ULAL™

Utility-Linked Automatic Liquidity™

A Protocol for Adoption-Driven Token Economics

THE THREE PILLARS

Pillar 1: AVB™ — Adoption-Verified Burn™
Pillar 2: The Utility Hold™
Pillar 3: The Philosophical Mandate

Robert H. Rose
Creator, ULAL™ Protocol
Founder, AI Agent Mark™ LLC
St. Louis, Missouri — March 2026

TABLE OF CONTENTS

A Note on Authorship

The protocol design, economic architecture, smart contract implementation, and proof of concept described in this white paper are entirely my own. The writing was developed collaboratively with Claude, Anthropic's AI assistant, working from my concepts, mathematical framework, and working Sepolia deployment.

This is no different from an author directing a team of writers, or an executive working with a communications professional to articulate a vision. The direction, the accountability, and the conviction behind every claim in this document are mine. Claude executed the writing under my direction.

I live in the truth of my reality. I am open about how this document was produced because transparency is the foundation of everything ULAL™ stands for.

— Robert H. Rose, March 2026

ABSTRACT

Every token economic model in existence ties value to one of three things: trading activity, treasury decisions, or speculative demand. None of them answer the most fundamental question in token economics:

> *How do you prove that a token has value because people actually use it — not because people hope others will buy it?*

ULAL™ — Utility-Linked Automatic Liquidity — is a protocol that answers this question mathematically, autonomously, and permanently.

At its core is AVB™ — Adoption-Verified Burn — a two-layer proof of utility mechanism where every real-world service consumption event simultaneously funds protocol operations, deepens a decentralized liquidity pool, and permanently removes tokens from circulation.

ULAL™ is built on three pillars:

- **Pillar 1 — AVB™ (Adoption-Verified Burn):** The mathematical mechanism. A two-layer architecture that provably burns tokens at every price point with zero deficit.
- **Pillar 2 — The Utility Hold™:** The onboarding mechanism. Prepaid consumptive credits, capped per user, restricted to service consumption during Tranche 1.
- **Pillar 3 — The Philosophical Mandate:** The governance principle. Tokenization must be tied to utility — enforced by smart contract logic that cannot be modified after deployment.

No human intervention. No treasury votes. No speculative demand required.

ULAL™ is designed for businesses built on forecastable consumption events — services, manufacturing, commodities, marketplaces, and digital goods. Any business where customers pay for something they use can implement the protocol. ULAL™ does not support and cannot be applied to appreciating assets, speculative holdings, or models where capital is required before the first customer transaction occurs. This is not a limitation — it is the Philosophical Mandate expressed as an architectural constraint. If the business model is "produce something, sell it, customers use it," ULAL™ powers it. If the business model is "buy this and hope it goes up," ULAL™ rejects it — by design, at the protocol level.

Beyond its technical innovation, ULAL™ addresses a structural inequity in how new enterprises are funded. Traditional capital markets require investment before adoption — forcing entrepreneurs to surrender equity in exchange for resources to discover whether customers exist. ULAL™ inverts this sequence. Every customer transaction simultaneously funds operations, builds liquidity, and strengthens the token economy. The first customer is the capital raise. The entrepreneur retains full ownership. No venture funding required. No permission needed. No geographic gatekeeping. ULAL™ is built for builders.

The proof is on the chain. Chain ID 20260320. March 20, 2026.

SECTION 1 — THE PROBLEM

1.1 The Speculation Trap

Since the emergence of ERC-20 tokens in 2017, the dominant token economic model has been speculation-driven: a project creates a token, promises future utility, sells tokens to investors, and hopes adoption follows investment.

The result is a system where token value is determined by what people expect others to pay — not by what people actually use. When expectations collapse, value collapses. When speculation ends, the token ends.

The DeFi ecosystem has produced sophisticated variations on this model — liquidity mining, yield farming, protocol-owned liquidity, revenue buybacks — but none have solved the fundamental problem:

> *Token value derived from speculation is indistinguishable from a Ponzi scheme until proven otherwise. And proof only comes after the fact.*

1.2 The Liquidity Problem

Decentralized exchanges require liquidity pools to function. Without deep liquidity, tokens cannot be traded without significant price impact — creating a chicken-and-egg problem: deep liquidity requires large capital deposits; large capital deposits require investor confidence; investor confidence requires proven adoption; proven adoption requires accessible trading.

Projects solve this by funding liquidity pools manually — a one-time founder deposit that depletes over time, or through liquidity mining programs that incentivize speculation rather than utility. None of these solutions tie liquidity depth to real adoption.

1.3 The Capital Problem

Every entrepreneur faces the same path: build a prototype, pitch investors, give up equity, burn the capital searching for product-market fit, raise again at the cost of more equity, and repeat until either the company succeeds or the founder owns a minority stake in their own vision.

This model has a fundamental flaw: it requires capital before there are customers. The entrepreneur must convince investors that people will use the product before anyone has used it. The investment is a bet on future adoption — and the cost of that bet is ownership.

For well-connected founders in major markets with access to venture capital networks, this path is difficult but navigable. For the independent builder — the self-taught developer in St. Louis, the solo entrepreneur in Lagos, the engineer in Medellín — this path is often closed entirely. Not because the idea is bad. Not because the market doesn't exist. Because the capital gatekeepers are elsewhere, and they don't return cold emails.

The result is a global economy where viable products die unfunded — not for lack of demand, but for lack of access to the capital required to discover that demand.

What if the product itself could fund its own growth? What if every customer transaction simultaneously funded operations, built liquidity, and strengthened the token economy — automatically, from transaction one, with no investor required?

1.4 The Missing Primitive

What DeFi has never had is a mechanism that:

- Automatically deepens liquidity as utility grows
- Burns tokens in provable proportion to real demand
- Requires no human intervention at any step
- Works mathematically at every token price point
- Rewards early adopters without requiring speculation

Until now.

SECTION 2 — THE ULAL™ PROTOCOL

2.1 Definition

Utility-Linked Automatic Liquidity (ULAL™) is a token economic protocol where every real-world utility transaction autonomously splits payment revenue into operational funding, decentralized liquidity provision, and token burn — at the smart contract level, with no human intervention required.

2.2 The Three Pillars

Pillar 1 — AVB™ (Adoption-Verified Burn) is the mathematical engine. It ensures that every service consumption event triggers a provable, zero-deficit token burn at any token price. AVB™ is detailed in Section 3.

Pillar 2 — The Utility Hold™ is the onboarding mechanism. It requires early adopters to purchase prepaid consumptive credits sized to practical usage, with functional restrictions that prevent speculation during Tranche 1. The Utility Hold™ is detailed in Section 4.

Pillar 3 — The Philosophical Mandate is the governance principle. Tokenization must be tied to utility — not as a marketing position, but as an architectural constraint enforced by immutable smart contract code. The Philosophical Mandate is detailed in Section 5.

2.3 Core Properties

- **Adoption-linear:** Liquidity grows proportionally with real usage — not price speculation
- **Non-reflexive:** Pool depth correlates with registered users — not market sentiment
- **Autonomous:** Every step executes via smart contract — no treasury votes, no human decisions
- **Transparent:** All flows visible on-chain — no private float, no hidden mechanics
- **Stabilizing:** Fixed operating fee creates predictable, stable liquidity inflow

2.4 Why ULAL™ Is Different

Model	Liquidity Source	Human Required	Adoption-Linked
Trading fee recycling	Speculation	No	No
Treasury buyback	Revenue	Yes	Partial
Protocol-owned liquidity	Bonding	No	No
Fee-on-transfer	Holder transfers	No	No
Liquidity mining	Speculation	No	No
ULAL™	Utility consumption	No	YES — mathematically

SECTION 3 — PILLAR 1: AVB™ (ADOPTION-VERIFIED BURN™)

3.1 Definition

AVB™ (Adoption-Verified Burn™) is the two-layer proof of utility burn mechanism at the foundation of ULAL™. It solves a problem that has never been formally addressed in token economics:

> *How do you structure a token burn so that it remains mathematically valid at every possible token price point?*

Every previous burn mechanism fails this test at some price level. AVB™ solves it permanently through a two-layer architecture.

3.2 The Mathematical Problem AVB™ Solves

Consider a naive burn design where registration costs a fixed dollar amount and a percentage of payment buys and burns tokens:

> *Registration = $1.00 | 30% buys AMT | At $0.10/AMT → buys 3 AMT | Required burn: 10 AMT | Deficit: 7 AMT*

Algebraically: if registration = k × P (where P = token price), and burn fee = x% of registration:

> *Burn fee (dollars) = x% × k × P AMT purchased = (x% × k × P) ÷ P = x% × k*

The price variable P cancels out entirely. The deficit is permanent and structural — not a rounding error. A successful token destroys its own burn mechanic. AVB™ solves this permanently.

3.3 The Two-Layer Solution

The following parameters illustrate the two-layer solution using the first AVB™ implementation: AI Agent Mark™, which sells AI Agent Passports™ — cryptographic identity credentials for personal AI agents. In this implementation, the “registration” is the purchase of a new AI Agent Passport, and the associated token is AMT (AgentMark Token). *These specific dollar amounts, token quantities, and cap values are implementation-specific choices, not protocol constants.* Any project implementing ULAL™ sets its own Layer 1 fee, Layer 2 burn amount, and dollar cap appropriate to its service and target market (see Section 9.2 for the calibration framework).

Layer 1 — Operating Fee (fixed dollar amount)

- Amount: $1.00 fixed — does not float with token price
- Split: 70% to protocol operations ($0.70) / 30% to ULAL™ liquidity pool ($0.30)
- Purpose: Funds the business. Grows liquidity linearly with adoption.

Layer 2 — Burn Fee (token-denominated, dollar-capped)

- Amount: 10 tokens at market price
- Cap: $10.00 maximum dollar equivalent

- Below cap (token < $1.00): always burns exactly 10 tokens
- Above cap (token > $1.00): burns $10.00 worth of tokens
- Total registration cost: always between $2.00 and $11.00

3.4 Mathematical Proof — Zero Deficit At Every Price

The following table uses fees associated with the AI Agent Mark Passport™.

Token Price	Op Fee	Burn Fee	Total	Tokens Burned	Cap	Deficit
$0.01	$1.00	$0.10	$1.10	10.0	No	ZERO
$0.10	$1.00	$1.00	$2.00	10.0	No	ZERO
$0.50	$1.00	$5.00	$6.00	10.0	No	ZERO
$0.99	$1.00	$9.90	$10.90	10.0	No	ZERO
$1.00	$1.00	$10.00	$11.00	10.0	Threshold	ZERO
$1.01	$1.00	$10.00	$11.00	9.9	Yes	ZERO
$2.00	$1.00	$10.00	$11.00	5.0	Yes	ZERO
$5.00	$1.00	$10.00	$11.00	2.0	Yes	ZERO
$10.00	$1.00	$10.00	$11.00	1.0	Yes	ZERO
$100.00	$1.00	$10.00	$11.00	0.1	Yes	ZERO

Ten price points. Ten zeros. The math is bulletproof.

This proof was verified on Ethereum Sepolia testnet on March 21–22, 2026. Result: 10/10 confirmed. Zero deficit at every price point. All systems green.

3.5 The Cap — Why A Maximum Is Strategic

The following table compares AI Agent Passport™ registration costs to comparable digital identity services. Other ULAL™ implementations will have different cost ceilings determined by their specific parameters.

Digital Identity Cost	Price Range	Frequency
.com domain registration	$10–15/year	Annual
.io domain	$30–50/year	Annual
Ethereum ENS name	$5–640+/year	Annual
SSL certificate	$10–100/year	Annual
ULAL™-powered registration	$2.00–$11.00	ONE-TIME

A one-time cost of $2.00–$11.00 is competitively positioned against recurring annual fees across every comparable digital identity model.

3.6 AVB™ as a Protocol Primitive

AVB™ is not specific to any single implementation. It is a protocol-level mechanism that any project with real-world utility consumption can deploy. The mathematical properties hold regardless of the implementing project, the specific token, or the service being consumed. The only requirements are: a real-world service with consumable utility, an ERC-20 token with burn capability, a Uniswap V2 or V3 liquidity pool, and a price oracle.

SECTION 4 — PILLAR 2: THE UTILITY HOLD™

4.1 Definition

The Utility Hold™ is a prepaid consumptive credit mechanism within the ULAL™ protocol. It requires early adopters to purchase a minimum balance of tokens sized to practical future usage — not speculation — with functional restrictions that limit token use exclusively to the implementing protocol's intended service during Tranche 1.

The Utility Hold™ is consumptive utility. Tokens purchased under a Utility Hold™ can only be used for their designated service function. No swapping. No secondary market. No other use. The smart contract enforces this at the code level — it is not a policy, it is an architectural constraint.

Implementation Example — AI Agent Mark™: In the first ULAL™ implementation, the Utility Hold™ restricts AMT tokens to AI Agent Passport™ registration and verification. A user who purchases 50 AMT under the Utility Hold™ can register 5 agents or perform 50 verifications — and nothing else — until Stage 2 unlocks.

4.2 The Business Precedent

The Utility Hold™ is not a new concept. It is a well-established business practice applied to token economics with one critical improvement: transparency.

Prepaid consumptive credits are common across industries as a method to grow a customer base, ensure repeat usage, and fund operations through customer prepayment:

- **Starbucks mobile app:** Requires users to load a minimum dollar balance before purchasing. Users cannot sell their Starbucks balance on an exchange. They cannot do anything with their credits except buy coffee. As of 2023, Starbucks held approximately $1.8 billion in unredeemed prepaid balances.
- **Transit cards:** Metro systems worldwide (Oyster, Suica, Clipper) require prepaid loading. The balance can only be used for transit.
- **Gaming platforms:** Xbox, PlayStation, and Steam wallets require credit loading. Funds cannot be withdrawn or transferred. They can only be spent within the platform ecosystem.
- **Laundromat card systems:** Commercial laundromats use prepaid cards that can only be used on machines within that facility. The customer loads $20. They use it to wash clothes. The card has no value outside the laundromat.

In every case, the customer prepays for future consumption of a specific service. The balance has no investment purpose. The business benefits from the float. The customer benefits from convenience and access.

The Utility Hold™ follows this identical pattern — with one fundamental difference.

4.3 The Transparency Difference

Dimension	Traditional Prepaid	Utility Hold™
Visibility of float	Private. Held in corporate accounts. Customers cannot see how funds are deployed.	Public. Every token is visible on the Ethereum blockchain. Anyone can verify balances, pool depth, and burn history.
Who benefits	The corporation. Starbucks earns interest on $1.8B in customer prepayments. Private financial benefit.	The ecosystem. Utility Hold™ balances deepen the ULAL™ liquidity pool, stabilizing the token economy for all participants.
Accountability	Quarterly SEC filings report deferred revenue in aggregate. No individual tracing.	Every individual token balance is verifiable on-chain. Every pool deposit is traceable. Every burn is permanent and public.
Expiration	Many prepaid balances subject to inactivity fees, expiration, or escheatment. Funds can be lost.	No expiration. Tokens remain the user's property indefinitely. They do not degrade, expire, or get absorbed.
Reversibility	Corporation can change terms, devalue rewards, or modify the program unilaterally.	Smart contract logic is immutable once deployed. Rules cannot be changed after deployment — not by anyone.

4.4 Mechanics

Minimum Purchase

Each ULAL™ implementation defines a minimum Utility Hold™ sized to practical usage of its specific service. The minimum is not arbitrary — it is calibrated to a reasonable quantity of future service consumption.

Parameter	Protocol-Level Rule	Rationale
Minimum purchase	Defined per implementation	Sized to practical usage, not speculation
Per-user cap	Defined per implementation	Prevents speculative accumulation
Functional restriction	Service use only during Tranche 1	Smart contract enforced — no swapping, no transfer
Stage 2 unlock trigger	Pool depth threshold	Tied to collective adoption, not managerial decision
Unlock mechanism	Automatic — smart contract	No human intervention, no board vote, no discretion

Implementation Example — AI Agent Mark™

Parameter	Value	Rationale
Minimum purchase	50 AMT ($5.00 at launch)	Covers 5 passport registrations
Per-user cap	50 AMT	Prevents speculative accumulation
Immediate burn	10 AMT (first registration)	User must consume to enter Tranche 1
Remaining hold	40 AMT	Locked for future registrations/verifications
Stage 2 unlock	ULAL™ pool reaches $50,000	~167,000 registrations at $0.30/registration
At unlock	Use for service, or swap via pool	Swap is secondary — not primary purpose

4.5 The Per-User Cap: Why It Matters

Economic Integrity: Without a cap, a single actor could purchase millions of tokens under the Utility Hold™ label, wait for Stage 2 unlock, and dump them on the ULAL™ pool — destabilizing the entire system. The per-user cap prevents any individual from accumulating a position that could damage the ecosystem.

Legal Defensibility: The cap demonstrates that the Utility Hold™ is sized to actual service consumption, not investment accumulation. A 50 AMT cap covers 5 registrations — a reasonable quantity for practical use. It does not cover 500 or 5,000. The cap is calibrated to use, not to profit.

This mirrors the Starbucks precedent: Starbucks limits prepaid card loads. A customer cannot load $100,000 onto a Starbucks card. The limit exists because the purpose is coffee, not capital storage. The Utility Hold™ cap exists because the purpose is service consumption, not token accumulation.

4.6 The Cabbage Patch Defense

If token value increases between Tranche 1 purchase and Stage 2 unlock, early adopters benefit from appreciation. This is a feature, not a vulnerability — and it has clear legal precedent.

In 1983, Coleco manufactured Cabbage Patch Kids as toys at a retail price of approximately $25. Within months, secondary market prices reached $250 or more — a 10x appreciation. No regulator classified Cabbage Patch Kids as securities. No enforcement action was taken against Coleco. The product was manufactured, marketed, and sold as a consumable good. The fact that demand outstripped supply and secondary prices rose was a consequence of genuine demand — not a designed investment scheme.

The same principle applies to tokens under the Utility Hold™:

- Manufactured purpose: Service consumption (registration, verification)
- Marketed purpose: Prepaid credits for future utility
- Sold as: Consumptive access to a specific protocol service

- Price appreciation: A possible consequence of adoption demand — not the designed purpose

The Utility Hold™ is stronger than the Cabbage Patch precedent in one critical respect: Cabbage Patch Kids survived to be resold. Tokens under AVB™ are destroyed upon use. A speculative vehicle cannot be constructed around an asset that ceases to exist at the moment of consumption.

4.7 Stage 2: The Unlock

Stage 2 is triggered automatically when the ULAL™ pool reaches a predefined depth threshold. This trigger is:

- Objective: A verifiable on-chain dollar value, not a subjective decision
- Automatic: The smart contract monitors pool depth and executes unlock without human intervention
- Collective: The threshold is reached through aggregate adoption — no single actor can trigger it
- Irreversible: Once triggered, Stage 2 cannot be reversed

At Stage 2, tokens previously restricted under the Utility Hold™ gain the additional option of being swapped via the ULAL™ pool. This is documented as a secondary feature — not the primary value proposition. The primary purpose remains service consumption.

What Stage 2 does NOT do:

- Does not create a speculative trading market — the ULAL™ pool is a liquidity mechanism, not an exchange
- Does not change the burn mechanism — AVB™ continues at the same rate
- Does not alter the operating fee — Layer 1 continues unchanged
- Does not require any action from the protocol creator — the smart contract executes autonomously

What Stage 2 does:

- Provides token holders with optionality: continue consuming services, or exit via the ULAL™ pool
- Signals network maturity — the pool has reached sufficient depth for stable swaps
- Enables the transition to Tranche 2 (Reg D 506(c)) for accredited investors

SECTION 5 — PILLAR 3: THE PHILOSOPHICAL MANDATE

5.1 The Principle

> *Tokenization must be tied to utility.*

This is not a marketing position. It is not a compliance strategy designed to evade regulatory scrutiny. It is an architectural conviction — enforced by smart contract code that cannot be modified after deployment.

Every design decision in the ULAL™ protocol flows from this single principle:

- **AVB™** burns tokens on use — because value should come from consumption, not speculation
- **The Utility Hold™** restricts tokens to service use during Tranche 1 — because early access should require real participation, not financial positioning
- The per-user cap limits accumulation — because the purpose is use, not hoarding
- The operating fee is fixed — because revenue should correlate with adoption, not token price
- The cost ceiling exists — because access should never become prohibitive
- Stage 2 is automatic — because unlocking should depend on collective achievement, not a boardroom decision

5.2 Why This Matters

The cryptocurrency industry has suffered a credibility crisis driven by projects that use the language of utility to disguise speculation. Tokens are called "utility tokens" in white papers while being marketed as investment opportunities on social media. The language says utility. The architecture says speculation.

ULAL™ inverts this. The architecture says utility. The smart contracts enforce utility. The burn mechanism proves utility. If the language ever drifted toward speculation, the code would contradict it — because the code cannot be changed.

This is the purpose of the Philosophical Mandate as a formal pillar of the protocol: it establishes that ULAL™'s utility-first design is not incidental, not a side effect, and not a compliance afterthought. It is the foundational architectural decision from which every other design choice derives.

5.3 Enforcement Through Architecture

The Philosophical Mandate is not enforced by policy, by terms of service, or by the goodwill of the protocol creator. It is enforced by immutable smart contract code:

- **AVB™ burns are atomic:** The burn occurs in the same transaction as the service consumption. There is no step where a user holds tokens in a speculative state during the burn flow.
- **Utility Hold™ restrictions are contract-level:** The smart contract rejects swap attempts during Tranche 1. This is not a UI restriction that could be bypassed — it is an on-chain constraint.

- **Stage 2 triggers are autonomous:** The PoolMonitor contract reads pool depth directly from the Uniswap pair. No human can accelerate or delay the unlock.
- **Operating fees are hardcoded:** The 70/30 split and $1.00 fixed fee are constants in the TwoLayerRegistrar contract. They cannot be changed after deployment.

The protocol creator cannot violate the Philosophical Mandate even if they wanted to. This is the strongest possible form of commitment: not a promise, but a mathematical impossibility.

5.4 The Gift and the Protocol

The first ULAL™ implementation — AI Agent Mark™ — submitted its identity standard to NIST on March 19, 2026, as a free, open standard for AI agent identity. The standard was given to the world deliberately. This was not an oversight. It was a strategy rooted in the Philosophical Mandate.

The standard is the gift — to every developer, every entrepreneur, every person building a personal AI agent. Free. Open. NIST-submitted.

ULAL™ is the protocol that transforms adoption of that gift into mathematical, provable, autonomous value.

> *We didn't charge for the door. We built the engine behind it.*

The gift creates the demand. The protocol captures the value. And the Philosophical Mandate ensures that capture mechanism always serves real utility — never speculation.

5.5 Built for Builders

ULAL™ is not designed for projects that have already raised $50 million in venture capital. Those projects have liquidity. They have runway. They have market makers on retainer.

ULAL™ is designed for the entrepreneur who has an idea, the ability to build it, and customers willing to use it — but no access to traditional capital markets.

Consider the traditional funding path and what ULAL™ replaces at each step:

Traditional Path:

> *Idea → Prototype → Pitch VCs → Give up 20–40% equity → Burn capital seeking product-market fit → Raise again (more dilution) → Hope adoption follows investment*

ULAL™ Path:

> *Idea → Prototype → First customer pays for service → ULAL™ activates: → $0.70 funds operations (no VC required) → $0.30 builds liquidity (no market maker required) → Token burn creates deflationary pressure (no speculative demand required) → Second customer compounds all three effects → Every subsequent customer IS the capital raise*

No equity dilution. The entrepreneur retains full ownership. There is no Series A, no board seat given to an investor who has never met a customer, no liquidation preference that pays investors before founders. The business funds itself through revenue from the first transaction forward.

No permission required. ULAL™ does not require approval from a venture partner, an investment committee, or an accredited investor network. It requires one thing: a customer who wants the service. The smart contract handles the rest.

No runway clock. Traditional startups measure survival in months of remaining capital. A ULAL™-powered business has no runway clock because every transaction is self-funding. The business is sustainable from its first dollar of revenue — not from its first dollar of investment.

No geographic gatekeeping. A builder in any city, in any country, with internet access and a real service to offer can deploy ULAL™. The protocol does not care where the entrepreneur went to school, who they know, or whether they can fly to Sand Hill Road for a pitch meeting.

This is not a theoretical benefit. The first ULAL™ implementation — AI Agent Mark™ — was built by a solo entrepreneur in St. Louis, Missouri with no venture funding, no institutional backing, and no capital raise. The standard was submitted to NIST. The smart contracts were deployed to Sepolia. The AVB™ mechanism was proven at 10/10 price points. All of it funded by the builder's own effort and designed to sustain itself from the first customer forward.

ULAL™ does not eliminate the need for capital at scale. Tranche 2 exists specifically to bring accredited investors into the ecosystem once adoption has been demonstrated with real revenue. But it inverts the sequence: adoption comes first, investment follows. The entrepreneur proves demand before offering equity — if they choose to offer equity at all.

The protocol is built for the builder who doesn't have a warm introduction to Sequoia. It is built for the developer who learned to code from YouTube. It is built for the entrepreneur who has been told "come back when you have traction" and never had the capital to get there.

ULAL™ is the traction engine. The first customer starts it. The smart contract runs it. The builder keeps what they built.

SECTION 6 — POOL DYNAMICS AND ECONOMIC MODELING

The following economic models use AI Agent Mark™ parameters as the reference implementation. Other ULAL™ implementations will produce different pool growth curves based on their specific Layer 1 fee and transaction volume.

6.1 Linear Growth Model

Under AVB™, the ULAL™ pool receives a fixed $0.30 per registration from Layer 1. This is a deliberate stabilizing design choice.

Registrations	ULAL™ Deposits	Pool Depth	Stage 2 Progress
1,000	$300	Seed	0.6%
10,000	$3,000	Functional	6%
50,000	$15,000	Growing	30%
100,000	$30,000	Meaningful	60%
167,000	$50,000	STAGE 2 ✓	100%

6.2 Why Linear Growth Is Better Than Reflexive Growth

A reflexive design — where ULAL™ funding scales with token price — creates dangerous feedback loops that amplify both rises and crashes. AVB™'s fixed $0.30 Layer 1 contribution decouples pool depth from price volatility. The pool grows with adoption — not speculation. This is stabilizing in both directions.

6.3 Supplemental Pool Funding

Verification fees provide an additional, usage-proportional source of ULAL™ pool funding. Since verifications occur more frequently than registrations, this creates ongoing pool replenishment tied to actual system usage:

Annual Verifications	At 20% to ULAL™ ($0.02 each)	Additional Pool Funding
100,000	$2,000	Supplements registration deposits
1,000,000	$20,000	Significant depth addition
10,000,000	$200,000	Major liquidity source

6.4 Impermanent Loss Analysis

The ULAL™ pool faces impermanent loss as the token appreciates relative to ETH. However, the continuous inflow from registrations provides ongoing replenishment that partially offsets IL drag. At 167,000 registrations (Stage 2 threshold), estimated IL at 10x price appreciation is approximately 25%, yielding a net pool value of approximately $37,500 — still sufficient for Stage 2 trigger on actual dollar value.

SECTION 7 — FIRST IMPLEMENTATION: AI AGENT MARK™

7.1 The Standard

AI Agent Mark™ — the open cryptographic identity standard for personal AI agents — was submitted to the National Institute of Standards and Technology (NIST) on March 19, 2026, as a formal public comment on AI Agent Identity and Authorization. The standard is free. Open. Permanent.

7.2 The First AVB™ Implementation

Each AI Agent Passport™ registration triggers the complete AVB™ pipeline:

> *User pays $2.00 minimum (at launch pricing) → Layer 1: $1.00 operating fee → $0.70 to AI Agent Mark™ LLC operations → $0.30 to AMT/ETH Uniswap pool (ULAL™) → Layer 2: $1.00 burn fee → Acquires 10 AMT at market price → Burns 10 AMT permanently → RegistrationEvent emitted on Ethereum → Cloudflare Worker detects burn event → Private EVM blockchain records passport → Daily Merkle root anchors to Ethereum*

7.3 Live Proof — Sepolia Testnet

The complete AVB™ pipeline was proven on Ethereum Sepolia testnet on March 20–22, 2026. End-to-end test result: 10/10 price points confirmed. Zero deficit at every price point.

Contract	Address	Status
AgentMark Token (AMT)	0xfcde1C592A56FAc3455F96C95b83Def783107996	✓ Live
TwoLayerRegistrar	0xE50A0E8e3E7DfE76A1d4E87fbCc17E923e1B4350	✓ Live
UtilityHold (v2)	0x045c0c8E10BBeA293F090b522CeB5E3261796F5a	✓ Live
PoolMonitor	0xF1856B526c5939e775aa3F70C956Cb0b318A39C7	✓ Live
AMT/ETH Uniswap Pool	0x50Cc758d1781E28A249A0A0e9A747e86E2AfCa30	✓ Live
MerkleAnchor	0x54F881DF69dc6AAEF6B9a3695F9Db0C2d9f39B89	✓ Live
PassportRegistry (Private)	0x5FbDB2315678afecb367f032d93F642f64180aa3	✓ Live
Private Chain ID	20260320 (March 20, 2026)	✓ Permanent
Worker Bridge	aiagentmark-bridge.bot-1-26a.workers.dev	✓ Active

7.4 The Feedback Loop

> *Standard adopted → Registrations trigger AVB™ burns → Supply decreases → Scarcity increases value → ULAL™ pool deepens from Layer 1 → Pool stability increases → More confidence → More adoption → Repeat — autonomously — forever*

SECTION 8 — TOKEN SALE STRUCTURE

8.1 Tranche 1 — Utility-Gated

Tranche 1 tokens are available exclusively to users who have consumed the implementing protocol's service. To purchase tokens in Tranche 1, buyers must first use the service. This structure ensures every ground-floor token holder is an active user — not a speculator. The builders get the ground floor. Not the investors.

Parameter	Value
Requirement	Minimum 1 service consumption (e.g., passport registration)
Minimum purchase	Defined per implementation (Utility Hold™)
Per-user cap	Defined per implementation
Functional restriction	Service use only — no swapping until Stage 2

Implementation Example — AI Agent Mark™ Tranche 1

Parameter	Value
Requirement	Minimum 1 passport registration
Tranche size	50,000,000 AMT
Price	$0.10 per AMT
Minimum purchase	50 AMT ($5.00 — Utility Hold™)
Per-user cap	50 AMT

8.2 Tranche 2+ — Reg D Rule 506(c)

Subsequent tranches are offered exclusively to accredited investors under Regulation D Rule 506(c), funded by Tranche 1 revenue and tied to adoption milestones. The legal investment scales with the revenue.

8.3 Full Token Distribution — AI Agent Mark™ Example

Allocation	%	Tokens	Notes
Tranche 1 — Utility-gated	5%	50,000,000	Adopters only @ $0.10
Tranche 2+ — Reg D 506(c)	40%	400,000,000	Accredited investors
LLC Reserve	30%	300,000,000	Vesting + lockup required
Development Fund	25%	250,000,000	Vesting + lockup required

SECTION 9 — PROTOCOL EXTENSIBILITY

9.1 Any Project Can Implement AVB™

ULAL™ and AVB™ are not proprietary to any single implementation. They are an open protocol that any project with real-world utility consumption can deploy. Requirements for AVB™ implementation:

- A real-world service with consumable utility
- An ERC-20 token with burn capability
- A Uniswap V2 or V3 liquidity pool
- A Chainlink price oracle
- Smart contract deployment on Ethereum or EVM-compatible chain

9.2 Implementation Economics: The Sweet Spot Formula

Section 9.1 establishes that any project with real-world utility consumption can implement AVB™. This section provides the economic framework for doing so.

The Core Principle

ULAL™ works when a business can define ***N*** — a forecastable number of sale or consumption events over a defined period. Each event is a transaction. Each transaction triggers the AVB™ pipeline. The more transactions, the more capital raised, the more tokens burned, the deeper the liquidity pool.

The underlying mechanic is always the same: **forecastable units produced and sold over a defined period.** Whether those units are software subscriptions, bushels of grain, manufactured products, or service fees — the protocol does not distinguish. It only requires that real customers pay for real utility.

The following categories are suitable for ULAL™ implementation:

Category	Transaction Event	Examples
Services	Subscription payment, certification, verification	SaaS, identity, compliance platforms
Marketplaces	Each deal processed through the platform	Listing services, matching platforms
Manufacturing	Each unit sold	Consumer products, industrial goods
Commodities	Each unit produced and sold	Agricultural, mining, energy
Digital Goods	Each license, download, registration	Software, content, credentials

What ULAL™ Does Not Support

Not every business model is compatible with ULAL™. The protocol explicitly does not support:

- **Appreciating assets requiring upfront capital:** Real estate development, land banking, or any project where the value proposition is the asset itself rather than a consumable service. A $50M building requires capital before the first tenant arrives — ULAL™ generates revenue as transactions occur, not before them.
- **Projects with no forecastable transaction volume:** If a business cannot project a realistic number of sale events over a defined period, it cannot calibrate the Sweet Spot Formula.
- **Models where the primary return is appreciation, not consumption:** If the business model is “buy this and hope it goes up,” ULAL™ rejects it architecturally.

This limitation is not a weakness — it is a feature. It is the Philosophical Mandate (Section 5) expressed as an economic constraint. *Tokenization must be tied to utility.* If the business model is “produce something, sell it, customers use it,” ULAL™ powers it. If the business model is “buy this thing and hope others will pay more for it,” ULAL™ rejects it — by design, at the protocol level.

The Five Variables

Every ULAL™ implementation begins with five variables, solved in order:

Variable	Definition	How to Determine
R	Target operational raise	Business plan: how much capital is needed?
V	Service or product value per transaction	What does the customer pay per unit or transaction?
N	Forecastable transaction volume	Realistic total units sold over the raise period
L1	Layer 1 fee (the anchor)	Solve: L1 = R ÷ (0.70 × N)
A	Affordability gate	Total ULAL™ cost ÷ product value: under ~5%

Once L1 is determined, the token parameters are calibrated:

- **Total Supply (S):** Set so that total burns (B × N) consume 30–50% of supply at full adoption.
- **Burn per Transaction (B):** Token-denominated. Determines deflationary pressure per adoption event.
- **Layer 2 Cap:** Dollar cap on the burn fee, keeping total maximum customer cost sensible relative to V.
- **Launch Price:** Sets the initial Layer 2 dollar cost. At launch, L2 cost = B × launch price.

The golden rule: the total ULAL™ cost should feel like a transaction fee, not a second purchase.

Worked Example 1: Technology Startup — $5M Raise (Services)

A SaaS platform offering a $50/month subscription targets 200,000 subscribers over four years. Industry data from Carta's 2025 Founder Ownership Report, based on 45,000+ startups, establishes the median dilution trajectory under traditional equity funding:

Stage	Capital Raised	Founder Ownership
Founding	$0	100%
Seed	$3M	~56%
Series A	$15M	~36%
Series B	$40M	~23%
IPO	**$150M+**	**8–10%**

The founder builds the vision. By IPO, investors own 90% of it. Applying the ULAL™ Sweet Spot Formula to the same business:

Parameter	Value
Layer 1 Fee	$3.00/month per subscriber
Operations revenue (L1 × 70%)	**$5.04M**
Pool depth (L1 × 30%)	$2.16M
Customer cost increase	$3.00 on a $50 service = 6%
Founder ownership	**100%**

No seed round. No Series A. No board seats. The product funds itself from subscriber one.

Worked Example 2: Soybean Production — $150K Raise (Commodities)

A 1,500-acre Midwest soybean operation — roughly the size of a typical commercial grain farm in central Illinois — demonstrates ULAL™'s applicability to commodity production.

Soybean farmers are currently facing a profitability crisis. The American Soybean Association projects an $89 per acre market loss for 2025, marking a third consecutive year of losses. Production costs run approximately $660 per acre while revenue per acre is approximately $600 at current prices (~$10/bushel, ~60 bushels/acre). Farmers need capital for precision agriculture technology, equipment maintenance, and operating reserves — but traditional agricultural lending requires collateral and carries interest costs that compound the margin squeeze. *ULAL™ offers an alternative: fund operations through the sale of the commodity itself.*

Parameter	Value
Farm size	1,500 acres
Yield	55 bushels/acre = 82,500 bu/season
Market price	~$10.00/bushel
Raise period	5 growing seasons
Total transactions (N)	412,500 bushels sold

Target raise (R)	$150,000
Layer 1 Fee (L1 = R ÷ 0.70 × N)	$0.52 → rounded to $0.50/bushel
Operations revenue (L1 × 70%)	**$144,375**
Pool depth (L1 × 30%)	$61,875
Affordability: L1 as % of commodity price	$0.50 / $10.00 = 5%
Farmer ownership	**100%**

Fifty cents per bushel. Five percent of the commodity price. Over five growing seasons, the farmer raises $144,375 in operational capital and builds a $61,875 liquidity pool — with no bank loan, no interest payments, no collateral requirements, and no equity surrendered. The buyer at the grain elevator pays $10.50 per bushel instead of $10.00. That differential is smaller than the daily price fluctuation on the Chicago Board of Trade.

Worked Example 3: Collectible Toy Launch — $500K Raise (Manufacturing)

The collectible toy market illustrates ULAL™ at consumer product scale. The U.S. toy industry generated over $28 billion in annual sales in 2024, with collectible figures among the fastest-growing segments. Blind box collectibles — sealed packages containing a random figure — have become a global phenomenon, with companies like POP MART generating $1.8 billion in 2024 revenue driven by a single character line that retails at $15–$28 per unit.

Consider an independent toy company launching a new collectible figure line:

Parameter	Value
Product	Collectible blind box figures
Average retail price	$25.00 per unit
Target sales over 3 years (N)	2,000,000 units
Target raise (R)	$500,000
Layer 1 Fee (L1)	$0.35 per unit
Operations revenue (L1 × 70%)	**$490,000**
Pool depth (L1 × 30%)	$210,000
Affordability: L1 as % of retail price	$0.35 / $25.00 = 1.4%
Total cost to buyer at launch	$25.45
Founder ownership	**100%**

Thirty-five cents per toy. The customer pays $25.45 instead of $25.00 — a difference no buyer would notice at checkout. The toy company raises half a million dollars in operational capital through customer adoption alone, builds a $210,000 liquidity pool, and retains 100% ownership. And because the product is *consumed* — children play with the figures, collectors display them, they are gifts — the model satisfies the Philosophical Mandate. This directly parallels the Cabbage Patch precedent detailed in Section 4.6: products manufactured and sold for

consumptive purposes are not reclassified as securities when secondary market prices rise due to demand.

Worked Example 4: Real Estate Services Marketplace — $5M Raise (Marketplaces)

A real estate listing and services platform — similar in concept to companies like Zillow or Redfin — demonstrates ULAL™ at marketplace scale. These platforms generate revenue through listing fees, premium placements, agent subscriptions, and lead generation services. The U.S. real estate services market generates billions annually from transaction-adjacent fees without ever purchasing a property.

Note: This is a *services platform* that processes real estate transactions for others — not a real estate development company that builds or holds property. The platform provides listing verification, agent matching, and transaction documentation. The underlying properties are not tokenized. Each service fee paid by an agent or property owner is the transaction event that triggers AVB™.

Parameter	Value
Service fee per listing	$99.00
Total listings over 4 years (N)	1,500,000
Target raise (R)	$5,000,000
Layer 1 Fee (L1)	$4.75 per listing
Operations revenue (L1 × 70%)	**$4,987,500**
Pool depth (L1 × 30%)	$2,137,500
Affordability: L1 as % of service fee	$4.75 / $99.00 = 4.8%
Total fee to customer at launch	~$104
Founder ownership	**100%**

Less than five dollars per listing on a $99 service fee. The platform raises $5 million in operational capital, builds a $2.1 million liquidity pool, and the founder retains 100% ownership. No venture capital. No Series A. No investors on the board demanding an advertising-driven monetization model that turns users into products.

The Pattern

Across all four examples — from a $50/month SaaS subscription to a $10 bushel of soybeans to a $25 toy to a $99 listing fee — four properties remain constant:

Property	Result
Deficit at every price point	ZERO — guaranteed by AVB™ (Section 3.4)
Burn target	30–50% of supply — meaningful deflation without exhaustion
ULAL™ cost as % of product value	1.4% – 6% at launch — invisible to the customer
Equity surrendered by the builder	**ZERO**

The only variables that change between implementations are the parameter values. The protocol is the same. The mathematical proof is the same. The result scales from a soybean farm to a technology platform — and beyond.

> *ULAL™ does not eliminate the need for capital at scale. Tranche 2 (Section 8.2) exists specifically to bring accredited investors into the ecosystem once adoption has been demonstrated with real revenue. But ULAL™ inverts the sequence: adoption comes first, investment follows. The entrepreneur proves demand before offering equity — if they choose to offer equity at all.*

SECTION 10 — SECURITY ANALYSIS

10.1 MEV Sandwich Attack Mitigation

The ULAL™ swap in Layer 2 is vulnerable to MEV front-running. Mitigation: slippage tolerance set to maximum 2%; Flashbots Protect integration for private transaction routing; minimum viable swap size limits attack profitability.

10.2 Oracle Manipulation

Price oracle data older than 15 minutes is rejected by the TwoLayerRegistrar contract. Chainlink's decentralized oracle network provides tamper-resistant price feeds with multiple independent data sources.

10.3 Pool Manipulation

Early-stage thin liquidity creates manipulation risk. Mitigation: founder seed liquidity deposit at launch; Utility Hold™ minimum purchase accelerates pool depth; Stage 2 unlock only after verified pool depth threshold.

10.4 Smart Contract Risk

All contracts require independent security audit before mainnet deployment. A bug bounty program is recommended prior to public launch.

SECTION 11 — LEGAL FRAMEWORK

11.1 AVB™ and the Howey Test

The ULAL™ protocol — and specifically the Three Pillars architecture — was designed with securities law considerations in mind. Under the Howey test (SEC v. W.J. Howey Co., 1946), an investment contract requires: investment of money, common enterprise, expectation of profits, and profits derived from efforts of others.

Howey Prong	Exposure	ULAL™ Response
Investment of money	Always met	Accepted — not where the analysis turns
Common enterprise	Arguable	Accepted — difficult to avoid in any token model
Expectation of profits	ADDRESSED BY DESIGN	Token burned immediately — no holding, no profit opportunity. Utility Hold™ restricted to service use.
Efforts of others	Near irrelevant	Value tied to adoption demand, not management. Revenue from fixed fee, decoupled from token price.

11.2 The Cabbage Patch Precedent

Value appreciation experienced by Utility Hold™ participants between Tranche 1 and Stage 2 is coincidental to — not designed for — the primary purpose of the token. As detailed in Section 4.6, the Cabbage Patch Kids precedent (1983) establishes that products manufactured, marketed, and sold for consumptive purposes are not reclassified as securities when secondary market prices rise due to demand. The ULAL™ token model is stronger than this precedent because the token is destroyed upon use.

Additional Howey strengthening factors under the Three Pillars:

- **Pillar 1 (AVB™):** Company revenue from fixed $0.70 operating fee — explicitly decoupled from token price. $11.00 ceiling actively suppresses speculative upside.
- **Pillar 2 (Utility Hold™):** Per-user cap sized to practical usage. Tranche 1 restricted to service consumption only. Tokens held in contract, restricted to service consumption — no swapping, no transfer, no secondary market.
- **Pillar 3 (Philosophical Mandate):** Utility-first architecture enforced by immutable smart contracts. No secondary market promotion. Entire user flow is consume-or-hold-for-consumption.

11.3 Public Sale Considerations

The Utility Hold™ Tranche 1 structure — requiring service consumption before token purchase — provides the strongest available utility token defense. Subsequent tranches under Reg D Rule 506(c) provide accredited investor protections. The legal investment scales with the revenue.

This document does not constitute legal advice. Projects implementing ULAL™/AVB™ should obtain independent securities counsel before conducting public token sales.

SECTION 12 — ROADMAP

Milestone	Status	Target
AVB™ concept — ULAL™ white paper	✓ Complete	March 2026
Sepolia testnet deployment	✓ Complete	March 2026
End-to-end test 10/10 green	✓ Complete	March 2026
TwoLayerRegistrar + UtilityHold + PoolMonitor	✓ Complete	March 2026
Independent security audit	Pending	Q2 2026

CONCLUSION

ULAL™ and AVB™ represent a new economic primitive — one where token value is not hoped for, not voted on, not speculated about.

It is mathematically derived from real human demand for real services.

Built on three pillars:

- **AVB™ —** the mathematical engine that proves every burn at every price point
- **The Utility Hold™ —** the onboarding mechanism that ensures every participant is a user first
- **The Philosophical Mandate —** the architectural conviction that tokenization must serve utility, enforced by immutable code

The proof is on the chain.

Chain ID 20260320. March 20, 2026.

> *Value derived from real adoption demand. Not speculation.*
>
> **Every payment feeds the pool. Forever.**

ABOUT THE AUTHOR

Robert H. Rose is a self-taught entrepreneur and developer based in St. Louis, Missouri. He is the creator of the ULAL™ protocol and founder of AI Agent Mark™ LLC.

In March 2026, he submitted the AIAgentMark™ open standard for AI agent identity to the National Institute of Standards and Technology (NIST) as a free, public standard — and built ULAL™ as the protocol to transform adoption of that standard into mathematically provable, autonomous value.

He designed and deployed the complete AVB™ smart contract stack on Ethereum Sepolia, proving zero-deficit burns at 10/10 price points. He holds a B.S. in Business Management from Bradley University.

Rob built SaniTrace — the first blockchain-based food safety tracking system — including custom hardware, private blockchain connectivity, and consumer SMS recall alerts.

Contact: rrose@aiagentmark.com