

Part A: Initial Definition & Research

A capital-efficient protocol where users deposit liquid staking tokens (like mSOL, jitoSOL) as collateral.

Users mint a stablecoin without losing access to the yield or utility of their collateral.

The deposited assets remain active — they can be staked, lent, or used in LPs via whitelisted strategies.

Yield earned from the collateral can automatically repay the loan or be routed elsewhere.

Designed for DeFi power users, DAOs, and protocols seeking liquidity without idle capital.

1) Core Value Proposition & Product-Market Fit (PMF)

A protocol that allows users to deposit liquid staking tokens (like mSOL, jitoSOL), mint stablecoins, and still use the collateral in DeFi strategies. This creates **liquidity without sacrificing yield**, and even allows **yield to repay debt automatically** — making capital more productive and flexible.

- **Capital Efficiency:** Your assets earn yield *and* let you borrow at the same time.
- **Yield-Aware Loans:** Debt can be repaid from passive income.
- **Composability:** Collateral isn't idle — it's deployed in other strategies.
- **Treasury Utility:** Protocols and DAOs can earn, borrow, and fund operations in one system.

2) Key Target Markets

1. DeFi Power Users / Yield Farmers
2. DAO Treasuries & Protocols
3. Stablecoin Liquidity Seekers

3) Competitor Landscape

UXD Protocol :- Yield isn't composable, users don't access or direct it.

Marginfi :- Borrowing = collateral is locked — can't use it in other strategies

JitoUSD:- Single-asset focused (jitoSOL only)

3) Founder Market Fit

I've been building in the Solana ecosystem using Rust, Anchor, and React — the exact tools needed to create performant, secure, and composable DeFi primitives. This protocol is a natural extension of the types of systems I already understand and ship.

Part B: Process Appendix





Initial Idea

An arithmetic stablecoin protocol where users can deposit less volatile tokens like USDC, SOL, etc.. and have complete access over the collateral deposited to invest further into the whitelisted protocol like lending, staking etc..

AI Prompts

Prompt:- Based on my idea a IDEA:- User deposits low-volatile tokens as collateral, in return, gets a stablecoin. The depositor has control over using the collateral in the whitelisted protocols (lending, staking etc). It's a solution to idle collateral in protocols by making it dynamic., help outline the core value proposition and initial thoughts on product-market fit. What are 2-3 key value areas?

Response:- Your idea is sharp — you're targeting one of the biggest inefficiencies in DeFi: idle collateral. Let's dive into your:

-  Core Value Proposition
 -  Initial thoughts on product-market fit
 -  2–3 key value areas
-  Core Value Proposition

A capital-efficient stablecoin protocol where users deposit low-volatility assets (e.g., wSOL, mSOL, jitoSOL) as collateral to mint a stablecoin, without locking the collateral — instead, they can dynamically use it in whitelisted DeFi strategies (lending, staking, LPing) to earn

yield.

This unlocks:

- Stablecoin liquidity without loss of productive potential.
- Self-repaying loans from yield generated by the collateral.
- Composability with trusted protocols while still ensuring solvency.

💡💡 Initial Thoughts on Product-Market Fit

💡💡 Target users:

- DeFi power users who stake SOL, mSOL, or jitoSOL and want extra stablecoin liquidity. •

Protocols holding large treasuries — want to optimize their idle tokens.

- Yield farmers who want to stay capital efficient.

💡💡 Early market validation signals:

- Similar ideas like Liquity (LUSD) and Alchemix on Ethereum gained huge traction. • Solana has high-yield staking tokens (like jitoSOL) and idle LP vaults — perfect base assets.

💡💡 Key Value Areas

1. 💡💡 Dynamic Collateral Utility

Problem: Today, when users mint stablecoins (like USH or UXD), their collateral sits idle.

Your Value: Your protocol lets users use their collateral while it's locked, earning yield and compounding value.

- Whitelisted protocols (e.g., marginfi, Kamino, Sanctum) are integrated via adapters. •

Protocol auto-tracks position health and yield.

- Users optionally delegate yield to repay debt.

✓ Boosts capital efficiency

✓ Reduces opportunity cost

2. Controlled Risk with Whitelisted Strategy Layer

Problem: Dynamic collateral is risky — it can be rugged or lose value suddenly.

Your Value: Instead of free-form collateral usage, only audited and permissioned yield strategies are allowed. You become the risk gateway.

- Collateral moves into yield strategy wrappers (PDA-controlled).
- Value is monitored in real time.
- If LTV goes too high, auto-liquidation is triggered.

- ✓ Reduces systemic risk
- ✓ Gives confidence to users and protocols

3. ❖❖ Self-Repaying Stablecoin Loans

Problem: Repaying stablecoin loans manually is tedious and often inefficient.

Your Value: Let users opt-in to auto-yield-to-repay mode — where a portion of yield gets routed to burn their debt over time.

- Smart contract tracks yield → applies to debt.
- Debt is repaid passively as long as yield flows in.
- Could show % of loan expected to be repaid over 30/60/90 days.

- ✓ Improves UX for borrowers
- ✓ Reduces liquidation risk over time

❖❖ Bonus Opportunity: Protocol-to-Protocol Vault Mode

Let other protocols deposit their SOL/mSOL idle treasury and mint stablecoins for programmatic liquidity, e.g., DAOs funding contributors using borrowable stablecoins backed by their yield-earning assets.

Want to Go Further?

I can help you next with:

- ❖❖ MVP architecture (vaults, minting, collateral wrappers)
- ❖❖ LTV & risk design model
- ❖❖ Strategy adapter contract pattern
- ❖❖ Frontend + smart contract flow

Just say the word.

Prompt:- now roast this project idea


Response:- Alright, let's roast this idea  — not to kill it, but to *stress test* it.




You're building a system where users:

- Deposit low-volatility tokens (e.g. mSOL, jitoSOL),
- Get a stablecoin in return,
- Still retain the ability to use that collateral in DeFi, and
- Maybe even auto-repay their loan with the yield.

Sounds genius, right?

The Roast: “The DeFi Free Lunch That’s Actually a Bear Trap”

 1. “Use your collateral *and* borrow against it” — Sure, what could go wrong? You're promising users the best of all worlds:

- Borrow liquidity 
- Keep earning yield 
- Avoid liquidation 

But here's the hard truth: You can't have high yield + high leverage + zero liquidation risk. That's like eating cake, not paying for it, and also renting it out to someone else for yield.

 *Eventually someone's going to default — and your protocol is going to be holding a bag of illiquid LP tokens from Kamino.*

 2. The “whitelisted protocols” part is a ticking governance

bomb You want to keep users safe by only allowing pre-approved strategies. But: •

Who maintains the whitelist?

- Who audits each adapter?

- What happens if Kamino or marginfi rug or change token mechanics?
- Who explains to the DAO why a \$1M vault got wiped from “whitelisted exposure”?

💡💡 *You're trading user freedom for protocol liability. Have fun being a centralized DeFi babysitter.*

💡💡 3. It's cold outside for new stablecoins

You want to launch a new stablecoin. Problem is:

- USD tried algo delta-neutral backing (meh).
- USH tried volatility-backed yield (still finding PMF).
- You're going to get crushed on liquidity, trust, and peg mechanics.

💡💡 *Why would I hold your stablecoin instead of USDC or jitoUSD unless I'm yield-maxxing or reckless?*

🏠 4. Self-repaying loans are DeFi's slowest rug

The self-repaying idea sounds nice — but let's be real:

- Yield isn't stable. It drops, it spikes, it goes negative.
- Users stop caring after depositing. If debt isn't cleared, they still get liquidated. •

It's a UX illusion: "You're safe... until you're not."

💡💡 *Alchemix did this on Ethereum and still saw mass liquidations during yield drops. You're not smarter than math.*

💡💡 TL;DR: The harsh truth

You're proposing a protocol that:

- Wants the yield of liquid staking,
- Wants the liquidity of a stablecoin,
- Wants the composability of DeFi,
- Wants the safety of strict governance...

But forgets that DeFi users either:

- Don't care about risk (DGens), or
- Don't trust new protocols (everyone else).

❓❓ Real Talk: It's still worth building, but you need to pick

one: • A yield-maximizing stablecoin (like jitoUSD, built on liquid staking).

- Or a vault system for productive collateral (like Gearbox or Sommelier).
- Or a self-repaying loan primitive with dynamic risk modeling.

Trying to do all three at once? That's protocol suicide.

Want help rewriting this as a leaner MVP that actually ships? Let's trim the fat and harden the core.

Prompt:- For this value proposition A capital-efficient stablecoin protocol where users deposit low-volatility assets (e.g., wSOL, mSOL, jitoSOL) as collateral to mint a stablecoin, without locking the collateral — instead, they can dynamically use it in whitelisted DeFi strategies (lending, staking, LPing) to earn yield.

Stablecoin liquidity without loss of productive potential.

Composability with trusted protocols while still ensuring solvency.

DeFi power users who stake SOL, mSOL, or jitoSOL and want extra stablecoin liquidity. Protocols holding large treasuries want to optimize their idle tokens. Yield farmers who want to stay capital efficient., suggest 2-5 key target demographics or market segments.

Response:- Based on your value proposition, here are 2–5 key target demographics / market segments that are most likely to adopt and benefit from your capital-efficient stablecoin protocol:

1. ❓❓ DeFi Power Users / Yield Maxis

Who they are:

- Users who actively rotate capital between staking, lending, and farming platforms (e.g. marginfi, Kamino, Meteora).
- Typically multi-wallet users with >\$1k–\$50k in staked assets (mSOL, jitoSOL, bSOL, etc).

Why they care:

- They want to retain yield on staked assets while accessing liquidity to farm more. •

The “self-repaying loan” narrative also appeals to them.

2. DAO Treasuries & DeFi Protocols with Idle Capital

Who they are:

- Protocols or DAOs with large treasuries in wSOL, mSOL, or jitoSOL.
- Examples: Mango DAO, Marginfi DAO, UXD DAO, or NFT DAOs holding treasury in LSTs.

Why they care:

- Want to extract liquidity from staked assets while keeping treasury “safe”.
- Enables programmatic yield generation, e.g., funding contributors with borrowed stablecoins.

3. Leverage Farmers & Loopers

Who they are:

- Users already leveraging LSDs by looping deposits + borrows (e.g., via Solend, Kamino). •

They understand LTV, liquidations, and farming risks.

Why they care:

- Your protocol simplifies this by giving them a stablecoin against productive assets, without manual looping.
- Automated strategies with whitelisted targets reduce friction.

4. Web3 Treasurers & Stablecoin Liquidity Seekers

Who they are:

- Builders, fund managers, and founders holding LSTs who want spendable liquidity for operations, salaries, or hedging.

Why they care:

- Can mint stablecoins using mSOL/jitoSOL, without selling or unstaking.
- Reduces tax and opportunity cost from swapping into USDC/USDT.

5. ❓❓ Experimental Protocol Aggregators / Yield Vaults

Who they are:

- Protocols that build on top of primitives (e.g. RoboVault, Sommelier-like projects).

Need building blocks that are modular, composable, and capital-efficient.

Why they care:

- Can plug into your system as a stablecoin layer or collateral backend.
- Yield-bearing, freely usable collateral is a DeFi goldmine if secured properly.

Prompt:- Identify key competitors for a project with this value prop targeting these markets DeFi power users who stake SOL, mSOL, or jitoSOL and want extra stablecoin liquidity. Protocols holding large treasuries want to optimize their idle tokens. Yield farmers who want to stay capital efficient. DAO Treasuries & DeFi Protocols with Idle Capital Leverage Farmers Stablecoin Liquidity Seekers . What are potential weaknesses in their offerings?

Response:- Here's a breakdown of the key competitors for your project, based on the core value proposition:

A capital-efficient stablecoin protocol where users deposit low-volatility, yield-bearing tokens (mSOL, jitoSOL, etc.) as collateral to mint a stablecoin, while retaining composable access to DeFi strategies.

Your target segments are:

- DeFi power users
- DAO treasuries & protocols with idle capital
- Yield-maximizing leverage farmers
- Stablecoin liquidity seekers

◆◆ Key Competitors & Their Weaknesses

◆◆ 1. UXD Protocol

Delta-neutral stablecoin backed by derivatives (short + LST).

✓ Strengths:

- Capital-efficient model using delta-neutral yield.
- Stablecoin (UXD) already live on Solana mainnet.
- Institutional backing and integrations (e.g., Jupiter, Solend).

✗ Weaknesses:

- Not composable: Users can't use collateral for farming or other strategies. •

Relies on external funding or derivatives for sustainability.

- UXD treasury yield is protocol-level, not user-controlled.
- Minting UXD requires access to derivatives platforms (complex backend).

◆◆ 2. Kamino Finance

Smart vaults for leveraged farming and liquidity provisioning.

✓ Strengths:

- Deep integration with Solana DeFi protocols (marginfi, Meteora). •

Efficient auto-leverage vaults for LSTs and LPs.

- Strong TVL growth and composability.

✗ Weaknesses:

- No native stablecoin: Doesn't offer liquidity against collateral.
- Collateral is locked into Kamino vaults — no free use elsewhere.

- No yield-to-repay or self-repaying loan model.
- More of an optimizer, not a lending primitive.

❖❖ 3. Jito (via jitoSOL)

Offers yield-bearing LST with MEV rewards.

✅ Strengths:

- One of the best-performing LSTs in Solana.
- Highly composable across Solana DeFi.
- Integrated into stablecoins like jitoUSD.

❌ Weaknesses:

- Does not offer a native stablecoin — relies on 3rd parties (e.g., Meteora or Marginfi).

No dynamic collateral unlocking or multi-protocol strategies.

- Not a direct competitor, but a critical input asset for your project.

❖❖ 4. Alchemix (Ethereum)

Borrow stablecoins against future yield.

✅ Strengths:

- Pioneered self-repaying loans.
- Deep community awareness.
- Highly composable ERC-20 stablecoin (aIUSD).

❌ Weaknesses:

- Ethereum-based (not on Solana).
- Poor capital efficiency — loans are limited to future yield, not principal.
- Overcomplicated UX, high gas costs.
- Yield sourcing is fragmented and inefficient.

5. LST-backed Stablecoins (jitoUSD, sanctumUSD, marginUSD) Liquid staking protocols launching native stablecoins.

Strengths:

- Native integrations with their own LSTs.
- Backed by productive collateral.
- Low friction for minting.

Weaknesses:

- Collateral is locked — can't use in external DeFi strategies.
- No flexible strategy layer for treasury management.
- No auto-repay, no whitelisted adapters, no DAO governance options.

6. Solend / Marginfi (DeFi Lending Protocols)

Strengths:

- Major TVL holders on Solana.
- Users can borrow against mSOL/jitoSOL.
- Institutional-grade infrastructure.

Weaknesses:

- Isolated lending: Cannot use collateral in other strategies.
- Yield is passive — users must manually manage debt/yield strategies. •

Not composable with a dynamic adapter layer.

- No stablecoin: Borrow USDC/USDT, but that requires selling yield-bearing assets.



Summary Table

Protocol Has
Stablecoin

Composable
Collateral

Yield-to-Re pay





DAO Treasury
Friendly

Weakness

UXD     Static, no DeFi access





Kamino     Locked vaults JitoUSD     No dynamic logic

Alchemix     Not Solana-native

Solend     Legacy lending Your Idea     Execution risk

?? Takeaway: Your Edge

What makes your project different and potentially stronger:

-  Dynamic, composable collateral — not locked, but deployed in approved DeFi strategies •
-  Self-repaying mechanism using real yield
-  DAO / protocol-level use cases for treasury liquidity
-  Capital-efficient model that doesn't sacrifice yield for liquidity