Atomix Litepaper

Version: 1.16 (April 2021)
www.atomix.finance

Overview

Atomix is a lending platform implemented in large part on blockchain, that enables the use of secured real-world assets as collateral for loans in stablecoins and offers an unmatched level of transparency and flexibility.

This document is a lite-paper describing the functionality of the system.

Contents

Contents

Participants

Originators/Borrowers

An early target is small to medium size businesses (Originators) seeking secured lines of credit but who have difficulty sourcing finance from traditional sources in the credit market. Some of these Originators have the right to receive future cash-flows which they would like to convert to funds in the present to free up capital for operations and expansion in exchange for providing security over their assets (including the right to receive cash-flows and any related security) as collateral. Atomix will provide them near-instant, flexible access to liquidity on competitive terms. Other Originators might own or wish to acquire other assets (which may or may not themselves generate income) against which they would like to raise funds.

Security will be taken over an Originator's assets to secure the ongoing payment obligations of the Originator under the loan from Atomix and tokens which evidence that security will be given to the Originator. Those tokens must be deposited with Atomix for the Originator to draw down under the loan, at which point the Originator becomes a Borrower for the purposes of this Litepaper.

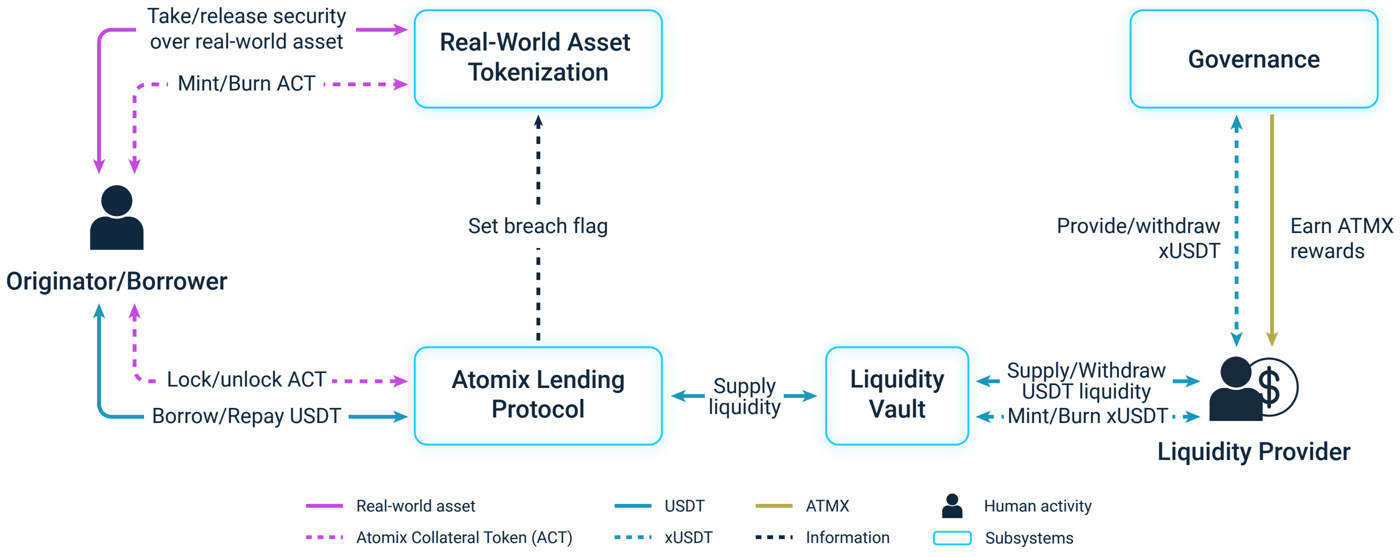
Where the term "asset" or "Asset" is used in this paper, it should be read as a reference to an individual asset or to a pool of assets, depending on the nature of the assets that an Originator wishes to use as collateral and the context. A single security document may be used to take security over multiple assets or a pool of assets of an Originator.

Liquidity Providers

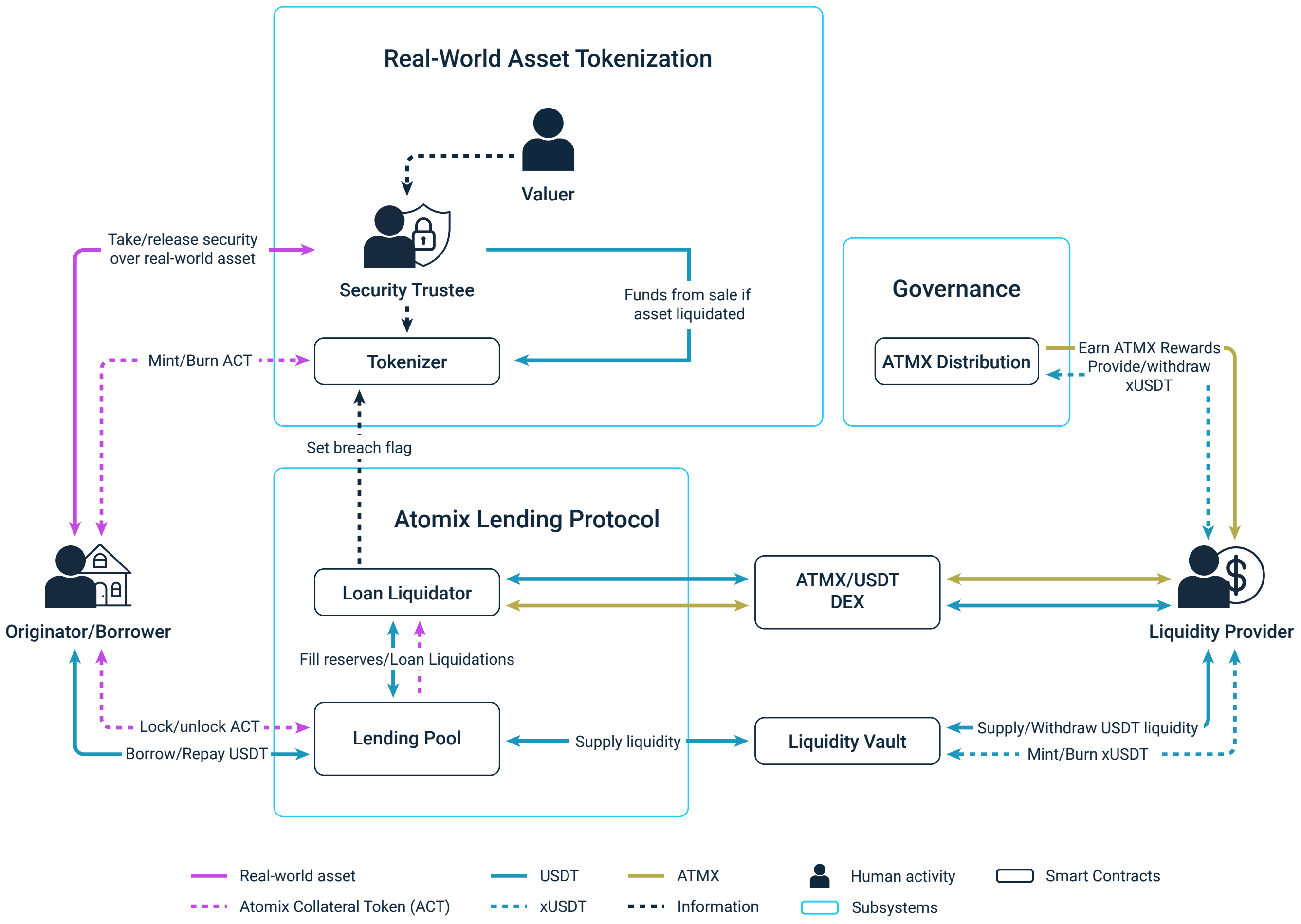
Liquidity Providers have funds available for investment and want to receive a competitive return plus governance tokens which can be used to modify protocol parameters or sold on the open market.

System Overview

This diagram shows the main subsystems.



This is a more detailed version of the previous diagram showing the main components of each subsystem.



Part 1: System design

Tokens

There are a number of different crypto tokens in the system. Their movement is described by coloured lines in the main system diagram.

USDT

This is Tether. A token already in wide circulation on the Ethereum blockchain. It is a "stablecoin" designed to maintain a 1-1 exchange rate with the US dollar.

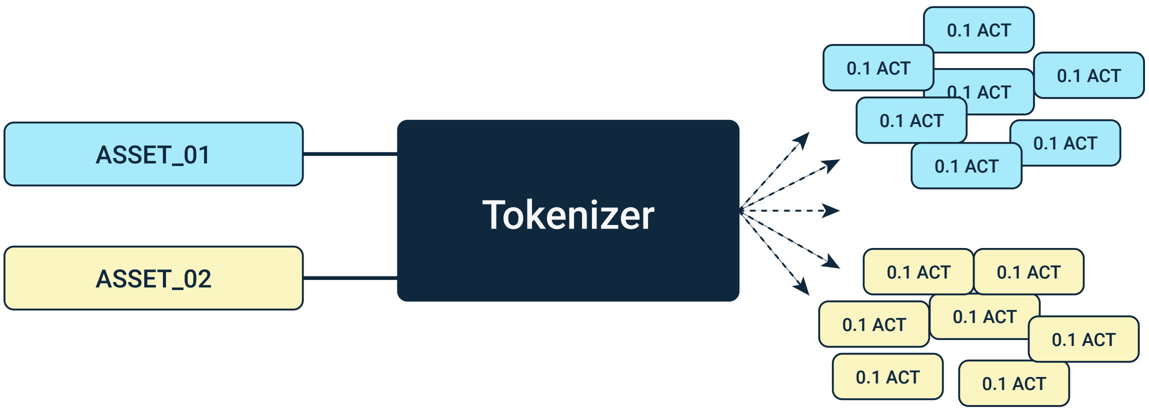
https://www.coingecko.com/en/coins/tether/usd

All secured assets in the system are valued in terms of USDT.

When interacting with smart contracts, users need to use a stablecoin cryptocurrency such as USDT to represent a fiat currency. U.S. dollars (as well as other fiat) can be exchanged for USDT in a number of external crypto-exchanges. These exchanges are not part of the Atomix system.

Atomix Collateral Token (ACT)

This is a crypto token minted by the Tokenizer. For each asset over which security is taken, one ACT will be minted. This token is infinitely divisible. Fractions of this ACT are fungible with respect to each other. However, a fraction of an ACT minted against one Borrower's asset is not interchangeable with ACT minted against a different asset.



In the example above, you can only release security taken over the ASSET_01 by burning blue ACT. You cannot use the blue ACT to release security taken over the ASSET_02.

ATMX

These tokens can be sold by holders on the open market as well as distributed by the protocol via the ATMX Distribution contract. Holders of ATMX have voting rights on deciding system parameters as well as updating the smart contract code.

xUSDT

This token is minted by the Liquidity Vault and represents USDT supplied by Liquidity Providers. When a Liquidity Provider supplies USDT, the system mints and transfers xUSDT to the Liquidity Provider. Later the Liquidity Provider can redeem their xUSDT plus a return.

Real-World Asset Tokenization

This subsystem deals with valuing real-world assets, creating security over those assets and minting tokens evidencing that security. It also deals with burning the tokens to release the security as well as enforcing the security when necessary in order to recover the debt due to the platform, including through a power of sale over the assets.

Security Trustee

The Security Trustee third party acts as the interface with Originators and other third parties in assessing the suitability and value of an Originator's Asset as collateral. The Security Trustee is also responsible for:

- Generating the security document which creates security over an Originator's Asset and digitally confirming that it has been executed by the Originator
- Instructing the Tokenizer to mint ACT
- Acting as a traditional security trustee with power to enforce the security taken over the relevant real-world asset when a Borrower is in breach.

Valuer

The valuer, a third party, gives an estimate of the Risk-Free value of an Originator's asset. This estimate will be used by the system for a specific time period (say 2 years). The valuation is used to determine the value of the one ACT token minted.

Risk-Free Value

The value ascribed to an Originator's Asset representing the amount which the Valuer is confident will be recovered if the security over that Originator's Asset were to be enforced and the Asset liquidated in the open market. For an asset like land, which does not fluctuate wildly in value, the Risk-Free Value could be high, e.g. 75% of the full value of the asset. For more volatile assets, it could be just 50%. It should also reflect any losses the system is likely to incur due to fees.

In the future, with the evolution of the Atomix protocol, the Originator may be able to take out insurance to cover the risk that the asset fails to meet this sale-price if liquidated due to unforeseeable market-related developments.

Real-world assets and Secured Assets

A real-world asset is an asset owned by an Originator over which security will be taken.

Once security has been taken over a real-world Asset (becoming a Secured Asset), an ACT token will be minted which evidences that Secured Asset.

Tokenizer

The Tokenizer is a smart contract used by the Security Trustee to mint and burn ACT, which evidences the Secured Asset. It is also used by the lending system to flag that a Secured Asset is backing a loan which is in breach and the security therefore is in danger of being enforced which could lead to the sale of the Secured Asset.

Security taking process

When security is taken over an Asset, that Asset becomes a Secured Asset, and ACT will be minted evidencing that Secured Asset. Before the security taken over the Asset can be released, any monies borrowed from the protocol must be repaid and all the ACT evidencing the security taken over that Asset must be burnt.

Taking security over an asset:

1. Originator requests that the Security Trustee creates security over the real-world asset
2. A valuer has provided a risk-free valuation
3. Security Trustee creates the security documentation which is signed by relevant parties and uploads the document hashes to the protocol
4. Security Trustee notifies the Tokenizer that security has been taken over the asset following which the Tokenizer mints 1 ACT which it passes to the Originator

Releasing security over an asset:

1. Originator requests that the protocol burns all the ACT for that asset
2. If there are no monies outstanding to the protocol then the ACT is burnt
3. Security Trustee releases the security held over the asset

ACT Invariant

Once a specific Secured Asset, S_0, is locked up and 1 ACT_0 token minted against it, then a number of actions can be performed using the Tokenizer including:

- Revaluing the ACT_0 token
- Burning a portion of the ACT_0 or minting more of it
- Revaluing the underlying asset itself
- Replacing the Secured Asset with cash from enforcement of the security (including a sale in the case of liquidation)

All of these actions must preserve a crucial invariant we call the ACT invariant.

Definitions:

Value(S_0):	*Value of the Secured Asset*
N_0:	*Quantity of ACT_0 in existence*
Value(1ACT_0):	*Value of 1 ACT_0*

Then the following invariant must be preserved:

$$\textit{Value}(S_0) \geq N_0 \times \textit{Value}(1ACT_0)$$

Security Trustee actions

There are various actions that the Security Trustee needs to perform with the Tokenizer:

- Minting ACT in relation to the Secured Asset
- Confirming all ACT has been burnt before releasing the Secured asset
- Revaluations of the Secured Asset

Value of ACT

Initially exactly $1ACT_0$ is minted against a Secured Asset, S_0. However, due to revaluations or liquidations, ACT_0 can be minted or burnt resulting in a different amount of ACT_0 existing. We need an equation to calculate the value of a given amount of ACT_0:

Definitions:

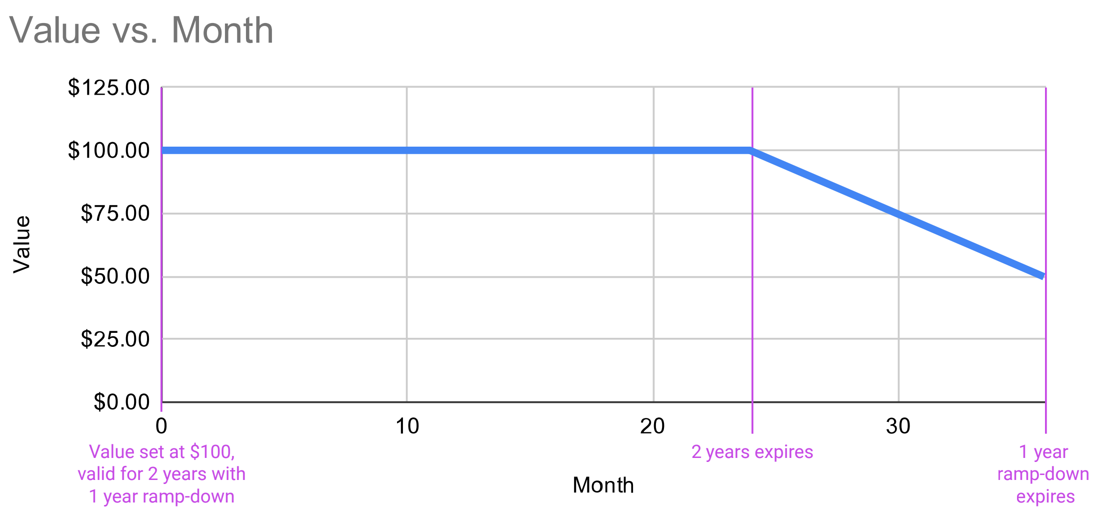
ACTAmount:	*Amount of ACT_0 we are trying to calculate the value of*
Value(S_0):	*Value of the Secured Asset evidenced by the ACT_0*
totalACT_0:	*Amount of ACT_0 in existence*

Now we can calculate the value of ACT:

$$\textbf{Value of ACT} = \frac{\textbf{ACT Amount x Value}(S_0)}{\textbf{total } ACT_0}$$

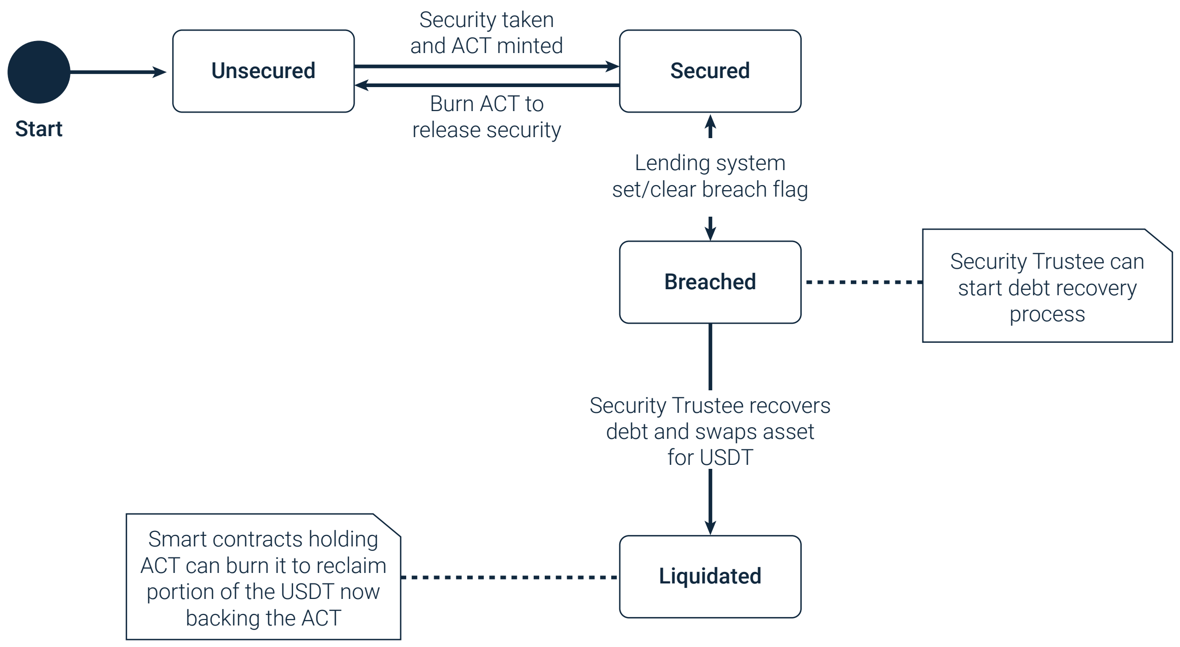
Time-limited valuations

When the valuer assigns a risk-free value to a Secured Asset, it will be valid for a specific period of time. However, rather than having the value suddenly drop to zero value after this period, the system will also assign a ramp-down period over which the value will steadily drop to half its original value.



Asset States

An asset can be in a number of different states. Its state is tracked by the Tokenizer.


Start
Unsecured
Security taken and ACT minted
Burn ACT to release security
Secured
Lending system set/clear breach flag
Breached
Security Trustee can start debt recovery process
Security Trustee recovers debt and swaps asset for USDT
Liquidated
Smart contracts holding ACT can burn it to reclaim portion of the USDT now backing the ACT

It starts off in an unsecured state. The Security Trustee takes security over the asset (creating a Secured Asset) and mints 1 ACT, evidencing that Secured Asset. The Originator is now free to lock up their ACT as collateral in the lending platform. The lending platform can set and clear a breach flag indicating if the Borrower has breached the terms of their loan.

While the loan is in a breached state, security taken in respect of the relevant asset is in danger of being enforced, which could lead to the liquidation of that asset. The Security Trustee can now start the debt recovery process (e.g. sending out warning letters). During this time, the Borrower can always pay off their debts to reclaim their ACT and burn it to release the security.

If, at the final stage of the debt recovery process, the loan is still in a breached state, the Security Trustee can enforce the security, which may lead to a sale of the Secured Asset. When the debt is recovered, the Security Trustee passes USDT to the Tokenizer. Now the state changes to Liquidated and any smart contracts holding ACT can burn them to claim a portion of the USDT.

Liquidity Vault

Liquidity Providers can supply USDT to the Liquidity Vault. When they do this, they receive xUSDT tokens representing the USDT supplied. When the Liquidity Provider redeems their xUSDT tokens, they get their USDT back plus a return.

The Liquidity Vault returns rate is algorithmically determined by the Atomix protocol.

Atomix Lending Protocol

An Originator can lock up their ACT in the Atomix Lending Protocol and become a Borrower. This subsystem lets Borrowers take out USDT loans backed by ACT tokens.

Lending Pool (LP)

The Lending Pool is a smart contract which enables Borrowers to deposit ACT and borrow USDT against its Secured Asset.

Borrowing collateral factor

Each Borrower has a borrowing limit which is a portion of the value of all the ACT they have deposited with the protocol. This portion is called the Borrowing Collateral Factor. For example, it could be between 60% and 80%.

Borrowing

From the moment of borrowing, a loan starts accruing interest which is added to the value of the loan. The Borrower needs to ensure that the aggregate value of their loan, including accrued interest, does not exceed their borrowing limit. At any time, the Borrower can repay any portion of their loan, borrow more USDT or deposit or withdraw ACT as long as they stay within their borrowing limit.

USDT will flow in and out of the Lending Pool as some loans are repaid and others are originated. USDT Liquidity can be supplied to the Lending Pool from the Liquidity Vault.

Interest algorithm

The Atomix protocol borrowing interest rate is determined algorithmically by the Atomix protocol.

Filling the Liquidation Reserves

Some Borrowers' interest is directed to a pool owned by the Loan Liquidator. This pool is used if loans go into breach to fund Loan Liquidations and is replenished when either the Borrower repays the in-breach loan or the underlying asset is liquidated/the debt is otherwise recovered.

Loan Liquidator (LL)

The Loan Liquidator is a smart contract with two main responsibilities:

- Loan Liquidations
- Managing the cash-in process

Loan Liquidations

If an account breaches its borrowing limit, the Loan Liquidator is activated. It transfers some of the Borrower's debt which is currently owed to the LP so that it is now owed to the LL receiving an equivalent portion of the ACT tokens from the LP in return. This is called a Loan Liquidation. This brings the loan to the LP back below the borrowing limit.

Now the Borrower's debt to the LP is smaller, but it now also owes a debt to the Loan Liquidator, which now also holds ACT as collateral against that debt. The Loan Liquidator, charges default interest on the debt at some multiple of the standard LP borrowing rate (say 2X).

The Borrower has a borrowing limit with the Loan Liquidator, set as a proportion of the collateral held by it. That proportion is called the Liquidation Collateral Factor and will be higher than the Borrowing Collateral Factor (which determines the borrowing limit to the LP).

It is possible for the Borrower to breach the Loan Liquidator's borrowing limit just as it could breach the LP's limit. A loan liquidation is triggered if the Borrower breaches either borrowing limit.

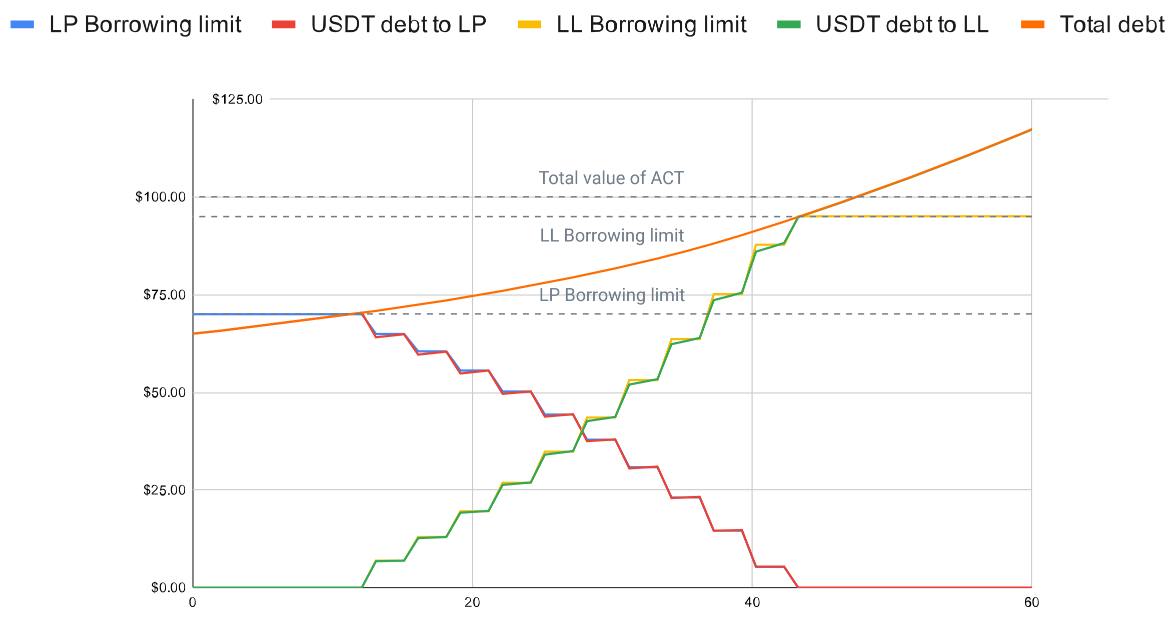
After a Loan Liquidation, the debt to the LP will always come back within the Borrowing Collateral Factor limit - even if that means transferring all the debt to the Loan Liquidator.

At any time, the Borrower can pay off the debt to the Loan Liquidator and recover their ACT, which will be transferred back to the Borrower's wallet.

If the Loan Liquidator holds ACT, then it will set the breach flag on the related loan. If it no longer holds any ACT, then it will clear the flag.

Loan Liquidation Example

Suppose a Borrower has a loan of $65 backed by ACTs worth $100. Time in months is on the x-axis.



This graph shows the loan to the LP steadily accruing interest until it breaches the Borrowing Collateral Factor limit ($70). At this point the Loan Liquidation is activated and transfers ACT to the Loan Liquidator in return for the paying off some of the debt. This ensures the loan is back within the borrowing limit, then when it breaches again, another Loan Liquidation is triggered.

This would continue until all the loan has been transferred to the LL, however, in reality we expect the security taken over the real-world asset to be enforced and the debt recovery process commenced long before all of the loan is transferred, at which point the loans will be "cashed in" - see later.

The orange line shows the total combined debt the Borrower has to both LP and LL.

Loan Liquidation Math

This section derives the formulas that the LL uses to determine how much ACT and USDT to transfer. Any percentages will actually be stored as values between 0 and 1 (so 2% would really be 0.02 when we do the calculations).

Notation: There are a number of parameters which have equivalent versions in the Lending Pool and the Loan Liquidator. In this case we give them the same name, but the Loan Liquidator versions have an added prime (') to distinguish them.

Tuning Parameters

These are system-wide parameters of the Liquidation algorithm that can be set by governance.

> ***C_0*:** *Collateral Overshoot - Amount below the borrowing limit that the loan should be after a Loan Liquidation. E.g. if this is set to 2% and the Borrowing Collateral Factor was 60%, then after a Loan Liquidation, the remaining loan would be 58% of the value of the remaining collateral, ensuring some breathing space before needing to liquidate again.*

> ***I_m*:** *Liquidator Interest rate multiplier - Multiple of the LP borrowing interest rate charged by the Loan Liquidator.*

> ***C_f'*:** *Liquidation Collateral Factor - The Borrowing limit of the Loan Liquidator as expressed as a proportion of the value of the collateral held by it.*

External turning parameters

These are system-wide parameters or other components which impact the behaviour of the Liquidator.

> ***C_f*:** *Borrowing Collateral Factor - The Borrowing limit of the LP as expressed as a proportion of the value of the collateral held by it.*

Input Parameters

These are variables which will be different each time a Loan Liquidation occurs.

C_v: *ACT Value - The total value of 1 ACT*

C_a: *LP ACT Amount - The amount of ACT held by the Lending Pool as collateral*

L_a: *LP Loan Amount - Value of USDT borrowed from the Lending Pool*

C_a': *LL ACT Amount - Amount of ACT held by the Loan Liquidator as collateral*

L_a': *LL Loan Amount - Value of USDT borrowed from the Loan Liquidator*

Loan Liquidation Criteria

A Loan Liquidation can be triggered if either the loan to the LP or the loan to the Liquidator breaches its borrowing limit. i.e. if:

$$L_a > C_v C_a C_f \quad \text{or} \quad L_a' > CvC_a' C_f'$$

In either case, the Loan Liquidation calculations will be the same.

Loan Liquidation Calculations

We have the C_0; the Collateral Overshoot, which is the amount we want the LP loan to be below its borrowing limit after the Loan Liquidation. However, we also want C_0' - the amount the Liquidator loan should be below its borrowing limit after the liquidation.

We set this to be such that, if there is no other intervention other than both loans accruing interest, both the LP and Liquidator loan will next breach their limits again after the same amount of time has elapsed. This keeps the number of liquidation actions to a minimum.

I.e.:

$$C_0' = C_0 \, I_m$$

Now we can set the target that each loan should be after liquidation as a proportion of the collateral held.

$$T_f = C_f - C_0$$

$$T_f' = C_f' - C_0'$$

To perform the liquidation, there are two values to calculate:

L_d: Amount of USDT to transfer from Liquidator to LP

C_d: Amount of ACT to transfer from LP to Liquidator

After performing these transfers, the LP loan amount should be equal to the target proportion of the collateral value. i.e.

$$(1) \qquad L_a - L_d = C_v T_f (C_a - C_d)$$

And conversely for the Liquidator loan

$$(2) \qquad L_a' + L_d = C_v T_f' (C_a' + C_d)$$

We need to solve these simultaneous equations for L_d and C_d. If we do (1) + (2), the L_d disappears and we have a single equation in terms of C_d. We can rearrange this to get it in terms of C_d and then substitute this back into equation (1). Resulting in:

$$C_d = \frac{(L_a + L_a' - C_v C_a T_f - C_v C_a' T_f')}{C_v (T_f' - T_f)}$$

$$L_d = L_a - C_v T_f (C_a - C_d)$$

Cashing-in liquidated assets

If the security relating to an underlying asset has been enforced and the debt owed by the Borrower recovered by the Real-World Asset Tokenization system, then both the LP and the LL can use the USDT which now backs the ACT to pay off any debts collateralized by that ACT.

To do this, the LP or LL use the Tokenizer to burn the ACT they hold, in return for a portion of the USDT. They can use this USDT to pay off the debt that was backed by that ACT.

The result we are after is that:

- However much is recovered when the security is enforced, the LP should get its entire debt paid off
- If the total USDT from enforcing the security is not enough to pay off the LP debt, then the LL should make up the shortfall
- Any USDT remaining from the enforcement of security after paying off the LP goes to paying off the LL debt.
- If there is any USDT left over after paying off the LL debt, the remainder is returned to the Borrower.

This process is managed by the Loan Liquidator:

1. All ACT transferred to Loan Liquidator (debts to both LL and LP remain unchanged)
2. LL cashes in all the ACT and receives USDT from Tokenizer
3. LL pays off entire debt of LP (even if it means drawing on its own reserves)
4. If there was any USDT left over after paying off the LP debt, it pays off the LL debt
5. If there is any USDT left over after paying off the LL debt, the remainder is handed to the Borrower.

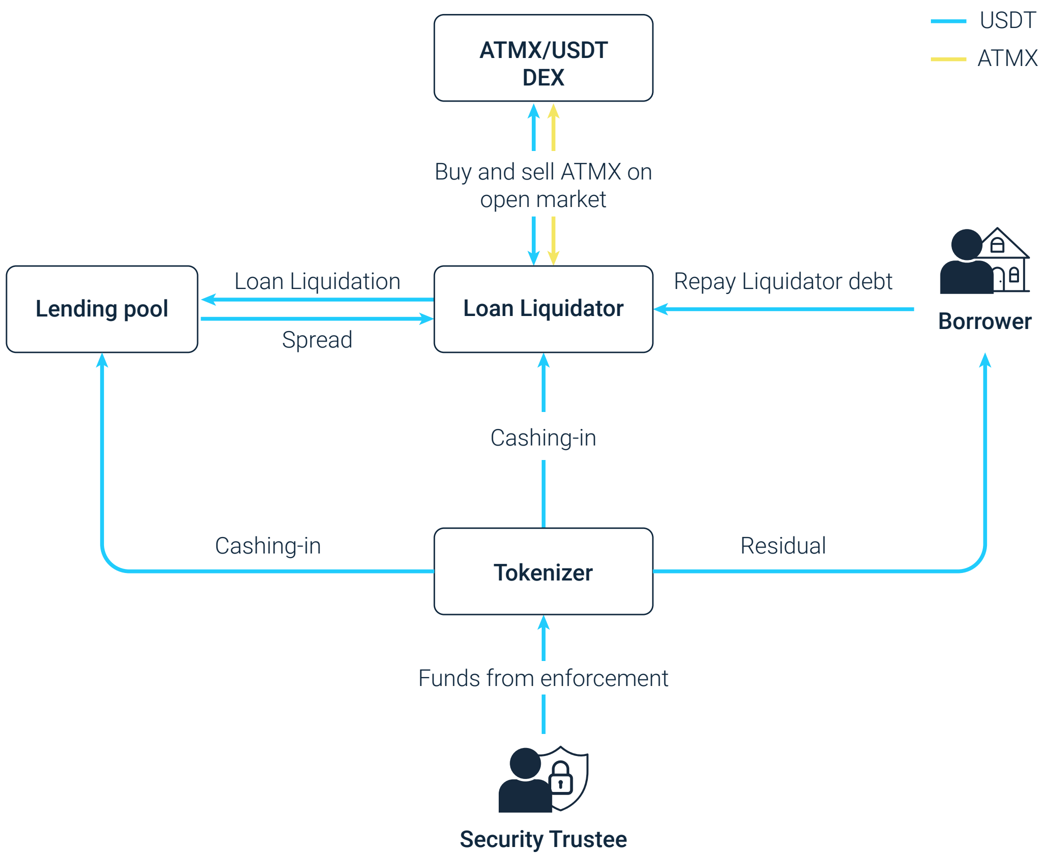
Liquidation Bot

An off-chain bot will keep track of events output by the LP and LL and anticipate when loans are likely to be breaching their limits and ensure Loan Liquidations are triggered in good time.

Similarly, this off-chain bot will keep track of any security being enforced and trigger the LL and LP to cash-in the relevant ACT.

Loan Liquidator Reserves and ATMX

The Loan Liquidator (LL) owns reserves of USDT and ATMX to fund Loan Liquidations. It is useful to understand how tokens flow in and out of this reserve.



- The LL will start off with a reserve of USDT and ATMX.
- Periodically, the LP takes a portion of the interest paid by Borrowers and transfers it to the LL's reserves.
- When the LL performs a Loan Liquidation, USDT is transferred back to the LP (in exchange for a portion of ACT - not shown).
- If the Borrower is in breach and wants to prevent security being enforced against it, they repay the LL, which covers everything spent on loan liquidation up to that point.
- If the Borrower does not do this, then the security taken over the real-world asset is enforced and the amounts recovered following such enforcement are transferred to the Tokenizer as USDT and this replaces the Secured Asset which was evidenced by the ACT.
- Both the LP and LL then cash-in their ACT recouping USDT and transfer any residual USDT to the Borrower.
- Finally, if the level of USDT in the pool gets too low, the Loan Liquidator will exchange some of its ATMX for USDT. If the level of USDT in the pool gets too high then it will exchange some of its USDT for ATMX.

Governance and ATMX

This subsystem deals with usage and distribution of Atomix governance tokens, ATMX. In the ordinary course of business, ATMX will be distributed to Liquidity Providers via the ATMX Distribution Contract. ATMX holders are able to:

1. Change the value of system tuning parameters
2. Upgrade any of the contracts to new code

Distribution mechanism

Liquidity Providers supply USDT liquidity to the Liquidity Vault and receive xUSDT in return. At a later date, they can redeem their xUSDT to get their USDT back plus a return. However, rather than just passively holding on to their xUSDT tokens, Liquidity Providers can choose to provide them to the ATMX Distribution Contract. While this contract holds the xUSDT tokens, the Liquidity Provider earns ATMX token rewards.

When the Liquidity Provider wishes to redeem their xUSDT, they first withdraw it from the ATMX Distribution Contract, after which they will stop earning ATMX rewards, and redeem them with the Liquidity Vault to get their USDT back plus a return. The Liquidity Provider can then use their ATMX to influence the protocol, or sell them on the open market.

ATMX tokens are minted at linear rate and distributed to Liquidity Providers in the ATMX Distribution Contract in proportion to the amount of xUSDT they have provided.

Roadmap

Atomix will initially focus on supporting a limited range of real-world asset types and in the future will broaden the range of asset types supported.

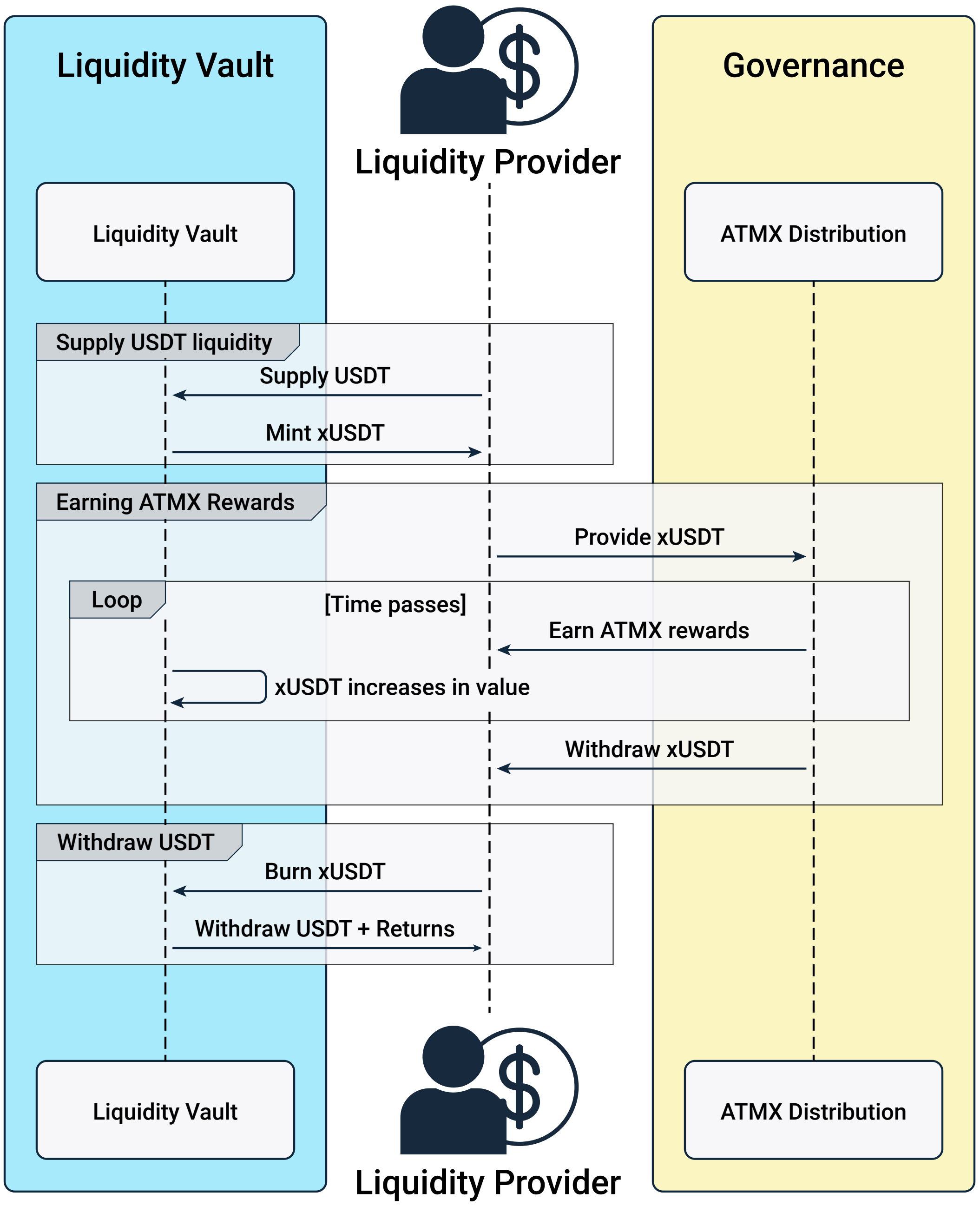
In future it will use more Fetch.ai technology including:

- Using the Fetch.ai ledger for high transaction rates (c.f. Eth level 2 scaling).
- Fetch.ai agents and collective learning acting as oracles for more sophisticated real-time valuations of assets.

Part 2: User Journeys

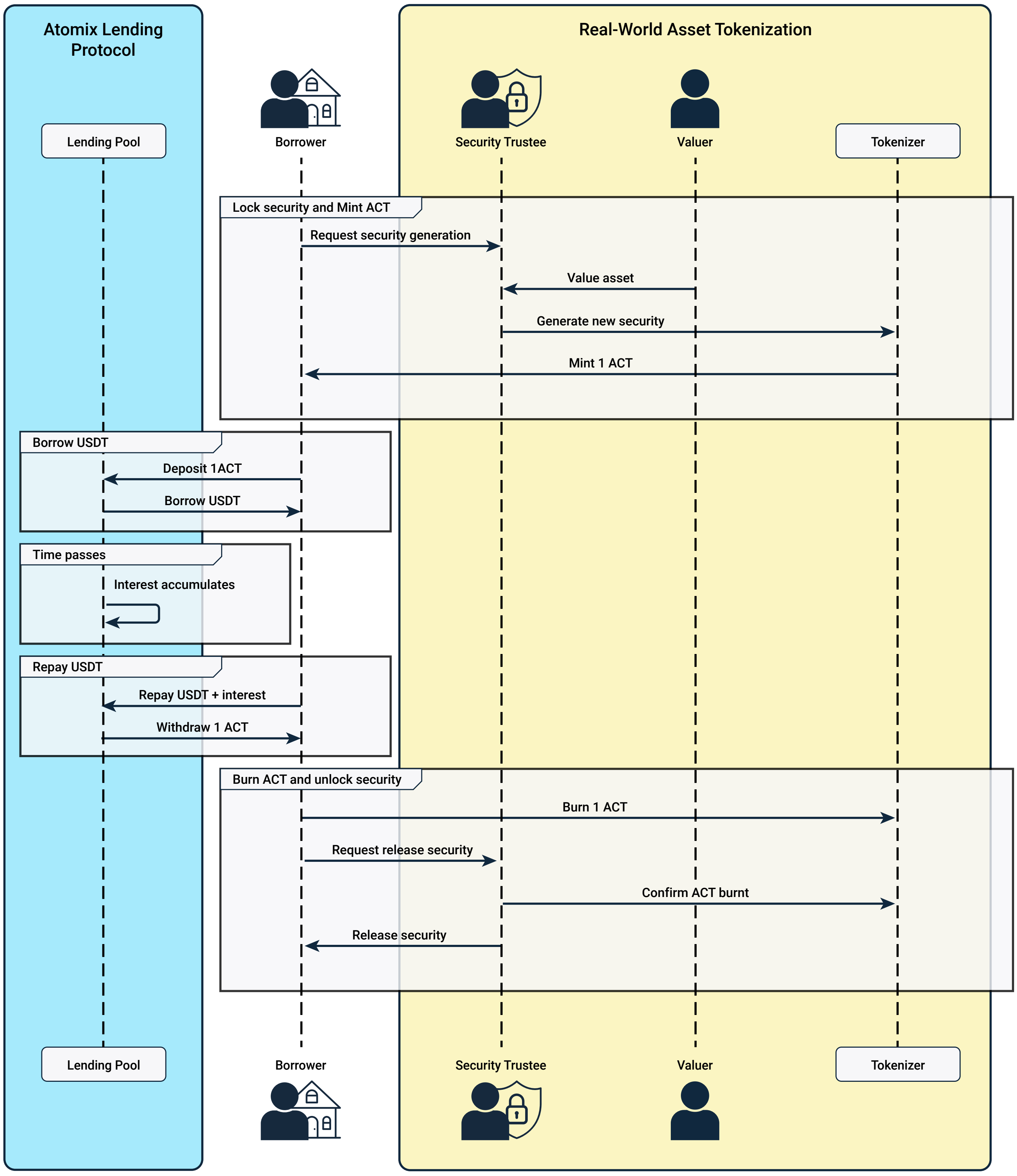
Supplying liquidity

This a standard Liquidity Provider journey.



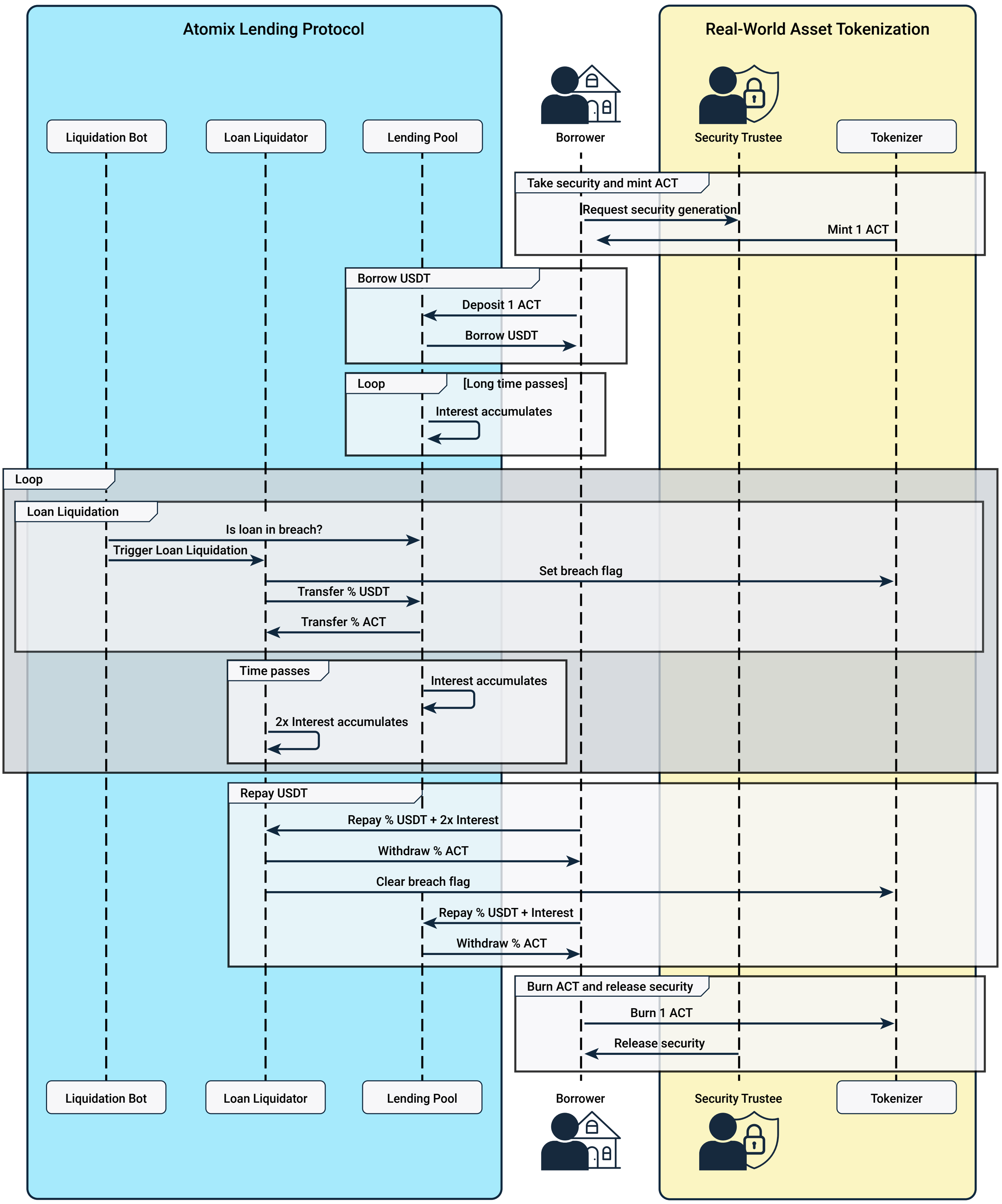
Borrowing and repaying

This shows a standard Borrower journey.



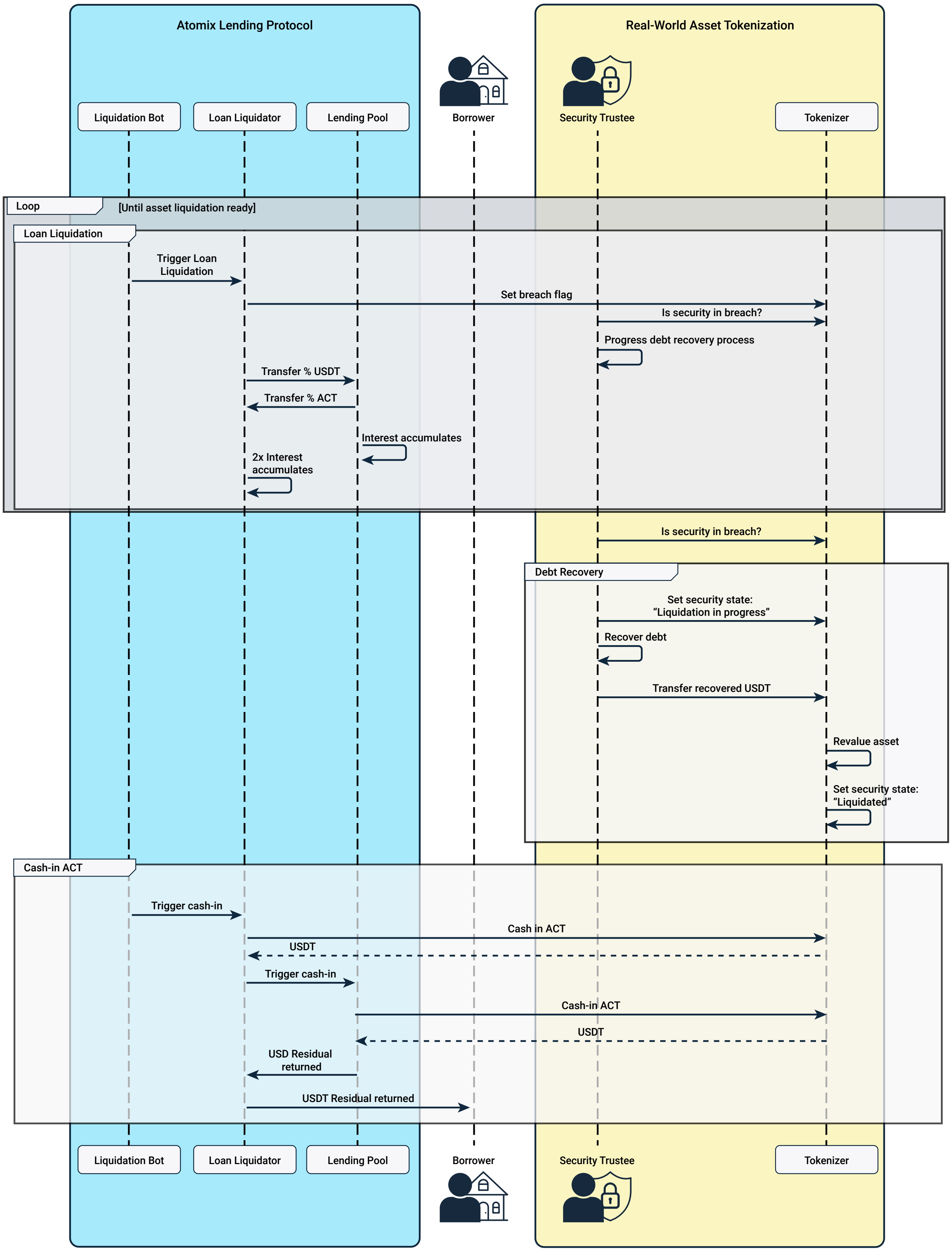
Breaching borrowing limit

This shows a Borrower's journey, but instead of repaying in a timely manner, they let the debt grow until it breaches the borrowing limit. Then after a while they decide to pay off the debt and release the security taken over the asset.



Debt recovery

This diagram shows how a series of loan liquidations result in triggering the debt recovery process. We start the journey from the first loan liquidation.



ATOMIX