

VerseFi - Complete Ideation Document

Executive Summary

Project Name: VerseFi
Tagline: The Cross-Collateral DeFi Layer for Prediction Markets
Target: Solana Colosseum Hackathon
Category: DeFi Infrastructure

Part 1: Problem Definition

The Core Problem

Prediction markets have a **dead capital problem**:

Current State:



User → Prediction Market → Buy conditional token → Wait months → Redeem (if correct)



Capital is locked with:

- ✗ Zero yield generation
- ✗ No composability
- ✗ No leverage options
- ✗ Can't use as collateral elsewhere
- ✗ Fragmented across multiple markets

Market Size:

- Polymarket: ~\$1.5B locked value
- Drift Prediction Markets: ~\$300M
- Kalshi & others: ~\$200M+
- **Total: \$2B+ in dead capital**

Secondary Problems

Problem 1: Fragmented Liquidity



Same event across multiple platforms:

- Polymarket: Trump 2028 (USDC collateral)
- Drift: Trump 2028 (SOL collateral)
- Different prices, different liquidity
- Users must manually check each platform

Problem 2: Collateral Lock-In



User bought yes-USDC on Polymarket

User now wants SOL exposure too

Options:

- A) Exit position, lose on spread
- B) Deploy NEW capital for SOL position

No way to swap USDC exposure → SOL exposure

Problem 3: No Leverage



Traditional Finance: Can leverage long positions

Prediction Markets: Must lock 100% capital upfront

Missed opportunity for leverage traders

Problem 4: LP Impossible



Prediction market AMMs exist (Polymarket uses CLOB)

But can't be LP earning fees on conditional token swaps

No way to market-make between different collateral types

Part 2: Evolution of Our Thinking

Initial Idea (Rejected)

Concept: Build our own prediction market with multiverse finance primitives

Why We Rejected It:

- ❌ Competing with established players (Polymarket, Drift)
- ❌ Hard to bootstrap liquidity
- ❌ Oracle/resolution challenges
- ❌ Need to create market infrastructure from scratch
- ❌ Long time to market

Second Idea (Partially Rejected)

Concept: Create AMM where we split users' SOL/USDC into yes/no tokens and provide dual-pool liquidity

Issues Discovered:

- Complex UX (users must understand splitting)
- Protocol takes directional risk if not perfectly balanced
- Still competing with existing markets
- Requires handling token creation/resolution

Key Insight That Changed Everything

Realization: Polymarket creates yes-USDC. Drift creates yes-SOL. **We don't need to create anything!**

The Pivot:



Before: Try to BE a prediction market

After: Be the INFRASTRUCTURE LAYER connecting all prediction markets

This is like:

- Uniswap vs exchanges (aggregator + DEX)
- 1inch vs DEXs (aggregator + best execution)
- Aave vs lending pools (universal lending layer)

Part 3: Final Solution - VerseFi

Core Value Proposition

"Turn fragmented prediction market positions into unified, composable DeFi assets"

Three-Layer Architecture



Layer 1: AGGREGATOR

- Query Polymarket, Drift, Kalshi, etc.
- Compare prices across markets
- Route to best execution
- One-click optimal trading



Layer 2: VERSE AMM

- Swap yes-USDC ↔ yes-SOL
- Standard constant product ($k = x \times y$)
- LPs earn trading fees
- Probability-agnostic design
- Cross-collateral liquidity bridge



Layer 3: VERSE LEND

- Borrow yes-USDC using yes-SOL collateral
- Same-verse = zero liquidation cascade risk
- Enable leverage on predictions
- Interest paid in same conditional tokens

Why This Works

1. Leverages Existing Liquidity



We don't create tokens → Markets do

We don't need oracles → Markets have them

We don't bootstrap from zero → Aggregate existing volume

Day 1: Access to \$10M+ daily volume from existing markets

2. Clear Value Addition



For existing markets:

- ✓ More users (we drive traffic)
- ✓ More volume (we facilitate trading)
- ✓ More use cases (DeFi on top of their tokens)

For users:

- ✓ Best prices (aggregation)
- ✓ More flexibility (swap between collaterals)
- ✓ Earn yield (LP fees)
- ✓ Get leverage (borrowing)

3. No Directional Risk



Protocol doesn't hold yes/no tokens

Protocol doesn't bet on outcomes

Protocol only facilitates trading

Revenue: Swap fees + lending interest

Risk: Only smart contract risk

Part 4: Key Innovations

Innovation 1: Probability-Agnostic AMM

Discovery: Probability cancels out in exchange rates

Mathematical Proof:



yes-USDC value = $P(\text{yes}) \times \$1$

yes-SOL value = $P(\text{yes}) \times \text{SOL_price}$

Exchange Rate = $[P(\text{yes}) \times \text{SOL_price}] / [P(\text{yes}) \times \$1]$

= $\text{SOL_price} / 1$

= SOL_price

P(yes) cancels out! ✓

Implication:



We can use STANDARD constant product AMM:

$k = \text{reserve_yes_usdc} \times \text{reserve_yes_sol}$

No need for complex probability-adjusted formulas!

No need to track P(yes) for pricing!

Just track SOL/USDC oracle price for arbitrage opportunities!

This is NOVEL because:

- No one has mathematically proven this for prediction markets
- Enables standard AMM machinery to work on conditional tokens
- Makes implementation simple and auditable

Innovation 2: Zero-Liquidation Lending

Discovery: Same-verse collateral and debt evaporate together

The Mechanism:



User borrows yes-USDC using yes-SOL as collateral

Scenario A: Event resolves YES

- Collateral: yes-SOL → SOL (full value)
- Debt: yes-USDC → USDC (full value)
- Collateralization ratio UNCHANGED
- No liquidation needed

Scenario B: Event resolves NO

- Collateral: yes-SOL → \$0
- Debt: yes-USDC → \$0
- Both evaporate simultaneously
- No underwater protocol

Why This is Safe:



Traditional Lending Problem:

Collateral: ETH (\$2000)

Debt: USDC (\$1000)

ETH crashes to \$800

→ Underwater! Liquidate!

Verse Lending:

Collateral: yes-SOL ($P \times \$150$)

Debt: yes-USDC ($P \times \$1$)

P changes from 0.6 → 0.3

→ Both halve in value

→ Ratio stays 150:1

→ No liquidation!

Collateral: yes-SOL

Debt: yes-USDC

Event resolves NO ($P \rightarrow 0$)

→ Both → 0

→ Ratio: 0/0 (undefined but both worthless)

→ No one owes anyone

This enables safe leverage for the first time in prediction markets!

Innovation 3: Cross-Collateral Aggregation

Discovery: Different markets use different collateral for SAME event

The Opportunity:



Polymarket: "Trump 2028" with USDC

Drift: "Trump 2028" with SOL

Kalshi: "Trump 2028" with USD

User preferences vary:

- SOL holders: Want to keep SOL exposure
- USDC holders: Want stable value
- Traders: Want best execution regardless

We bridge between them!

Market Opportunity:



Without VerseFi:

User must choose ONE market

Misses better prices elsewhere

Can't diversify collateral types

With VerseFi:

Route to best price automatically

Swap between collateral types in AMM

Unified liquidity across all markets

Part 5: User Personas & Journeys

Persona 1: Price-Sensitive Speculator

Profile:

- Wants to bet on Trump winning
- Cares about getting best price
- Doesn't care about collateral type

Journey with VerseFi:



1. Visit VerseFi interface
2. Select event: "Trump 2028 - Yes"
3. Enter amount: \$1,000
4. VerseFi shows:
 - Polymarket: 2,127 yes-USDC (\$0.47 each)
 - Drift: 14.18 yes-SOL (\$70.50 each)
 - Best: Polymarket
5. Click "Buy at Best Price"
6. VerseFi routes to Polymarket
7. User receives: 2,127 yes-USDC

Persona 2: SOL Maximalist

Profile:

- Bullish on both Trump AND SOL
- Wants double exposure
- Prefers to keep capital in SOL

Journey with VerseFi:



1. Visit VerseFi
2. Select: "Trump 2028 - Yes (SOL exposure)"
3. Deposit: 10 SOL
4. VerseFi routes to Drift
5. User receives: ~21.3 yes-SOL
6. If Trump wins AND SOL pumps: Double gains!

Persona 3: Diversified Trader

Profile:

- Wants Trump exposure
- Wants to hedge SOL volatility
- Sophisticated strategy

Journey with VerseFi:



1. Buy yes-USDC from Polymarket (\$500)
2. Buy yes-SOL from Drift (3 SOL)
3. Use VerseFi AMM to rebalance:
 - Swap some yes-USDC → yes-SOL
 - Or vice versa
 - Achieve desired 50/50 balance
4. Now hedged against SOL volatility

Persona 4: Liquidity Provider (Market Maker)

Profile:

- Wants to earn fees
- Believes market is mispriced (sees alpha)
- Doesn't want to take full directional risk

Journey with VerseFi:



1. Deposit to VerseFi:
 - \$10,000 USDC
 - 50 SOL
2. VerseFi executes:
 - Buys yes-USDC from Polymarket
 - Buys yes-SOL from Drift
 - Adds both to AMM pool
3. User receives LP tokens
4. Earns 0.3% on all swaps
5. Expected returns:
 - Base: $P(\text{yes}) \times \text{position value}$
 - Plus: Trading fees (15-30% APY)
 - If believes $P(\text{yes})$ higher than market: Bonus alpha

Persona 5: Leverage Degen

Profile:

- VERY bullish on Trump
- Wants maximum exposure
- Understands liquidation risk

Journey with VerseFi:



1. Buy yes-SOL from Drift (10 SOL worth)
2. Deposit to VerseFi Lending as collateral
3. Borrow yes-USDC (66% LTV)
4. Swap yes-USDC → more yes-SOL in AMM
5. Now has 1.66x leverage
6. (Optional) Repeat for more leverage
7. If Trump wins: Amplified returns
8. If Trump loses: Total loss (but debt evaporates too)

Part 6: Technical Architecture

Smart Contracts (Solana/Anchor)

Contract 1: Aggregator Router



rust

```

pub struct AggregatorRouter {
    pub supported_markets: Vec<MarketIntegration>,
}

pub struct MarketIntegration {
    pub market_id: Pubkey,
    pub name: String,           // "Polymarket", "Drift"
    pub collateral_type: CollateralType, // USDC, SOL
    pub program_id: Pubkey,
    pub api_endpoint: String,
}

pub fn get_best_quote(
    event_id: String,
    amount: u64,
    collateral_preference: Option<CollateralType>
) -> BestQuote;

pub fn execute_trade(
    market: Pubkey,
    event_id: String,
    amount: u64,
) -> Result<TokenAccount>;

```

Contract 2: Verse AMM



rust

```

pub struct VersePool {
    pub verse_id: [u8; 32],    // Hash of event
    pub token_a: Pubkey,      // yes-USDC (from Polymarket)
    pub token_b: Pubkey,      // yes-SOL (from Drift)
    pub reserve_a: u64,
    pub reserve_b: u64,
    pub k: u128,              // Constant product
    pub lp_token_mint: Pubkey,
    pub fee_bps: u16,         // 30 basis points
    pub oracle_sol_usd: Pubkey, // Pyth price feed
}

```

```

pub fn add_liquidity(
    amount_a: u64,
    amount_b: u64,
) -> Result<u64>; // Returns LP tokens

```

```

pub fn remove_liquidity(
    lp_tokens: u64,
) -> Result<(u64, u64)>; // Returns (token_a, token_b)

```

```

pub fn swap_a_to_b(
    amount_in: u64,
) -> Result<u64>; // Returns amount_out

```

```

pub fn swap_b_to_a(
    amount_in: u64,
) -> Result<u64>;

```

Contract 3: Verse Lending



rust

```
pub struct LendingPool {  
    pub verse_id: [u8; 32],  
    pub collateral_token: Pubkey, // yes-SOL  
    pub borrow_token: Pubkey,    // yes-USDC  
    pub total_collateral: u64,  
    pub total_borrowed: u64,  
    pub collateral_ratio: u16,    // 150% = 15000  
    pub interest_rate_bps: u64,  // Annual interest  
}
```

```
pub fn deposit_collateral(  
    amount: u64,  
) -> Result<()>;
```

```
pub fn borrow(  
    amount: u64,  
) -> Result<()>;
```

```
pub fn repay(  
    amount: u64,  
) -> Result<()>;
```

```
pub fn withdraw_collateral(  
    amount: u64,  
) -> Result<()>;
```

Data Flow



User Interface (Next.js)



Aggregator API (queries markets)



Best Price Routing



Execute on Market (Polymarket/Drift)



Tokens arrive in user wallet



User can:

- Hold
- Swap in VerseFi AMM
- Deposit to Lending
- Provide LP

Part 7: Economics & Sustainability

Revenue Streams

1. AMM Swap Fees



Fee: 0.3% per swap

Split: 70% to LPs, 30% to protocol

Volume estimate:

- \$1M daily swaps (conservative)
- $0.3\% \times \$1M = \$3,000/\text{day}$
- Protocol: \$900/day
- Annual: \$328,500

2. Lending Interest Spread



Borrow rate: 10% APY

Deposit rate: 7% APY

Spread: 3%

Volume estimate:

- \$5M borrowed (conservative)

- $3\% \times \$5\text{M} = \$150,000/\text{year}$

3. Total Protocol Revenue



Year 1 (conservative): \$478,500

Year 2 (10x growth): \$4,785,000

LP Economics

Scenario: LP provides \$10,000 liquidity



Position:

- 50% yes-USDC (from Polymarket at \$0.47)

- 50% yes-SOL (from Drift at \$70.50)

Revenue sources:

1. Trading fees: 15-30% APY

2. Probability alpha: If LP believes $P(\text{yes}) > \text{market price}$

Expected returns:

- Conservative (no alpha): 15% APY = \$1,500/year

- With alpha (correct directional bet): 50-100% ROI

- Risk: If event resolves against LP: Lose principal

Part 8: Competitive Analysis

vs. Prediction Markets Directly

Feature	Polymarket	Drift	VerseFi
Create markets	✓	✓	✗
Trade	✓	✓	✓ (routes)
Best price	✗	✗	✓
Cross-collateral	✗	✗	✓
LP fees	✗	✗	✓
Leverage	✗	✗	✓

Relationship: Symbiotic, not competitive

- We drive volume to them
- We make their tokens more useful
- We're infrastructure, not replacement

vs. DeFi Protocols

Feature	Uniswap	Aave	VerseFi
Conditional tokens	✗	✗	✓
Cross-market	✗	✗	✓
Same-verse safety	N/A	✗	✓
AMM	✓	✗	✓
Lending	✗	✓	✓

Positioning: "Uniswap + Aave for prediction markets"

Part 9: Go-to-Market Strategy

Phase 1: Launch (Weeks 1-4)

Focus: Prove concept



- ✓ Deploy contracts to devnet
- ✓ Integrate Polymarket + Drift
- ✓ Launch with 1 high-profile event ("Trump 2028")
- ✓ Seed \$50K liquidity
- ✓ Get first 100 users

Phase 2: Growth (Months 2-3)

Focus: Scale usage



- ✓ Add more markets (Kalshi, etc.)
- ✓ Launch 10+ events
- ✓ Reach \$1M TVL
- ✓ 1,000+ users
- ✓ \$10M+ monthly volume

Phase 3: Ecosystem (Months 4-6)

Focus: Build moat



- ✓ Enable anyone to add new markets (permissionless)
 - ✓ Launch governance token
 - ✓ Partnerships with prediction market platforms
 - ✓ Mobile app
 - ✓ Advanced features (options, structured products)
-

Part 10: Risk Analysis

Technical Risks

Risk 1: Smart Contract Bugs

- Mitigation: Extensive testing, audit before mainnet
- Severity: High
- Likelihood: Medium

Risk 2: Oracle Manipulation

- Mitigation: Use multiple oracles, delay resolution
- Severity: High
- Likelihood: Low

Risk 3: Market Integration Breaks

- Mitigation: Fallback to other markets, monitoring
- Severity: Medium
- Likelihood: Medium

Market Risks

Risk 1: Low Adoption

- Mitigation: Strong GTM, clear value prop, incentives
- Severity: High
- Likelihood: Low (solving real problem)

Risk 2: Regulation

- Mitigation: Comply with existing framework, legal counsel
- Severity: High
- Likelihood: Medium

Risk 3: Market Dominance by One Player

- Mitigation: Multi-market strategy, still valuable as aggregator
- Severity: Medium
- Likelihood: Low

Part 11: Success Metrics

For Hackathon

Technical:

- ✓ Working contracts on devnet
- ✓ Integration with 2+ prediction markets
- ✓ Live AMM with real trades
- ✓ Functional lending protocol

Product:

- ✓ Clean UI showing value prop
- ✓ Demo with real scenarios
- ✓ Clear documentation

Innovation:

- ✓ Novel mathematical proofs
- ✓ First cross-collateral prediction market aggregator
- ✓ Production-ready code

Post-Hackathon (6 months)

Traction:

- \$5M+ TVL
- 5,000+ users
- \$50M+ monthly volume
- Integration with 5+ markets

Revenue:

- \$50K+ monthly revenue
- Path to profitability clear

Ecosystem:

- 2+ projects building on top
 - Partnership with major prediction market
-

Part 12: Why This Wins

Technical Excellence

✓ Novel mathematical proof (probability cancellation) ✓ Clean architecture (3 modular contracts) ✓ Solana-native (Anchor, Pyth, efficient) ✓ Production-ready code quality

Product-Market Fit

✓ Solves real problem (\$2B dead capital) ✓ Clear value prop (aggregation + DeFi) ✓ Large TAM (prediction markets growing fast) ✓ Multiple revenue streams

Feasibility

✓ Can build in 4 weeks (reasonable scope) ✓ Leverages existing infrastructure ✓ Clear milestones ✓ Demo-able with real impact

Innovation

✓ First-ever cross-collateral prediction market aggregator ✓ Academic-quality mathematical proofs ✓ Enables new use cases (leverage, LP, swaps) ✓ Composability unlocks ecosystem

Team Execution

✓ Clear problem understanding ✓ Iterative thinking (multiple pivots) ✓ Strong technical design ✓ Practical approach (leverage existing vs build from scratch)

Conclusion

VerseFi represents a fundamental shift in how prediction markets work:

From: Isolated, single-collateral, non-composable positions

To: Unified, cross-collateral, DeFi-composable assets

We're not building another prediction market.

We're building the financial infrastructure layer that makes ALL prediction markets 10x more useful.

This is our winner. 🚀