{{PegLoan}}: A trust-minimized collateral-backed stable-coin system

Technical research whitepaper

> Notice: This document is a research proposal, and in no way makes promises about the full functionality of any derived implementation, nor does it represent a commitment to build, deploy or operate such a system. Its sole aims is to envision the process of defining, planning and productizing such a system and product.

This research proposal describes a **collateral-backed stable-coin** system implemented on a public blockchain, where the exclusive backing collateral is the **native blockchain asset token**, and where the stable-coin is pegged to the value of a widely accepted **reference currency**.

A collateral backed stable-coin system ensures the stable value of a digital token, through securing backing collateral of equal or higher value, and by balancing token supply and demand through adjusting monetary variables using market pricing information. Variations of such a system, such as MakerDAO's DAI token, have already been implemented on public blockchain smart contracts. In case of our *Minimum Viable Product (MVP) implemented on Ethereum*, the *native blockchain asset token* is **ETH** and the *reference currency* is **USD ($)**. Unlike other similar systems however, the proposed system contains *no on-chain governance process* and *no tokens for governance or equity*. The proposal posits that eliminating these, in favor of an on-chain incentive system, **reduces centralization**, and **increases the capital efficiency** of the system.

Actors

- **Money users** - These are *regular consumers* that *use*, or *store* the pegged currency money in their blockchain-based *savings account*. The major target groups that could most benefit from such an offering are *the un-banked* and those with limited access to a *stable Store of Value* (SoV) from countries with dysfunctional monetary policies. This could also be used by the *early adopter community* of a public blockchain.
- **Loan takers** - Most commonly existing *native token holders* that decide to *take a loan* against their holdings. The main incentive for this group consists of being *long native token*, while being able to *deploy its value*. They may also initially be *motivated by the technology* itself, however this is not a long-term sustainable incentive.
- **Price feed providers** - Providers of accurate rates of exchange between native token, the reference asset as well as the pegged currency. These providers are expected to compete in establishing trust amongst the loan takers, and in return be monetarily compensated based on the level of trust they establish, and based on following the hard rules set by the loan system.
- **Loan liquidators** - Liquidators *compete* with each other to monitor, speculate on the liquidation status of loans (debt positions), and *trigger liquidation* when they foresee *liquidation conditions* in accordance with the loan system's hard definition of liquidation.

Tokens and currencies

The system supports pegging to any relatively stable real world currency, as well as other relatively stable baskets of assets, always backed with the most trust-minimized blockchain collateral (ETH in case of Ethereum). The main viable implementation of this system however will focus on the US dollar due to its relative ubiquity at time of this writing.

Pegged currency money

The pegged currency money is the product that is ultimate offered to everyday digital money users and holders. It will have a stable value, the unit of which, should already be (or will be) widely accepted and used in everyday commerce transactions by buyers and merchants. The Ethereum based MVP implementation of this system will use a custom ERC20 that represents the value of the US Dollar ($).

The token requires the base functionality already available in common programmable digital tokens, such as basic transfer functionality between accounts. The Ethereum community's version of this functionality is described by the ERC20 standard also described here;

Additional functionality will be implemented to respectively "mint" or "burn" tokens upon "issuance" or "return" of the pegged currency on the loan contract.

Reference currency

This is the real world currency, or unit of account, the pegged currency will be pegged to. For example, the US dollar ($) is the most commonly used reference currency in most existing stable-coin implementations.

The system is designed such that it would work with any relatively stable currency, or basket of goods, assets and/or currencies. Any user can create a new currency peg to their target currency reference, by deploying the open source contracts including a custom token representing their pegged currency.

Backing asset token

As mentioned, any public blockchain's native asset token satisfies the role of backing asset given that it is likely to be:

- The most trust-minimized asset token on the blockchain.
- Widely used as a Store of Value (SoV) by the blockchain users, and early adopters.

It is essential for the system to prevent significant issuance of pegged currency, without securing a corresponding value of backing asset, per loan instance, and for the system as a whole. Similarly, it is crucial for the system to respond to backing asset release requests, when corresponding pegged currency is being returned to the system. In a healthy system, the value of the backing asset token is greater than the pegged currency by a healthy margin, in order to insure against the possibility of a major devaluation in that backing asset.

Loan (debt position)

Loans (aka debt positions) are the logical structures that hold the state and value of the collateral backing aspect of the system. They have the following notable properties:

- There is a separate loan contract instance per instance of real world loan taken
- The loan contract holds the state and value related to each loan
- The loan taker is the ultimate owner of the value contained on a loan instance, and directly interacts with it.
- The greater system is consulted every time a change is made to the loan. Changes disallowed by the main system, result in automatically reverted transactions.

- A specific threshold of backing to issuance is considered healthy and enforced by the system. For example at one point in time: `backing / issuance = 150%`.

Loans consist of the following state or calculated values:

- Native token deposited into the contract
- Pegged currency issued by the loan taker
- Pegged currency stored in the loan
- Fees accumulated since last structural change (Calculated)
- Liquidation requests and state

Loan fee

Loan takers pay a fee based on the value of the loan's issued pegged currency, and based on a variable daily percent rate (See ***Loan fee rate*** under monetary variables) set by the main system's monetary policy engine. The fee has to be paid as part of structural changes made to the loan, including changes to allocation or issuance. For example, at a given point, the loan fee rate might be `4%` per year, (equivalent to `0.01096%` per day). Assuming no change to the rate (unlikely), the loan taker should expect to pay `$400` per year in fees on a loan that has issued `$10,000` worth of pegged currency, upon closing the loan. In reality, the yearly fee is determined by the sum of all daily rates throughout that year.

A part of the fees paid by loan takers, will go to the price feed providers. The specific value going to price feed providers is calculated based a weekly variable rate determined by the monetary engine (See ***price feed revenue rate*** under monetary variables). This value is allocated to specific price feed providers based on the allocation table values (See loan price feed allocation) of the loan.

The rest of the fees paid by loan takers, will go to the savings pool, to eventually be distributed to savings account owners based on the monetary engine's savings interest daily rate (See ***Savings interest rate*** under monetary variables).

Loan price feed allocation

Every loan (aka debt position) can allocate one or many price feeds providers by weight for their loan. For a given loan, these allocations are weighted by percentages that add up to `100%`. For example: a loan may allocate `40%` to the price feed maintained by ***Foundation X***, `35%` to ***Rating company Y*** and `25%` to ***Financial company Z*** for a total of `100%`.

A part of the fees paid by loan takers, will eventually go to the price feed providers as revenue. The specific value going to price feed providers is calculated based a weekly variable rate determined by the monetary engine (See ***price feed revenue rate*** under monetary variables). That value is distributed to specific price feed providers, based on the allocation table values of the loan. Loan takers are responsible for determining the allocation mentioned above, based on which price feed providers they trust and/or want to financially support.

At level of the entire system, we add up the total allocation values resulting from allocation percentages multiplied by the issuance value of each loan. For example consider a single loan with issuance value `$100,000`, that has three allocations A at `50%`, B at `35%`, C at `25%`. The single loan will contribute to the price feed providers' revenue pools by `$50,000` for A, `$35,000` for B and `$25,000` for C. If the whole system consisted of say `200` loans, that are same as the one above (very unlikely), the total allocation numbers for the price feeds will be `$10,000,000` for A, `$7,000,000` for B and `$5,000,000` for C. Based on a projected ***price feed revenue*** *rate* of say `1%` per year, the projected increase to each of their revenue pools would be `$100,000` for A, `$70,000` for B and `$50,000` for C.

From the perspective of reducing risk of system capture, it is ideal that there be more price feeds allocations, and that they be easily changed in case of bad behavior by price feeds. On the other hand however, an excessive number of allocations will cause excessive gas cost and cognitive overhead for the loan takers and other stakeholders. An average of 3 allocations seems like a good target, thus a max value of 5 is reasonable. See maximum price feed allocation.

Price feed allocation process is added as a separate call to the loan contract despite adding and extra step in some cases (eg. single allocation creation), and the call's additional complexity, due to the following reasons:

- **No single allocation API** - We aim not incentivize single allocations, and thus choose not to provide an API as a simpler version of contract creation with single allocation. Existence of this API may incentivize the wrong behavior.
- **Allocation as a separate step** - We aim to make allocation changes no more difficult than the initial allocation, and so, we will reuse the allocation process both for initial creation and subsequent changes made to allocation.
- **Unfortunate but unavoidable friction point** - The level of complexity proposed does not exceed the loan takers assumed level of sophistication and tolerance for usability costs.

Liquidation process

Liquidation can be proposed by any blockchain user, who agrees to put up a deposit, and request for a specific debt liquidation position to be triggered. However the final liquidation decision would be accepted only if the specific monetary conditions of the position merit it and after the delayed dispute process also issues the validity of that request.

Liquidation trigger decision - A 100% allocation to one price feed will mean that liquidation decision will be solely made based on the price feed, and contents of the position itself. A combination of say 3 price feeds with 33.3% each will mean that liquidation condition will trigger only if all 3 price feeds are in approximate agreement, otherwise the position will have to go through the dispute resolution process.

-ToDo

Price feed

The main responsibility of price feed providers is the truthful **daily** providing of **historical prices** to the contracts, consisting of 2 exchange rates, one for the native token in reference currency, and the other in its pegged version. These rates are expected to be the median rates based on volume. The system enforces having historical prices available within 1 day (See maximum price feed delay) enforced by an imposed penalty to the provider's revenue pool.

For example, in our MVP implementation, this consists of ETH value in USD as well as in pegged USD as reported accurately from exchange market activity. If there was `10,000` ETH traded for USD during the day, and `50%` of volume was under `$201.2` USD and the other `50%` was over `$201.2`, then `$201.2` should be reported as the ETH/USD price. Similarly, let's assume the Pegged USD was traded against ETH, with its median valued at `$0.9805` that day. The price provider would report the following and results:

- ETH/USD = `201.2`

- ETH/Pegged USD = `205.2`

Price feed values are results of formulas calculated from known market exchange prices and volumes. The ideal case is for all providers to have full access to the universe of all *legitimate* trades along with their volume and price, and good faith providers should strive to do so. However, due to many restraining factors, including uncertainty around authenticity of info from some sources or exchanges, as well as pure technical limitations, each provider will choose a specific set of samples they can depend on, at any given time. They should publish their methodology for compiling these historical prices, so the community can verify them in favor of higher level of trust and predictability for the ecosystem, as public good.

High trust providers are also expected to report something called **instant price** almost **immediately**, as they notice changes of over `1%`. This reporting by highly trusted providers, would be strongly encouraged by the ecosystem members (community) however there is no hard penalty imposed by the system itself.

Price feed liquidity pool

Loan fees - Each price feed has a corresponding global revenue pool that increases in value as a result of an incoming portion of loan fees. See loan fees and ***price feed revenue rate*** under monetary variables for more details.

Dispute resolution credibility - The total size of the price feed liquidity pool affects their level of credibility during dispute resolution phase.

Dispute penalties (slashing) - During the dispute process, any penalties issued to the price feed provider will come out of their liquidity pool. This is to incentivize the provider into constructive behavior.

Delayed/gradual/partial payouts - Price feed providers need to have a sustainable business model and so they are expected to have an income from the loan fees. This income takes the form of a delayed and gradual payout from the price feed liquidity pool belonging to the feed provider. The payout is long term delayed in order to incentivize the price feed provider to maintain a constructive long term engagement with the platform. The payout is partial because there needs to be a significant liquidity pool to use in case of dispute penalties.

Price feed delay and resolution

The level of delay and resolution offered by a price feed provider will depend on a few important considerations amongst others, and we trust the providers to consistently reach an equilibrium based on these factors:

1. **Market conditions** - The level of demand and competitive differences in the market determines characteristics of the offered price feed. At one extreme the market's demand may not exceed that of the system where loan takers accept the `1 day` delay and `1 hour` resolution, at another extreme, the market may demand instantaneous recording of prices on the blockchain, to enable almost instantaneous loan confirmation.
2. **Cost of reducing delay** - Cost of reducing price recording delays on the blockchain, driven mainly by gas cost. At one extreme the provider may choose to record values 1 time per day with minimum gas cost (commonly <1$), or at the other extreme they may incur that gas cost on almost every block for instantaneous records.
3. **Cost of high resolution** - Cost of providing high resolution records driven mainly by storage cost. At one extreme, the provider may choose to record prices every `1 hour`, minimizing storage cost. At another

extreme, the provider may choose to provide up to the block price records despite significant storage cost.

Dispute process

- ToDo

Price feed consensus violation penalties

- ToDo

Savings account

Locking an amount of pegged currency in the savings contract, will allow for a collection of interest based on the variable daily savings rate (See ***savings interest rate*** under monetary variables), for the duration time of that locking. An call to withdraw the accumulated interest, calculates the to-date savings interest rate, savings interest amount and sets a timestamp for future interest withdrawal calls. An owner call to close the savings account, should be performed after withdrawing all applicable interest to that date.

Savings rate

Loan fees are also varied based on the global equilibrium price of the pegged currency, in order to incentivize increase or decrease the pegged currency supply based on changing the demand for pegged currency by consumers and thus affecting the equilibrium price. See ***savings interest rate*** under monetary variables.

Savings pool

Savings pool is a portion of pegged currency tokens held by the main system contracts, set to be payed out to money owners as interest, when they lock their tokens in a savings account contract for periods of time. A portion of the fee paid by loan takers is always transferred to this pool. Also, any penalties imposed on price feed are taken out of the offending provider's revenue pool and transferred into the savings pool. Payout is based on the ***savings interest rate*** (see savings rate and variable definition under monetary variables).

System state

Main system contracts hold the following variables related to the areas of the system like Savings, Loans, and Price Feeds:

- Loan contracts
- Price feed contracts
- Savings account contracts
- Total and individual price feed allocations (in pegged currency)
- Price feed revenue pools (in pegged currency)
- Savings pool (in pegged currency)
- Finalized medium trust price feeds
- Current (initialized) medium trust price feeds
- Archive of rates
    - 7 x Daily
    - 5 x Weekly, 5 weeks (~ monthly), 25 weeks (~ bi-yearly), 125 weeks (~2.4 yearly), 625 weeks (~ 12 years), 3125 weeks (~60 years)

    - Savings interest rate
    - Loan fee rate
- Short-term historical prices
    - 7 x Daily (for last week)
    - 5 x Weekly (for last month)
    - (Native)/(Reference) median price
    - (Pegged)/(Reference) median price

The system will have multiple ***states of operation***:

- **Normal**: where no anomalies in historical prices have been detected, and where daily and instant prices are within known variability thresholds `10%`.
    - Currency issuance will be instant within transaction.
    - Loan liquidation will take `1 week`
- **Uncertain**: where minor anomalies have been detected such as 1. change in instant price is possibly not reported by minority 2. drastic recent changes in allocations such as `10%` drops 3. Volatility of reported prices exceeding `20% weekly` despite lack of dispute.
    - Currency issuance will take `2 days` and will be according to minimum of historical native token prices
    - Loan liquidation will take `2 weeks` and will be evaluated against the maximum of historical native token prices
- **Disputed**: where major disagreement is occurring between medium trust providers
    - Currency issuance will take `1 week` and will be according to minimum of historical native token prices
    - Loan liquidation will take `5 weeks` and will be according to maximum of historical native token prices

Automated monetary system

The main goal of the automated monetary system is to balance supply of pegged currency with its demand, and maintain a price pegged to the reference currency, as reported by the price feed providers as historical prices. The main variables that the monetary system can affect are the following:

- Price feed revenue rate
- Loan fee rate
- Savings interest rate
- Loan collateral threshold ratio

Monetary variables

Loan fee rate - Is a daily rate set by the automated monetary system that determines the fee paid by loan takers upon structural changes made to their loan, such as allocation or issuance changes. For example, at a given point in time, the loan fee rate could be `0.01096%` per day (equivalent to `4%` per year). Assuming no future change to the rate (which is unlikely), the loan taker should project to pay `$400` per year in fees on a loan that has issued `$10,000` worth of pegged currency, upon closing the loan. The actual annual rate will be calculated by adding all the variable daily rates.

The monetary policy engine aims to vary the rates based on the global equilibrium price of the pegged currency, in order to incentivize increase or decrease the pegged currency supply based on the creation or cloning of debt positions, and thus affect the equilibrium price of the pegged currency itself from the supply side.

Price feed revenue rate - Is a weekly rate set by the automated monetary system that determines what percentage of total system issuance is expected to eventually be paid out to price feed providers. The revenue comes from part of the loan fee stream and so the price feed revenue daily rate has to always be lower than the loan fee daily rate. The rest of the loan fees will go into the savings pool to be paid out to savings account owners. For example, at a given point in time, the price feed revenue rate could be `0.01917%` per week (equivalent to `1%` per year). Assuming no future change to the rate (which is unlikely), a price feed provider can project its revenue pool to receive `$20,000` yearly assuming a total issuance allocation of `$2,000,000`. This could be the case if total system issuance is `$10,000,000` and the providers average weighted allocation percentage is `20%`.

Savings interest rate - Is a daily rate set by the automated monetary system that determines the interest paid to owners of savings accounts. For example, at a given point in time, the savings interest rate can be `0.01096%` per day (equivalent to `4%` per year). Assuming no future change to the rate (which is unlikely), the savings account owner can project an interest of `$400` per year on a savings account balance of `$10,000`.

Loan collateral threshold ratio -

- To Do

Notable constraint adjustments

In an ideal loan (debt position) system, one would expect perfect trust of the price feeds, very fine grained price feed resolution, and instantaneous response to loan taking requests or liquidation requests in appropriate conditions. However, given the constraints of highly decentralized protocols on public blockchain systems, as well as the urgent need for simplification, in order to reduce risks and increase efficiency, we choose to bend the ideal rules as long as these changes are communicated clearly to and are accepted by stakeholders, and as long as they result in a secure system overall.

Loan taking process adjustment

Normally the loan taking process should be expected to complete in one transaction, however during times of dispute between the rates provided by the system's price feed providers, it is reasonable to delay the loan taking process by a few days. Normal operation should be resumed on all other occasions.

Liquidation process adjustment

The most acceptable cases of liquidation from the perspective of a loan taker is when there is a significant and non-intermittent drop in the value of the collateral, and they've had enough time to respond to it. We can adjust the definition of acceptable liquidation to such cases, and ask the loan liquidators to take on the additional risk of having to predict if the drop is non-intermittent. We would however need to compensate the liquidator for the additional risk they are taking. It is thus become acceptable to condition liquidation upon the drop being persistent over the course of a few days and expect the loan liquidator to make a judgement on the likelihood of this persistence, and take a profit or loss accordingly.

Time constraint adjustments

One of the key insights that helps with efficiency and simplicity of this on-chain loan system is that using a collateral in smart contract to peg relatively stable currencies, does not require a high level of time sensitivity, in the following ways:

- **Liquidation process delay** - Given the intent and requirements from loan takers as well as loan liquidators, the liquidation triggering or the dispute process do not necessarily have to occur in real time, and can be delayed, even lasting for days, as long as they occur in a predictable manner.
- **Low time resolution** - One does not require a perfectly full resolution set of prices, in order to confirm a persistent drop in prices over a long period of time. The only case where higher resolution helps is determining the liquidation bid winner, as the first actor that submits a liquidation request right before passing solvency threshold. (ToDo - mitigate against the risk of collusion to win liquidation by manipulating price feed)

Price feed penalties

All percent-based (%) penalties are enforced with respect to the corresponding provider's price feed revenue pool. The penalized value os transferred to the savings pool.

- **Missing price report penalty** - Missing any set of price values for each 1 day results in a penalty of 5% for that specific violation.
- **Reference price inaccuracy penalty** - Any one day that the historical reference currency price is reported to be 5% more or less than the system-wide median price, a penalty of 5% is imposed on the provider.
- **Pegged price inaccuracy penalty** - Any one day that the historical calculated peg currency price is reported to be 1% more or less than the system-wide median price, a penalty of 5% is imposed on the provider.
- **Savings account vote penalty** - Every week, if a consistent set of savings account owners vote to penalize a specific price feed address, the system will penalize that provider by a percent calculated by formula: `100% * total value of all voting savings accounts / total value of all savings accounts`

Appendix

Terms

Asset - Any valuable object. It may not necessarily be used as money for exchange of value.

Money - A tool used by humans to exchange value. Any valuable object (such as a coffee mug) can theoretically be used as money, perhaps on rare occasions. However a good money candidate, can be used in more contexts.

Gold, Bitcoin (BTC) and Ether (ETH) are all examples of asset moneys, that can be used for exchange on occasion, but are not necessarily the best option when we have access to more widely accepted alternatives like USD.

Currency - A type of money that is commonly used in day to day commerce, as it is used and accepted by many buyers and merchants. Good currencies are stable in value and are therefore good Stores of Value (SoV). Good currencies also are used as Units of Account (UoA) by more people, and are accepted by more people as Medium of Exchange (MoE). US Dollars ($) is the example of a good currency money, so are Chinese Yuan, Japanese Yen, and Euro.

Token - The digital representation of value on a public blockchain. They can be the digital representation of a real world asset, or they can have inherent digital value like in the case of Bitcoin, Ether and others.

Medium trust price feed - Price feeds either in the top `25` in weighted allocation, or allocated at least `4%` of all loans' weighted value. The historical prices from all these feeds are used in weighted form to determine the median daily historical price.

High trust price feed - A select group of price feeds either in the top `5` in weighted allocation, or allocated at least `20%` of all loans' weighted value. Through social contract, High trust feeds will be, expected to report price changes of more than `5%`, on the blockchain within `60 seconds`. All high trust feeds are also considered part of the medium trust collection.

Low trust price feed - Simply any non-zero allocated feed that is not medium trust. Any price feed that is allocated a non-zero % of a loan with non-zero native token deposits, which is not a medium trust feed.

Magic values

In software engineering, magic values refer to constant values selected by the software author, that determine the constraints that the system operates under. The selection of these values often involves significant deliberation and requires a level of justification. It is always a good question to ask: *Why was that specific "magic value" selected, instead of higher or lower values?*

Maximum price feed delay - `1 day` - `86400 seconds`

Maximum price feed allocations - `5`

High trust price feed count - `5` - Every feed with at least `20%` is included.

Medium trust price feed count - `25` - Every feed with at least `4%` is included.

Front running resistance

Performing liquidations asynchronously and through and request initialization and finalization process, and providing aan initial grace window of a few blocks, ensures that no one can front-run another request and prevent them from participating also. At worst case, the two almost-simultaneous requests will share in the proceeds.

Also anonymizing liquidation requests, ensures that no front-runners can highjack the reputation of another liquidator, and requires them to also have their own logic for evaluating the liquidation conditions, at which point it no longer is much of a front running attempt anyways.

Volatility

Historically, stable currencies have not shown a high level of short term volatility, where for example the price goes down 20% one day and goes back up a few minutes or hours. As such transaction delays have a much smaller chance of triggering costs to a stakeholder. There is often however clear longer term trends that can be observed with stable currency, at different points in time.

However one always needs to be prepared for short term volatility of the native value token due to unknown knowns such as general boom and bust cycles (or events), as well as unknown unknowns we can't even imagine. Although not likely to occur in case of a mature blockchain, it is entirely reasonable to consider an hour long drop in the order of 50%, or a sudden rise in the order of 400%, as a possibilities, and prepare for them.

Note on volatility of Ether

Since Ether (ETH) is the other side of the first viable product based on our proposed financial instrument, its volatility also affects our considerations. ETH in the recent years, due to its increasing uses as gas, speculative investment, and collateral, has been relatively more stable as compared to years before. Also due to its future staking use in Ethereum 2.0, we expect its long term volatility to decrease in general. However, it remains primarily a speculative asset subject to significant volatility.

Versioning - in case of emergency (hard-forks)

The ultimate goal of such a decentralized smart contract system is to be ownerless and live forever. That is, if the current open source implementation were the solution to our original stable-coin problem, there should be no subsequent version needed.

However, due to possibilities of future upgrades to the underlying blockchain itself, as well as due to the remote possibility that the system may, at some point, operate in some unexpected ways, we are required to at least consider the possibility of winding down this version in favor of a next one.

We are currently witnessing this type of transition with the SAI to DAI migration by Maker. There are a few lessons to be learned from this experience, in the remote case migration is needed.

Decentralized price feed

One of the price feeds to the system on Ethereum can be constructed as a decentralized contract, based on market pricing information provided by the UniSwap V2 decentralized exchange. This ownerless, decentralized price feed would then automatically transfer any of its revenue payouts back into the loan system as part of the savings pool.

Ecosystem and game theory

The stable-coin system lives on the public blockchain, however the players interacting with it directly or indirectly, all live in the real world and will follow strategies and courses of action, based on their incentives and the changing ecosystem. In order to ensure the success of the stable-coin system, and the money product that it represents, we need to fully understand the ecosystem subsuming it and mitigate any foreseeable issues that may threaten the product's success.

Strategic dynamic changes

A few possibilities around shifting of strategic power between groups of actors:

- Conflict between price feed providers - Sub-group of price feed providers going rogue to eliminate another group from ecosystem
- Price feed providers collusion - Feed providers colluding to abuse the system through liquidations
- Price feed providers and loan taker collusion - Feed providers and loan takers colluding to take advantage of the money holders and users
- Influence consolidation through loan taking aggregators - Consolidation and influence of loan takers preferences through aggregators (like InstaDapp), and collusion with price feed providers for kickbacks

- Shortage of pegged currency - This can occur during wind-down of protocol in case of hard-fork, causing loss of collateral for loan takers

ToDo - Detail above dynamic changes along with mitigation steps considered by the system

Protocol changes

A few possible outcomes around changes to the protocol itself:

- Perpetual operation (no-fork) [default] - The same community will rely on the same protocol perpetually.
- Soft-fork - We end up with 2 sub-communities and 2 protocols adopted by each.
- Hard-fork - We end up with the same community moving to a new protocol. The old protocol will be transitioned and retired.

Perpetual operation (no-fork)

Summary: The same community will rely on the same protocol perpetually.

A successful general-purpose consumer money product is assumed to be held and used by large numbers of everyday people, which means the cost of altering the money during usage is extremely high to the individual people using it, as well as any other stakeholders. As a result, perpetual operation, meaning no disruptions to routine operation forever, is the ideal end-state of the system as designed, one that incurs the least costs in a large number of people having to switch from one protocol to another.

When the viability of the ecosystem are not in significant danger, this should be considered the default option. Of course, this is a subjective criteria, and is subject to the community's interpretation.

Soft-fork

Summary: We end up with 2 sub-communities and 2 protocols adopted by each.

There is always a non-zero probability that beliefs about operation of the ecosystem, amongst its actors could increasingly diverge in time, in which case, a community split is possible. In such a case, those whose beliefs are more aligned with the existing operations and the original parameters of the system are likely to stay with it, and incur less costs due to not having to switch to a new version. The existing sub-community however will incur a lot more cost.

The cost incurred by the sub-community leaving the ecosystem, includes but is not limited to, the cost of money users exchanging out of the pegged currency, the loan takers' cost of having to purchase sufficient pegged currency to cover loans (debt positions) and close them, as well as the cost of exiting price feed providers being slashed for the drops in their allocation numbers.

The original community will also experience a period of less certainty due to the large volume of pegged currency and deposits leaving the system, and incur other costs depending on size of the existing group.

Hard-fork

Summary: We end up with the same community moving to a new protocol. The old protocol will be transitioned and retired.

There is always a non-zero probability that the community as a whole may come to a conclusion that changes to the on-chain protocol are required in order for the ecosystem to remain viable. In such a case, the community as a whole would have to coordinate simultaneous transition to a new version of the system, one that uses parameters agreed upon by the community as a whole.

Given that the system is immutable and holds a large amount of state, in the form of value, transitioning out of it will be very costly, including but not limited to:

- **Uncertainty and unpredictability** is a significant cost to an ecosystem that operates optimally based on predictability, trust, and truthful price reporting as a schelling point for all stakeholders. Any effective mitigation to the this will include:
    - An effective transition plan, one that considers mitigations to possible outcomes, and ensures the interests of all stakeholders are preserved as much as possible.
    - Clear communication of the transition to all stakeholders, instructions on how to proceed.
- **Price volatility** is inevitable given that transition out of the current system requires movement of large amounts of value out of the system, in form of exchanging out of the pegged currency money, and in form of closing loans (debt positions) using large amounts of purchased pegged currency. Higher amounts of uncertainty, as mentioned above, will adversely affect volatility.
- **Money users' overhead** - very high cost of having to even care about this, cost of effort needed to exchange out of the pegged currency.
- **Money users' exchange costs** - Exchanging tokens has a non-zero fee regardless of the exchange used, this is no different.
- **Loan takers' overhead** - high cost of having to care about this in the first place, as well as the effort to purchase sufficient pegged currency to cover loans (debt positions) and close them.
- **Loan takers' exchange costs** -
- **Price feed provider slashing** - the cost of exiting price feed providers being slashed for the drops in their allocation numbers, as well as losses of time based payouts.
- **Possibility of un-closable loans due to lost pegged currency** -
- ToDo - others ...

Community

Community's potential points of common belief:

- **On-chain governance minimization** - Skepticism of governance through tokens, and belief in minimizing it.
- **Market truth as schelling point** - The common belief that truth is the ecosystem's schelling point. Community members and ecosystem participants believing they can sustainably benefit from helping operate and interacting with the system, and that truthfully reported price feed information is in everyone's interest.
- **Crypto-sovereignty** - The belief that sufficiently decentralized public blockchains as a whole from an unconventional sovereign jurisdiction on the internet, one that lacks many of the restrictions and friction points of today's geographical jurisdiction, and is more suitable for free economic activity. This consists of a few other shared beliefs around censorship resistance, decentralization and trust minimization.
- **Crypto-nationalism** - (a better and alternative definition of token maximalism) is the sometime exclusive allegiance to a specific public blockchain. It consists of the following set of beliefs:
    - Belief and feeling of belonging to that specific public blockchains community

- The specific blockchain represents the best potential unconventional sovereign jurisdiction on the internet, which is akin to a nation.
- More extreme versions of crypto-nationalism will advocate for exclusive citizenship in many contexts, at the cost of other public blockchains.
- The native digital asset of this specific public blockchain is the most trust-minimized and is a superior store of value.
- The specific blockchain will become the most ubiquitously used public blockchain in the future.