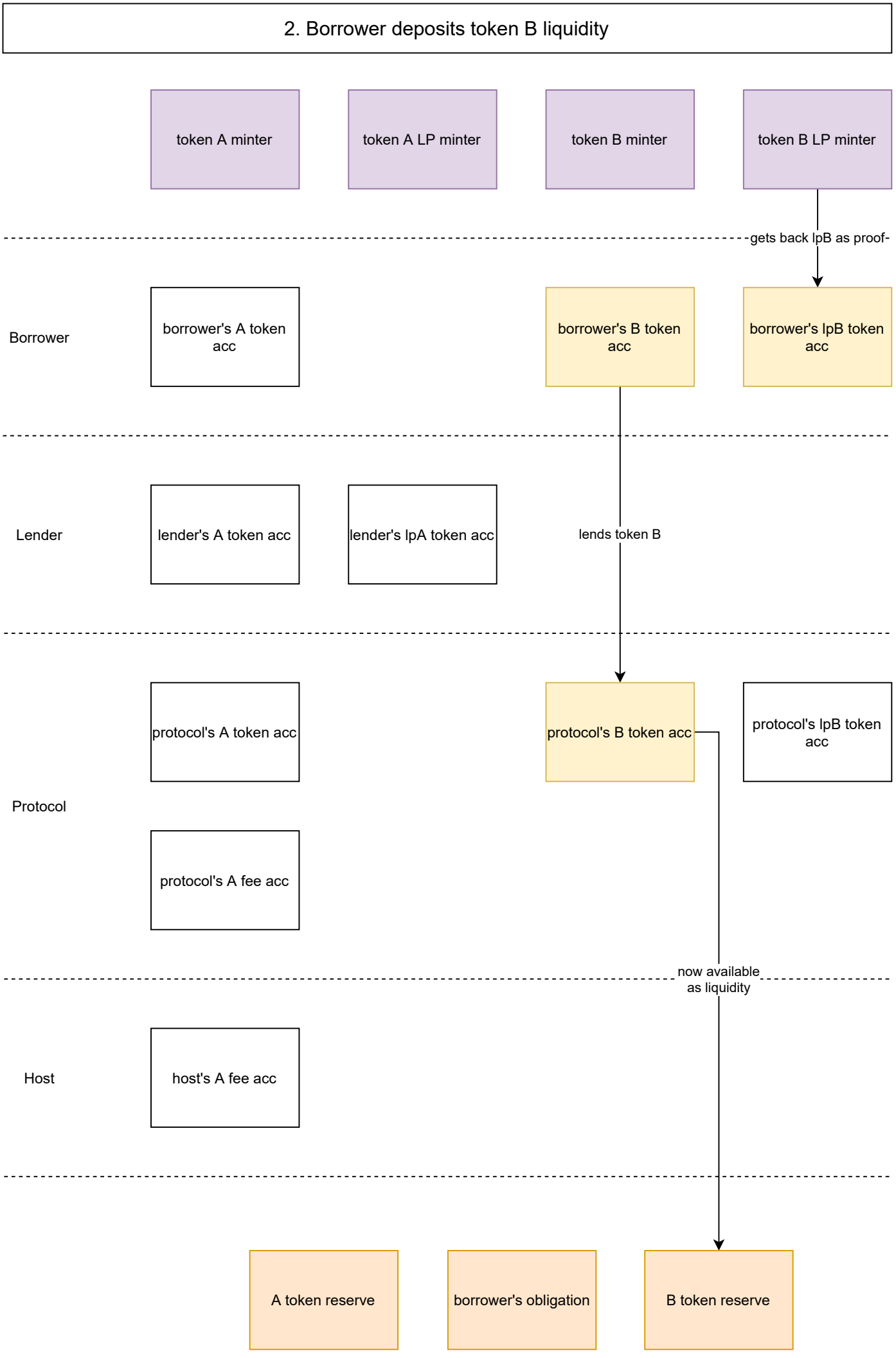


Next there are oracle accounts. Like the name suggests, they are responsible for providing price information for the protocol to decide on how much an individual user can borrow (given their deposits), and where their liquidation happens. Oracles are provided by Pyth network. More here.

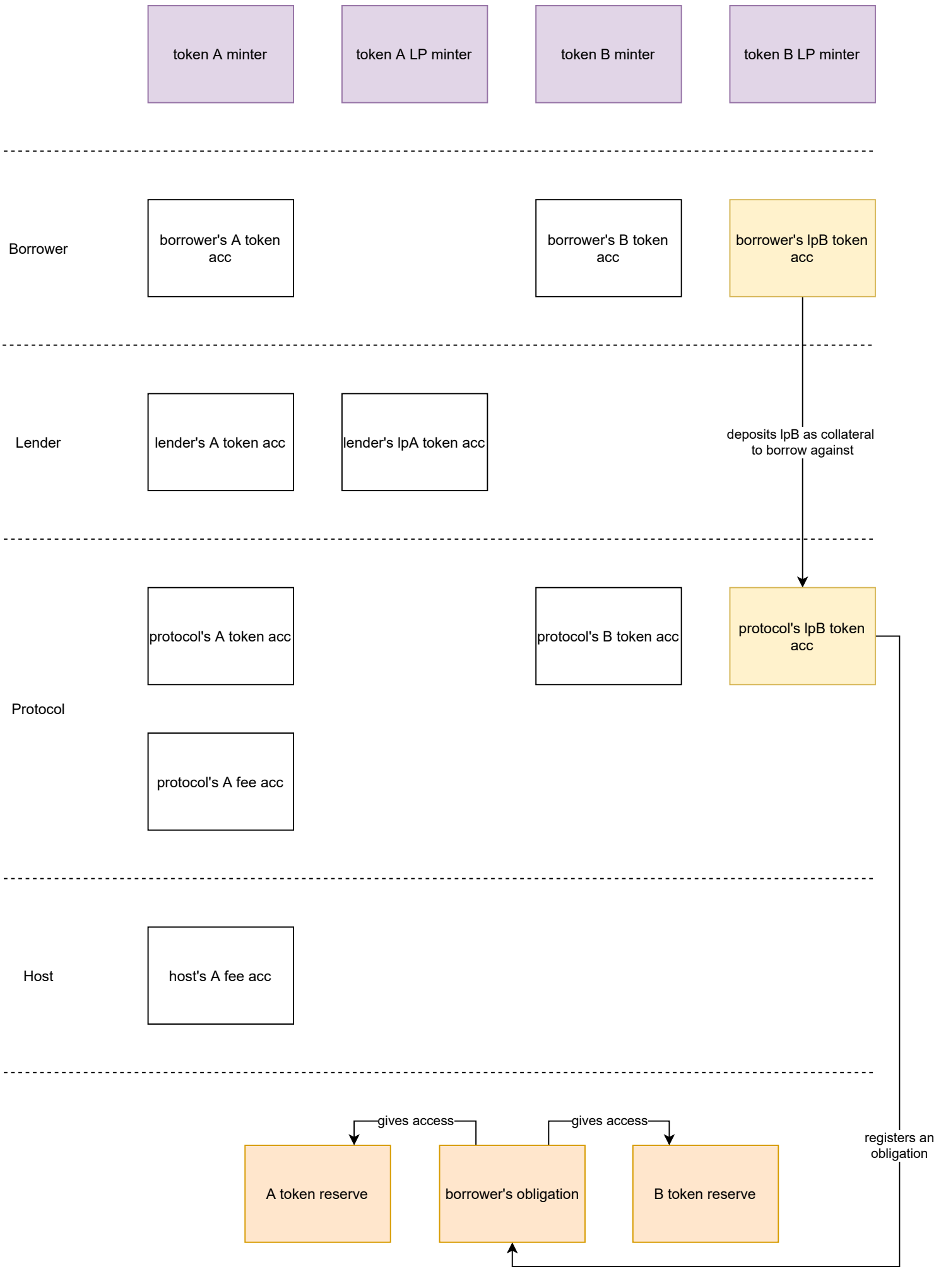
The rough architecture is as follows:



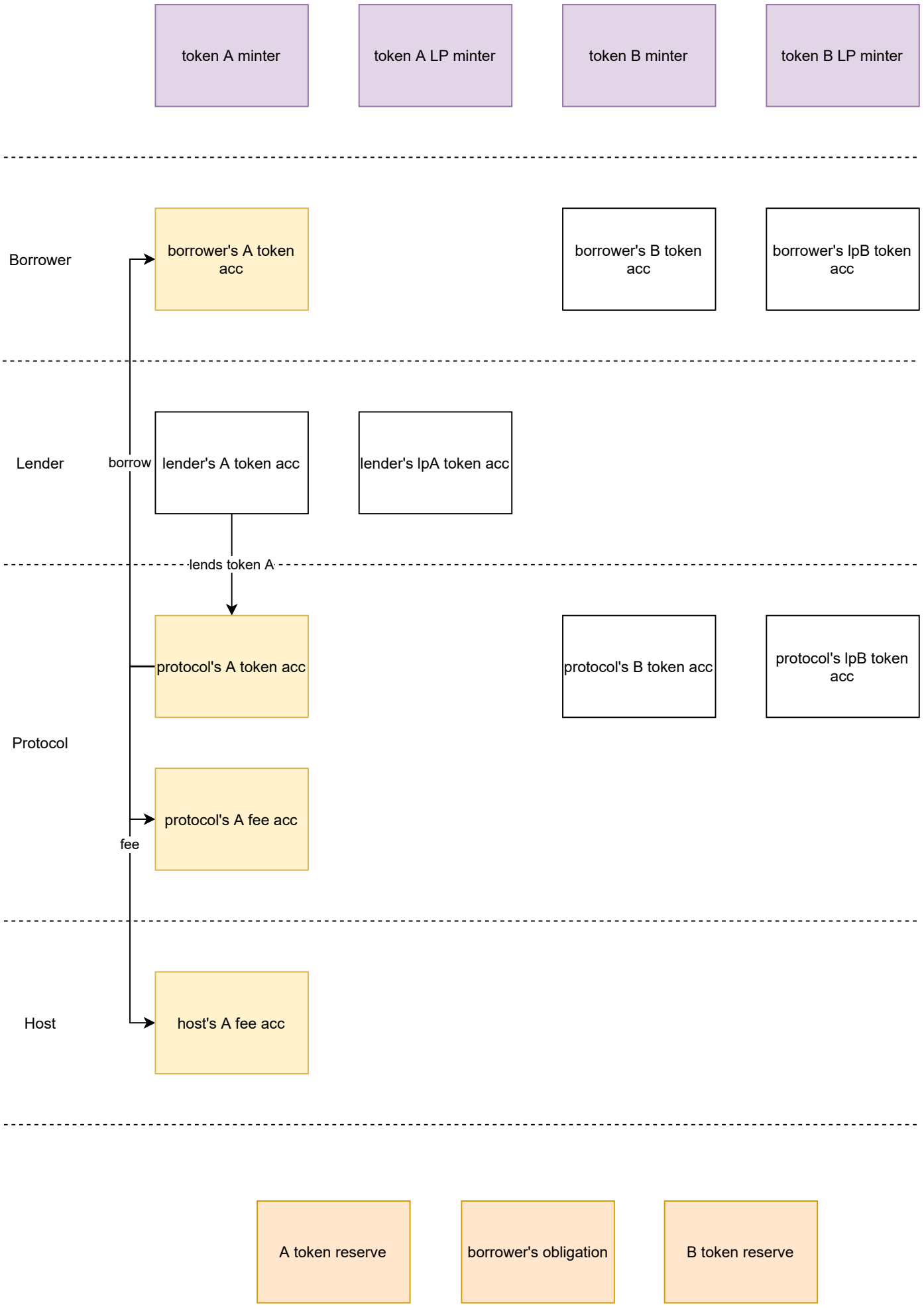

1. Lender deposits token A liquidity
token A minter
token A LP minter
token B minter
token B LP minter
Borrower
borrower's A token acc
gets back lpA as proof
borrower's B token acc
borrower's lpB token acc
Lender
lender's A token acc
lender's lpA token acc
lends token A
protocol's A token acc
protocol's B token acc
protocol's lpB token acc
Protocol
protocol's A fee acc
now available as liquidity
Host
host's A fee acc
A token reserve
borrower's obligation
B token reserve

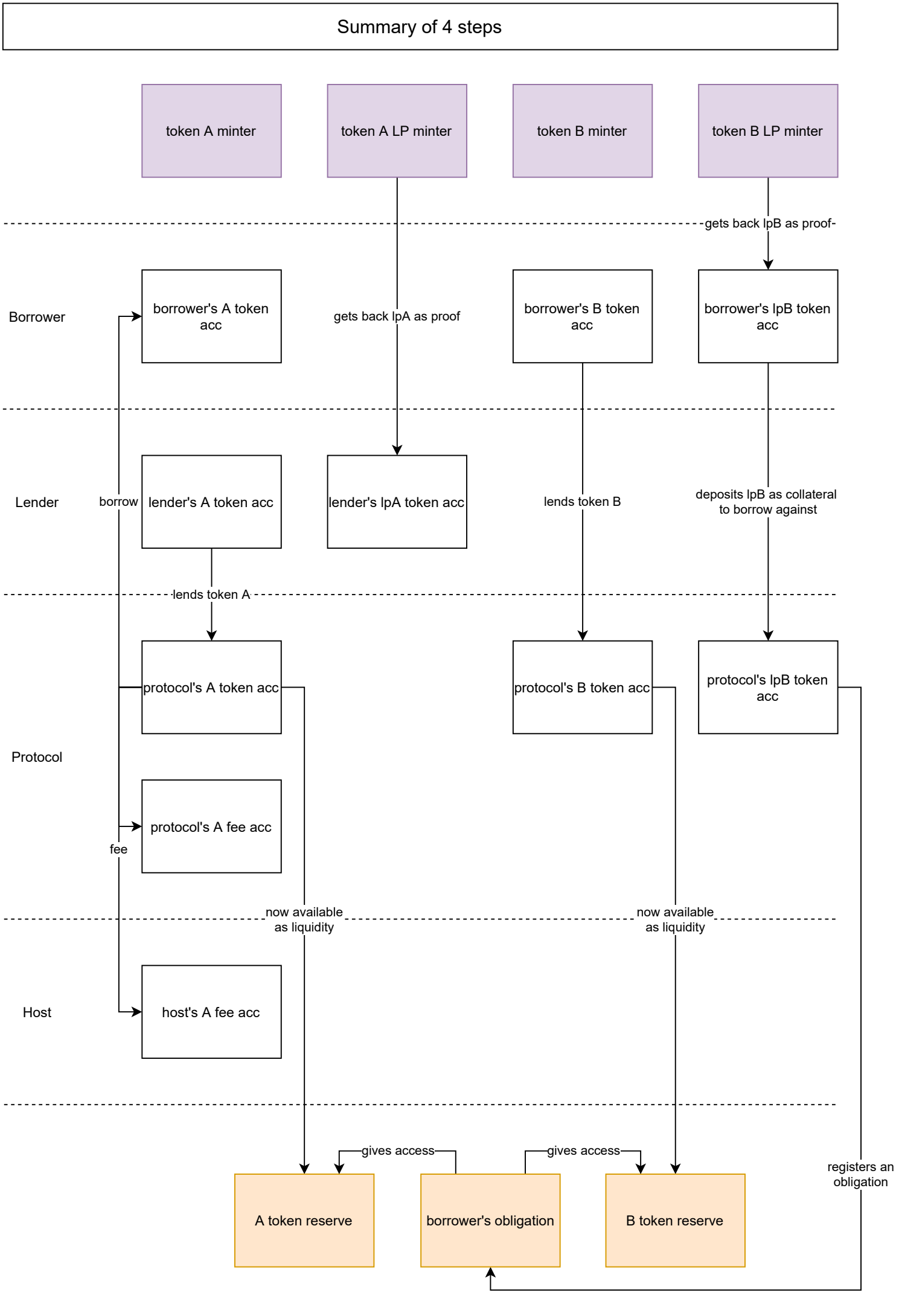

2. Borrower deposits token B liquidity
token A minter
token A LP minter
token B minter
token B LP minter
gets back lpB as proof
Borrower
borrower's A token acc
borrower's B token acc
borrower's lpB token acc
Lender
lender's A token acc
lender's lpA token acc
lends token B
Protocol
protocol's A token acc
protocol's B token acc
protocol's lpB token acc
protocol's A fee acc
now available as liquidity
Host
host's A fee acc
A token reserve
borrower's obligation
B token reserve

3. Borrower deposits bLP tokens as collateral



4. Borrower withdraws token A liquidity against their bLP




Summary of 4 steps
token A minter
token A LP minter
token B minter
token B LP minter
gets back lpB as proof
Borrower
borrower's A token acc
gets back lpA as proof
borrower's B token acc
borrower's lpB token acc
Lender
borrow
lender's A token acc
lender's lpA token acc
lends token B
deposits lpB as collateral to borrow against
lends token A
protocol's A token acc
protocol's B token acc
protocol's lpB token acc
Protocol
protocol's A fee acc
fee
now available as liquidity
now available as liquidity
Host
host's A fee acc
gives access
gives access
registers an obligation
A token reserve
borrower's obligation
B token reserve