AURION PROTOCOL

Cross-Protocol Credit Aggregation and Risk Control Layer

Technical Whitepaper

Version 1.0

Abstract

Aurion introduces a non-custodial credit layer that operates above existing DeFi money markets such as Aave and Compound. Unlike traditional lending protocols that combine liquidity provision with risk management, Aurion separates these functions to unlock superior capital efficiency and cross-protocol credit portability. The protocol aggregates user positions across multiple lending venues into unified credit accounts, enforces sophisticated risk controls through mandatory routing architecture, and introduces delegated credit pools that provide balance sheet guarantees without deploying capital into lending markets. This architecture enables institutional-grade credit infrastructure while maintaining full protocol composability and user sovereignty over assets.

Table of Contents

1. Introduction

The decentralized finance ecosystem has evolved sophisticated mechanisms for pricing and allocating liquidity through automated money markets. However, these protocols conflate two distinct functions: liquidity provision and credit risk underwriting. This conflation creates inefficiencies that limit capital utilization and prevent the emergence of portable, cross-protocol credit infrastructure.

Aurion addresses this architectural limitation by introducing a meta-layer protocol that sits above existing lending markets. Rather than replacing Aave, Compound, or other established protocols, Aurion enhances them by providing unified credit orchestration, sophisticated risk management, and institutional-grade credit guarantees through delegated capital pools.

The protocol achieves this without taking custody of user assets, lending pool capital, or accrued interest. Instead, Aurion maintains authority only over credit state, risk parameters, and the routing of borrow requests. This design ensures that Aurion remains composable with existing DeFi infrastructure while introducing new primitives for credit allocation and risk management.

1.1 Key Innovations

- **Cross-Protocol Credit Aggregation:** Users maintain a single credit identity that spans multiple lending protocols, enabling portfolio-level risk assessment and credit allocation.
- **Mandatory Router Architecture:** All borrowing occurs through protocol-controlled routers that enforce risk rules before executing transactions on underlying protocols.
- **Delegated Credit Infrastructure:** Third-party capital providers can allocate credit guarantees to users without deploying funds into lending pools, creating a market for balance sheet risk separate from liquidity provision.
- **Non-Custodial Design:** The protocol never takes possession of collateral, borrowed assets, or interest payments, eliminating an entire class of custody risks.

2. Problem Statement

Current DeFi lending protocols face fundamental limitations that constrain their evolution toward institutional-grade credit infrastructure:

2.1 Fragmented Credit Identity

Users interact with multiple lending protocols independently, resulting in siloed positions that cannot be evaluated holistically. A user with substantial collateral on Aave and Compound is treated as two separate entities with no shared credit capacity. This fragmentation prevents optimal capital allocation and creates redundant overcollateralization requirements.

2.2 Conflation of Liquidity and Credit Risk

Existing protocols require liquidity providers to simultaneously absorb credit risk. This coupling prevents specialized market participants from focusing on their core competencies—liquidity providers want predictable, low-risk returns, while credit underwriters seek risk-adjusted yield through selective exposure. Neither group can operate efficiently when forced to participate in both functions.

2.3 Limited Risk Management Tools

Protocols like Aave and Compound offer no mechanisms for pre-emptive risk control beyond basic health factor monitoring. Liquidations occur reactively after positions become undercollateralized. There is no infrastructure for credit scoring, graduated risk tiers, or proactive position management that could reduce liquidation events and protect both borrowers and lenders.

2.4 Absence of Delegated Credit Markets

Traditional finance separates loan origination from balance sheet risk through mechanisms like credit guarantees, insurance, and securitization. DeFi lacks equivalent infrastructure. Capital allocators cannot provide credit enhancements to borrowers without directly lending funds, and borrowers cannot access additional leverage by securing third-party credit guarantees.

2.5 Protocol Integration Barriers

Most lending protocols lack hooks or callbacks that would enable external systems to enforce custom risk logic. This architectural constraint means that sophisticated credit infrastructure must either fork entire protocols or operate completely separately. Neither approach enables the composability required for sustainable innovation.

3. System Architecture

Aurion operates as a meta-layer protocol positioned above existing lending markets. It does not replace or fork these protocols; instead, it provides orchestration, risk management, and credit enhancement services that augment their capabilities.

3.1 Architectural Principles

3.1.1 Non-Custodial Operation

Aurion never takes custody of:

- User collateral deposited on underlying protocols
- Borrowed principal from lending pools
- Interest accrued on any positions

The protocol only maintains custody of:

- Credit state and risk parameters stored in protocol contracts
- Delegated credit guarantee funds held in isolated pools
- Protocol fees collected from borrowing operations

3.1.2 Protocol Agnostic Design

The architecture uses protocol adapters to interface with different lending markets. Each adapter implements a standardized interface while handling protocol-specific integration details. This design enables Aurion to support any lending protocol that exposes standard borrow and collateral query functions, without requiring modifications to those protocols.

3.1.3 Mandatory Router Control

Since underlying protocols like Aave and Compound do not support execution hooks or callbacks, Aurion enforces control through a mandatory routing architecture. Users must execute borrow operations through Aurion routers, which validate credit eligibility before calling underlying protocol functions. This design leverages the principle of voluntary participation—users who want access to Aurion's credit infrastructure agree to route transactions through its control layer.

3.2 System Diagram

The following diagram illustrates the relationship between users, Aurion components, and underlying lending protocols:

```
│                     User                       │
└────────────────┬────────────────────────────────┘
                 │
    ┌────────────▼────────────────┐
    │   Credit Account (Aurion)  │
    │  - Aggregated collateral   │
    │  - Total debt              │
    │  - Credit score            │
    │  - Delegated credit        │
    └────────────┬────────────────┘
                 │
    ┌────────────▼────────────────┐
    │    Credit Router (Aurion)  │
    │  - Validates eligibility   │
    │  - Enforces risk limits    │
    └──────┬──────────────┬────────┘
           │              │
    ┌──────▼─────┐  ┌────▼────────┐
    │    Aave    │  │  Compound   │
    │ (Collateral│  │(Collateral) │
    │  + Debt)   │  │  + Debt)    │
    └────────────┘  └─────────────┘
```

4. Core Components

4.1 Credit Manager

The Credit Manager serves as the central state coordinator for all credit accounts. It maintains the authoritative record of each user's aggregated credit position and enforces protocol-wide risk invariants.

Primary Functions:

- Account creation and lifecycle management
- Aggregated collateral and debt tracking across protocols
- Credit score calculation and update
- Delegated credit allocation and tracking
- Risk tier assignment

4.2 Credit Router

The Credit Router is the enforcement mechanism that ensures all borrowing operations comply with Aurion's risk framework. It acts as a gatekeeper between users and underlying lending protocols.

Core Responsibilities:

- Pre-transaction credit validation
- Protocol adapter selection and invocation
- Post-transaction state synchronization
- Emergency freeze enforcement

4.3 Protocol Adapters

Each supported lending protocol requires a dedicated adapter that translates Aurion's standardized interface into protocol-specific function calls. Adapters handle variations in parameter encoding, return value formats, and state query mechanisms.

Adapter Interface:

- getCollateralValue(user, asset): Query user's collateral on the protocol
- getDebtValue(user, asset): Query user's outstanding debt
- executeBorrow(user, asset, amount): Execute borrow on behalf of user
- getHealthFactor(user): Query protocol-specific health metrics

4.4 Credit Pools

Credit Pools represent the protocol's key innovation in separating liquidity from credit risk. These pools accept capital from Credit Providers who seek to earn fees by underwriting balance sheet risk without deploying funds into active lending positions.

Pool Characteristics:

- Capital remains in the pool and is never lent
- Funds serve as loss-absorbing buffers for guaranteed credit
- Providers earn fees from borrowers who utilize delegated credit
- Pools can be risk-stratified with different fee tiers

4.5 Risk Oracle

The Risk Oracle provides real-time price feeds, volatility metrics, and protocol health indicators required for accurate risk assessment. It aggregates data from multiple sources to ensure robustness against manipulation.

4.6 Insurance Fund

The Insurance Fund serves as the protocol's backstop against tail risk events. It absorbs losses that exceed both user collateral and delegated credit guarantees, protecting Credit Providers and ensuring system stability during extreme market conditions.

5. Credit Account Model

Each user participating in Aurion is represented by a Credit Account—a unified abstraction that aggregates all positions across supported lending protocols. This account serves as the single source of truth for determining borrowing eligibility and enforcing risk limits.

5.1 Account State Structure

Each Credit Account maintains the following state:

Field	Description
totalCollateralValue	USD value of all collateral across Aave, Compound, and other supported protocols
totalDebtValue	USD value of all outstanding borrows across all protocols
delegatedCreditValue	Total credit guarantees allocated from Credit Pools
creditScore	Numerical score based on historical repayment behavior and position health
riskTier	Classification tier (Conservative, Moderate, Aggressive) determining max LTV and fees
protocolPositions	Mapping of protocol addresses to position details (collateral, debt per asset)

5.2 Aggregation Logic

The Credit Manager queries each protocol adapter to retrieve current position data. Collateral and debt values are normalized to USD using oracle price feeds, then summed to produce aggregate metrics:

```
totalCollateralValue = Σ (protocol_collateral_i × price_i ×
haircut_i)

totalDebtValue = Σ (protocol_debt_i × price_i)
```

Haircuts are applied to collateral values to account for asset volatility and liquidity risk. More volatile assets receive larger haircuts to ensure conservative risk assessment.

5.3 Credit Score Calculation

Credit scores are computed based on multiple factors:

- Account age and transaction history
- Historical health factor maintenance (lower scores for accounts that have approached liquidation)
- Repayment consistency and timing
- Collateral quality and diversification
- Protocol-level liquidation events (negatively impacts score)

Scores range from 0-1000, with higher scores qualifying users for improved borrowing terms, lower fees, and increased delegated credit allocation.

6. Borrow Flow and Routing

The borrowing process in Aurion follows a strictly enforced sequence that ensures risk validation occurs before any funds are disbursed. This section details the exact execution flow for a typical borrow operation.

6.1 Pre-Requisites

Before initiating a borrow through Aurion, users must:

- Have deposited collateral directly on Aave or Compound
- Created a Credit Account through Aurion's Credit Manager
- Granted approval for Aurion's router to execute transactions on their behalf

6.2 Detailed Borrow Sequence

Step 1: User Initiates Borrow Request

The user calls the Credit Router contract with parameters specifying:

- Target protocol (Aave or Compound)
- Asset to borrow (e.g., USDC)
- Borrow amount
- Interest rate mode (for Aave: stable or variable)

Function signature:

```
creditRouter.borrow(protocol, asset, amount, rateMode)
```

Step 2: Credit Validation

The Credit Router invokes the Credit Manager to validate eligibility:

```
credit_manager.validateBorrow(user, protocol, asset,
amount)
```

The Credit Manager performs the following checks:

- Query current collateral values across all protocols via adapters
- Calculate aggregated LTV including existing debt plus requested amount
- Verify aggregated LTV does not exceed maximum for user's risk tier
- Check that protocol-specific health factor remains above safety threshold
- Confirm delegated credit backing is sufficient for requested exposure
- Verify no emergency freeze conditions are active

If any validation check fails, the transaction reverts with a specific error code. No state changes occur.

Step 3: Borrow Execution via Adapter

Upon successful validation, the Credit Router selects the appropriate protocol adapter and executes the borrow:

For Aave:

```
aaveAdapter.executeBorrow(asset, amount, rateMode,
referralCode, onBehalfOf=user)
```

For Compound:

```
compoundAdapter.executeBorrow(cToken, amount)
```

The adapter invokes the underlying protocol's borrow function on behalf of the user. Borrowed funds are transferred directly to the user's wallet—Aurion never takes custody.

Step 4: State Synchronization

After successful execution, the Credit Router updates internal accounting:

```
credit_manager.recordBorrow(user, protocol, asset, amount)
```

This updates the user's Credit Account state to reflect increased debt. The protocol also:

- Emits a BorrowExecuted event for indexing and analytics
- Updates fee accumulator for the relevant Credit Pool
- Increments protocol-level utilization metrics

6.3 Critical Enforcement Mechanism

The mandatory router architecture works because users who want access to Aurion's credit enhancement features (higher LTV, delegated credit, portable credit score) must voluntarily route their borrows through Aurion contracts. Users can still interact with Aave or Compound directly, but they forfeit Aurion benefits. This creates a strong incentive for compliance without requiring protocol-level integration.

7. Delegated Credit Pools

Delegated Credit Pools represent Aurion's core innovation: the separation of credit risk underwriting from liquidity provision. These pools enable third-party capital providers to earn yield by assuming balance sheet risk without deploying funds into lending markets.

7.1 Operational Mechanics

7.1.1 Pool Funding

Credit Providers deposit capital into isolated pool contracts. Unlike traditional lending pools where deposits are immediately lent out, funds in Credit Pools remain idle until they are required to cover losses. The pools function purely as collateral buffers that backstop borrower obligations.

7.1.2 Credit Allocation

Pool capital is allocated to borrowers as credit guarantees. When a borrower with $100k in collateral receives $20k in delegated credit, they can borrow as if they had $120k in collateral. The $20k credit guarantee is not a loan—it is a conditional promise to absorb losses if the borrower's position deteriorates.

7.1.3 Loss Absorption Waterfall

In the event of borrower default, losses are absorbed in the following sequence:

1. 1. Borrower's own collateral on underlying protocols
2. 2. Insurance Fund contributions
3. 3. Delegated Credit Pool capital

This structure ensures Credit Providers are protected by multiple layers of cushion before their capital is at risk.

7.2 Fee Structure

Borrowers who utilize delegated credit pay access fees independent of the underlying protocol's interest rate. These fees are calculated as:

```
credit_fee = (delegated_credit_utilized / total_borrow) ×
base_fee_rate × time
```

Fees are distributed according to a predetermined split:

Recipient	Allocation	Purpose
Credit Providers	70%	Compensation for assuming credit risk
Insurance Fund	20%	Backstop reserve

		accumulation
Protocol Treasury	10%	Protocol development and operations

7.3 Risk Stratification

Credit Pools can be configured with different risk parameters to cater to varying risk appetites among Credit Providers:

- Conservative Pools: Lower fee rates, stricter borrower eligibility, minimal leverage
- Moderate Pools: Balanced risk/return, standard credit score requirements
- Aggressive Pools: Higher fees, serve lower credit score borrowers, accept higher LTV ratios

8. Risk Management Framework

Aurion's risk framework operates at two levels: it maintains conservative buffers to ensure underlying protocols never observe unsafe positions, while simultaneously enabling higher effective leverage for users through delegated credit. This dual-layer approach requires sophisticated invariant enforcement and continuous position monitoring.

8.1 Fundamental Invariants

Invariant 1: Aggregate Solvency

For every Credit Account, at all times:

```
totalCollateralValue + delegatedCreditValue ≥
totalDebtValue
```

This ensures that even in extreme scenarios, the combination of user collateral and Credit Pool backing is sufficient to cover outstanding debt.

Invariant 2: Protocol-Level Safety

For each protocol position (Aave, Compound), the protocol's native health factor must remain above a safety threshold:

```
protocol_health_factor ≥ min_safe_health_factor
```

This prevents Aurion from ever pushing a user's position into a state where the underlying protocol would trigger liquidation. The safety buffer ensures Aurion can intervene before protocol-level liquidations occur.

Invariant 3: Credit Pool Coverage

Total allocated delegated credit cannot exceed available pool capital:

```
Σ (delegatedCreditAllocated) ≤ totalPoolCapital +
insuranceFund
```

8.2 Risk Parameters by Tier

Risk Tier	Max LTV	Credit Fee	Min Credit Score
Conservative	75%	2% APR	700
Moderate	85%	4% APR	500
Aggressive	95%	7% APR	300

8.3 Continuous Monitoring

The Risk Oracle continuously monitors:

- Collateral price movements and volatility spikes

- Protocol-specific health factors for all active positions
- Aggregate system utilization across Credit Pools
- Market conditions that might trigger cascading liquidations

When risk thresholds are breached, the protocol can activate graduated interventions including increased monitoring frequency, borrowing restrictions, or mandatory collateral top-ups.

9. Liquidation and Collateral Management

Aurion cannot directly control liquidation mechanisms on Aave or Compound, as these protocols execute liquidations autonomously when positions breach health factor thresholds. Instead, Aurion implements pre-emptive risk management to prevent positions from ever reaching liquidation conditions on underlying protocols.

9.1 Pre-Emptive Risk Control Tools

9.1.1 Dynamic Borrow Caps

As market volatility increases or collateral values decline, Aurion reduces maximum permissible borrowing capacity. This prevents users from taking additional leverage that would push positions closer to liquidation thresholds.

9.1.2 Collateral Top-Up Requirements

When a position's health factor falls below a warning threshold (but remains above protocol liquidation levels), Aurion can require users to deposit additional collateral within a grace period. Failure to comply results in borrowing restrictions until the position is reinforced.

9.1.3 Credit Recall Mechanism

In extreme scenarios, Aurion can recall delegated credit from at-risk positions. This forces borrowers to either reduce debt or provide additional collateral to compensate for the withdrawn credit guarantee. Credit recall is executed gradually with advance notice to minimize disruption.

9.1.4 Incentivized Repayment

Users who voluntarily reduce leveraged positions when flagged by the risk system receive fee discounts or credit score bonuses. This creates positive incentives for proactive risk management rather than relying solely on punitive measures.

9.2 Loss Realization Process

If despite all preventive measures a position is liquidated on Aave or Compound, Aurion's loss realization follows this sequence:

4. 1. Underlying protocol liquidators seize collateral and repay debt
5. 2. Aurion monitors for liquidation events via contract event logs
6. 3. Remaining deficit (debt exceeding recovered collateral) is calculated
7. 4. Insurance Fund absorbs first layer of deficit
8. 5. If deficit exceeds Insurance Fund, relevant Credit Pools absorb proportional losses

9. 6. User's Credit Account is marked with default flag, severely impacting credit score

This structured approach ensures that while Aurion cannot prevent liquidations, it maintains clear accountability and loss distribution that protects Credit Providers while preserving system integrity.

10. Fee Structure and Economics

Aurion's economic model is designed to align incentives between borrowers, Credit Providers, and the protocol itself. Fees are charged independently of underlying protocol interest rates, creating a distinct revenue stream that compensates participants for credit infrastructure services.

10.1 Credit Access Fees

Borrowers pay credit access fees calculated based on the proportion of their borrowing capacity derived from delegated credit:

```
annual_fee = (delegated_credit_used / total_debt) ×
tier_base_rate
```

For example, a user with $100k total debt comprising $70k backed by their own collateral and $30k enabled by delegated credit pays fees only on the $30k delegated portion.

10.2 Fee Distribution

All collected fees are distributed according to the following allocation:

- 70% to Credit Providers (proportional to capital allocated)
- 20% to Insurance Fund (automatic accumulation)
- 10% to Protocol Treasury (development and operations)

Distribution occurs continuously via automated accounting rather than batch settlements, ensuring real-time fee accrual without gas-intensive distribution transactions.

10.3 Economic Sustainability

The protocol is designed to be economically self-sustaining without inflationary token emissions. Revenue from credit access fees funds:

- Credit Provider yields competitive with alternative DeFi opportunities
- Insurance Fund growth sufficient to cover tail risk scenarios
- Protocol development, security audits, and operational costs
- Risk Oracle infrastructure and maintenance

11. Governance and Protocol Evolution

Aurion implements a hybrid governance model that balances decentralized decision-making with expert risk oversight. This structure recognizes that while broad stakeholder input is valuable for strategic direction, certain risk parameters require specialized expertise to manage safely.

11.1 Governance Structure

11.1.1 Token Holder Governance

AURION token holders participate in governance by voting on:

- Protocol fee structure and distribution percentages
- Addition of new supported lending protocols
- Treasury fund allocation and grants programs
- Risk Council member selection and replacement

11.1.2 Risk Council

The Risk Council consists of elected experts who have authority over time-sensitive risk parameters:

- Asset-specific collateral haircuts and LTV limits
- Emergency freeze activation during market dislocations
- Credit Pool risk tier assignments
- Oracle source selection and fallback mechanisms

Risk Council actions are subject to bounded discretion—they cannot modify core protocol invariants or violate solvency guarantees. All actions are publicly logged and can be vetoed by token holder governance if they exceed mandated authority.

11.2 Upgrade Mechanism

Protocol upgrades follow a timelocked governance process:

10.1. Proposal submission with technical specification
11.2. Community discussion period (minimum 7 days)
12.3. Token holder vote (requires quorum and supermajority)
13.4. Timelock delay (48 hours minimum) before execution
14.5. Execution by authorized multisig or automated timelock contract

12. Security and Formal Verification

Given Aurion's role as critical credit infrastructure, security is paramount. The protocol employs multiple layers of defense including formal verification, extensive testing, and conservative design principles.

12.1 Core Security Principles

- Non-custodial architecture eliminates entire classes of custody-related vulnerabilities
- Explicit invariants enforced at every state transition
- Fail-safe defaults: transactions revert rather than proceeding with invalid state
- Isolated component design prevents cascading failures

12.2 Formal Verification Approach

Critical protocol components undergo formal verification using multiple tools:

Certora Prover

Used to verify that core invariants hold under all possible execution paths. Specifications cover solvency conditions, credit allocation limits, and fee distribution accuracy.

Foundry Invariant Testing

Automated fuzz testing continuously validates invariants against randomized transaction sequences. Test suites execute millions of operations to discover edge cases that manual testing might miss.

Scribble Instrumentation

Runtime assertions embedded in smart contract code provide continuous validation during development and testing. These assertions are removed in production deployments but inform formal specification development.

12.3 Audits and Ongoing Security

Prior to mainnet launch, the protocol will undergo:

- Multiple independent security audits from tier-1 firms
- Public bug bounty program with substantial rewards
- Staged rollout beginning with testnet, then mainnet with capped total value
- Continuous monitoring and incident response procedures

13. Implementation Roadmap

Aurion's development follows a phased approach that prioritizes core functionality, rigorous testing, and gradual feature expansion.

13.1 Phase 1: Foundation (Months 1-4)

Scope

- Aave integration only (single protocol support)
- Credit Account implementation
- Credit Router with basic risk validation
- Conservative risk engine (no credit scoring, simple LTV limits)
- Basic oracle integration

Deliverables

- Core contracts deployed to Arbitrum testnet
- Initial test suite with invariant coverage
- Documentation and developer guides

13.2 Phase 2: Expansion (Months 5-8)

Scope

- Compound integration (multi-protocol aggregation)
- Delegated Credit Pools implementation
- Credit scoring system deployment
- Risk tier system with differentiated parameters
- Insurance Fund mechanisms

Deliverables

- Multi-protocol position aggregation functional
- Credit Provider interface and pool management
- Initial security audit completion
- Beta launch on Arbitrum mainnet with TVL caps

13.3 Phase 3: Maturity (Months 9-12)

Scope

- Tradable credit positions (tokenization of credit capacity)
- DAO and institutional fund account types
- Structured credit products and tranches
- Advanced risk analytics and dashboards

- Full governance activation

Deliverables

- Feature-complete protocol launch
- Removal of TVL caps after stability period
- Comprehensive formal verification completion
- Token distribution and governance transition

13.4 Why Arbitrum

Arbitrum is the optimal deployment environment for Aurion due to:

- Deep Aave and Compound liquidity already established
- Active DeFi user base familiar with lending protocols
- Low transaction costs enabling frequent position monitoring
- Growing institutional participation requiring credit infrastructure
- Strong ecosystem support and developer tooling

14. Conclusion

Aurion introduces a fundamental architectural innovation to decentralized finance: the separation of liquidity provision from credit risk underwriting. By operating as a meta-layer above existing lending protocols, Aurion enables functionality that would be impossible within those protocols' native architectures.

The protocol's key contributions include:

- Cross-protocol credit aggregation that creates unified, portable credit identities
- Delegated credit infrastructure enabling capital-efficient risk markets
- Mandatory routing architecture that enforces sophisticated risk controls without protocol forks
- Non-custodial design that eliminates an entire class of security vulnerabilities

Unlike previous attempts to build on top of existing DeFi protocols, Aurion does not require cooperation or modification from underlying platforms. It leverages voluntary participation—users who want enhanced credit features route through Aurion; those who prefer vanilla interactions can continue using protocols directly.

This design creates sustainable moats through network effects: as more Credit Providers allocate capital and more borrowers build credit histories, the protocol becomes increasingly valuable to all participants. Credit scores become portable across protocols, delegated credit guarantees enable higher effective leverage than competitors can offer, and cross-protocol risk management reduces overall system fragility.

Aurion represents the next evolution in DeFi credit markets—moving beyond simple money market primitives toward institutional-grade credit infrastructure that maintains the composability and permissionless innovation that define decentralized finance. By enabling specialized market participants to focus on their core competencies, Aurion unlocks capital efficiency improvements that benefit the entire ecosystem.

The path forward is clear: start with solid fundamentals on a single protocol, expand to multi-protocol aggregation, and ultimately build the credit layer that DeFi has always needed but never had the architecture to support.

Appendix: Technical Specifications

A. Smart Contract Architecture

Core Contracts

- CreditManager.sol - Central account state and validation
- CreditRouter.sol - Borrow request gateway
- CreditAccount.sol - Per-user position abstraction
- CreditPool.sol - Delegated credit capital management
- InsuranceFund.sol - Protocol backstop mechanism

Protocol Adapters

- AaveAdapter.sol - Aave V3 integration
- CompoundAdapter.sol - Compound V3 integration
- IProtocolAdapter.sol - Standard adapter interface

Risk Infrastructure

- RiskOracle.sol - Price feeds and volatility data
- RiskEngine.sol - LTV calculations and tier management
- CreditScoring.sol - Score calculation and history

B. Key Invariants

```
1. totalCollateralValue + delegatedCreditValue ≥ totalDebtValue

2. protocol_health_factor[i] ≥ min_safe_health_factor ∀ protocols i

3. Σ(allocated_credit) ≤ pool_capital + insurance_fund

4. user_ltv ≤ max_ltv[risk_tier]
```

C. External Dependencies

Dependency	Purpose	Fallback Strategy
Chainlink Price Feeds	Asset pricing	Secondary oracle + circuit breaker
Aave Protocol	Lending venue	Pause new borrows if unavailable
Compound Protocol	Lending venue	Pause new borrows if unavailable

For more information, visit aurion.finance