



— WHITEPAPER · V2.0

ArbiForge

Zero-capital token launches on Arbitrum One — the whole supply in a Uniswap V3 pool locked forever, creators earning 80% of every trade's fee, and pairs that include tokenized gold and stocks.

October 2026 · Arbitrum One (chain 42161) · Open source (MIT)

1. Abstract
2. The problem
3. The ArbiForge solution
4. Why Arbitrum One
5. Real-world-asset pairs
6. Architecture
7. The launch transaction
8. Creator fee economics
9. Routing engine
10. Security model
11. Business model
12. \$FORGE token (proposed)

- 13. Governance
- 14. Roadmap
- 15. Risks
- 16. Legal notice

1. Abstract

Launching a token people can trust still asks creators for three things most of them don't have: capital to seed a liquidity pool, the discipline to lock it, and a reason to keep building after day one. Buyers, meanwhile, must audit every new contract for taxes, blacklists, hidden mints and withdrawable liquidity.

ArbiForge turns a launch into **one transaction on Arbitrum One that needs no liquidity capital**. A fixed-supply, ownerless ERC-20 is deployed; its Uniswap V3 pool is created at the 1% fee tier against ETH, a stablecoin, an ecosystem token or a verified real-world asset; and the *entire* supply is placed on one side of the pool as a single position minted straight into the **ForgeLocker** — a contract with no function to remove it. Buyers' ETH becomes the liquidity as the price rises along the position's curve. Every trade pays the pool's 1% fee, of which **80% goes to the creator** for as long as the token trades. There is no bonding-curve contract and no migration: the pool a token launches in is the pool it trades in forever, visible to every wallet, aggregator and tracker from the first block.

2. The problem

2.1 Liquidity capital is a barrier

Classic launchpads ask the creator to deposit ETH next to the tokens. Most creators can't or won't put meaningful capital at risk, so pools start tiny and easy to manipulate — or the creator keeps the LP tokens and can pull them later.

2.2 Trust is broken

Token generators routinely ship optional taxes, blacklists, pausable transfers, proxy upgrades and owner mints. Buyers cannot tell an honest token from a honeypot without reading bytecode, and “liquidity locked” claims often point to third-party lockers with admin keys or expiry dates.

2.3 Creators have no aligned income

Where creators do earn, it is usually through a token tax — the same mechanism scammers abuse, flagged by every scanner and broken by many routers. Without a clean, ongoing income, creators stop building once the launch is over.

2.4 Bonding curves add a migration step

Curve-then-migrate launchpads keep new tokens inside a proprietary contract until a threshold, invisible to the wider DeFi stack, and the migration itself is a moment of risk (sniping, failed migrations, liquidity moves).

2.5 New tokens ignore the RWA wave

Tokenized treasuries, gold and stocks are live on Arbitrum, yet nearly every new token is priced in ETH only, and creators have no way to know which RWA tokens can technically sit in a permissionless pool.

3. The ArbiForge solution

Component	What it does
Launch wizard	Three steps: token, market (pair, starting market cap, optional first buy), review. One transaction (plus one approval for ERC-20 first buys).
ForgeFactory	Deploys the token, creates and initializes its Uniswap V3 1% pool, mints the single-sided position into the locker, runs the first buy, starts the anti-sniper window and records the launch — atomically.
LaunchToken	1,000,000,000 fixed supply, 18 decimals, no owner, no tax, no mint, no blacklist, no pause, no proxy. A self-expiring 30-second, 1%-per-wallet anti-sniper window.
ForgeLocker	Holds every launch position forever; collects the pool fees and splits them between creator (80% by default) and platform. Anyone can trigger a payout.
ForgeRouter	Ownerless, fee-less multi-leg router across Uniswap V3 and V2-style DEXes, with on-chain quoting — buyers pay ETH even when the pair is gold or a stock.
Pair universe	Curated, on-chain-verified pair assets: blue chips, stablecoins, RWAs, 1,000+ Reality tokenized stocks and Arbitrum ecosystem tokens.
On-chain identity	Logo (compressed in the browser), description, links and category stored in the factory — no metadata server.
Token pages & analytics	On-chain candle charts, trades, holders, personal position & P&L, creator-fee claiming, watchlist, explorer with heat meters, platform analytics and a live trade feed — no indexer, no backend.
Operator console	Deploy and administer the platform from a browser wallet — no private keys in files.

4. Why Arbitrum One

- **Ethereum security, L2 costs.** A full launch costs cents in gas instead of tens of dollars on mainnet.
- **Canonical Uniswap V3.** The same audited Uniswap V3 factory, position manager and routers as on Ethereum, used by every wallet and aggregator on Arbitrum.
- **An RWA hub.** Tokenized gold (XAUt0), yield-bearing dollars from Sky, Maple, USD.AI, Theo and Ondo, institutional funds and Reality's tokenized stocks are live on Arbitrum One — verified on-chain for this paper.
- **No public mempool.** The first-come-first-served sequencer makes front-running a launch impractical, and the launch is atomic anyway.

5. Real-world-asset pairs

A pool's quote asset defines what a token is priced in, what sellers receive and — with ArbiForge — what most creator fees are paid in. ArbiForge offers RWA quote assets that pass three tests: (1) the contract is the issuer's canonical deployment on Arbitrum One, (2) a transfer from a real holder to a brand-new address succeeds, so a permissionless pool can hold it, and (3) the asset has a measurable DEX route so buyers paying ETH can reach it.

Asset	Issuer	Backing	Pool-ready
XAUt0	Tether	Physical gold (1 token = 1 troy oz)	Yes
sUSDS	Sky	USDS earning the Sky Savings Rate	Yes
syrupUSDC	Maple	Over-collateralized institutional loans	Yes
USDai / sUSDai	USD.AI	Treasuries / GPU-backed loans	Yes
thBILL	Theo	Short-duration U.S. T-bills	Yes
USDY	Ondo	Short-term treasuries & deposits	Yes (thin liquidity)
BUIDL · BENJI · USTBL · EUTBL	BlackRock · Franklin · Spiko	Tokenized money-market funds	No — KYC whitelist
rTokens (rAAPL, rNVDA, rSPY, rGLD, rSGOV ... 1,000+)	Reality	U.S. stocks & ETFs held 1:1 in custody	Yes — ERC-8056, freely transferable (most liquidity on Bitget)

Pairing with XAUt0 denominates the token in gold and pays its creator in gold; pairing with an accruing asset (sUSDS, syrupUSDC, thBILL) means both the pool's quote side and the creator's fees keep their real-world yield.

6. Architecture

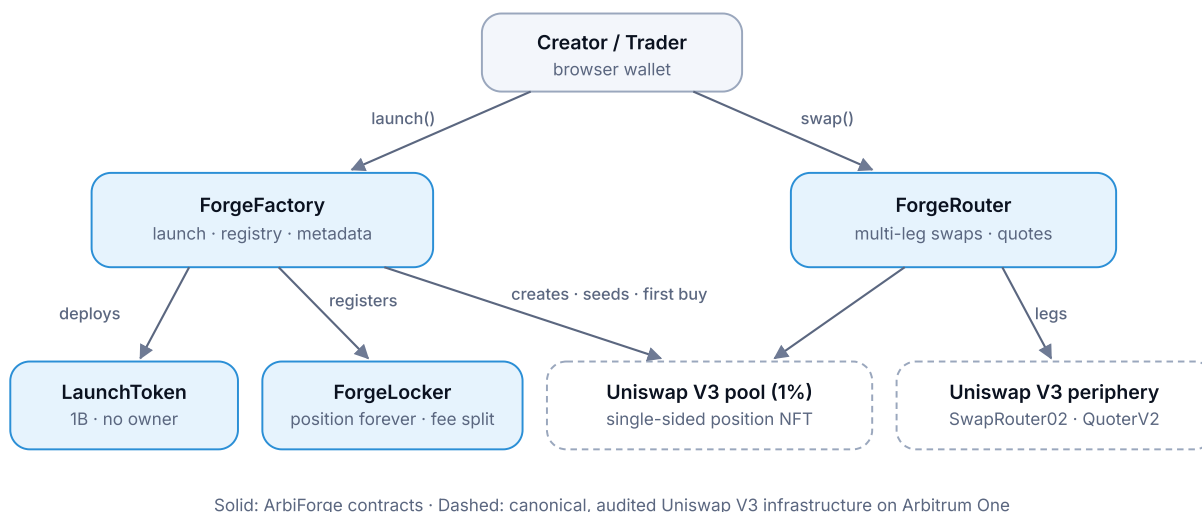


Figure 1 — Contracts and their relationships. The front-end is a static site reading the chain through public RPCs and Multicall3; there is no backend.

6.1 LaunchToken

OpenZeppelin `ERC20` + `ERC20Burnable` + `ERC20Permit`, with no owner. The constructor mints exactly 1,000,000,000 tokens to the factory. The only extra logic is the anti-sniper window: once the factory calls `activate` (once, at the end of the launch), for 30 seconds no non-exempt wallet may hold more than 1% of the supply. It cannot be extended or restarted.

6.2 ForgeFactory

Validates parameters, books the flat launch fee, deploys the token, creates the Uniswap V3 pool at the 1% tier (tick spacing 200), initializes it at the creator's start price, mints the single-sided position straight to the locker, burns placement dust, executes the optional first buy through SwapRouter02, starts the anti-sniper window and stores a `Launch` record (token, creator, pool, quote, position id, timestamp, creator share, first-buy tokens) plus creator-editable JSON metadata. Admin powers: launch fee (≤ 0.05 ETH), fee recipient, platform share for future launches ($\leq 30\%$), pausing *new* launches — under two-step ownership.

6.3 ForgeLocker

Deployed by the factory in its constructor. Owns every launch position NFT. It has no function to decrease liquidity, transfer a position or change a split. `collectFees(token)` — callable by anyone — collects the position's fees, pays the creator share to the token's creator recipient and the rest to the platform fee recipient, and records cumulative earnings.

`setCreatorRecipient` lets only the current recipient redirect its own stream.

6.4 ForgeRouter

Executes up to four legs, each either a Uniswap V3 path through SwapRouter02 or a path on a registered V2 router (Camelot's referrer signature supported). It wraps/unwraps ETH, measures each leg by balance difference, checks token continuity, enforces minimum output and deadline, and exposes `quote()` for one-call simulation. It has no owner and charges no fee.

7. The launch transaction

1. Check launches are not paused, name/symbol/metadata are valid and `msg.value` = fee (+ first buy for ETH pairs).
2. Deploy `LaunchToken`; the factory holds all 1B tokens.
3. Create the token/quote pool at the 1% tier. The token did not exist before this transaction, so the pool cannot exist either.
4. Initialize the pool at the creator's price, oriented for whichever side the token landed on (token0 or token1).
5. Mint one position holding the whole supply in the range from the first tick above the current price to the maximum (or the mirror range if the token is token1) — so it contains tokens only — directly to the `ForgeLocker`; burn the dust.
6. Register the position and fee split in the locker.
7. Optionally execute the creator's first buy; revert if it would exceed 10% of the supply.
8. Start the 30-second anti-sniper window, record the launch and emit `TokenLaunched`.

Pricing follows Uniswap V3 concentrated-liquidity math. For a start market cap M (in the quote asset) and a buy of x after the fee:

```
start price          = M / 1,000,000,000
share of supply out ≈ x / (M + x)           // like a constant-product curve wit
price after buy      = start price × ((M + x) / M)2
```

A 1.5 ETH start market cap means 0.15 ETH buys ~9% of the supply and lifts the price ~21%; the 10x market-cap mark is reached after roughly $2.2 \times M$ of net buying ($\sqrt{10} - 1$). Because the position spans to the maximum price, the pool never runs out of tokens.

8. Creator fee economics

Every swap through a launch pool pays Uniswap's 1% fee on its input — ETH (or the pair asset) on buys, tokens on sells. The fee accrues to the only position in the pool: the locked one. On collection the locker splits it:

	Creator	Platform
Default split	80%	20%
Bounds	always $\geq 70\%$	$\leq 30\%$, future launches only
Paid in	the pair asset (buys) and the launched token (sells)	

Example: a token trading \$10,000 a day generates \$100 in fees: \$80 to its creator, \$20 to the platform — about \$2,400 a month to the creator, for as long as the market lives. The fee is enforced by the pool, so it is collected whether a trader uses ArbiForge, Uniswap's interface or any aggregator, and it never appears as a token tax.

Unlike a curve-then-migrate model, there is no graduation event to time: the pool and its fee stream exist from the first block. The *Forged* milestone (default \$50,000 market cap) is a display badge only.

9. Routing engine

For a swap $A \rightarrow B$ the front-end builds a token set: A, B, the hubs WETH, USDC, USDT0 and ARB, plus the pair asset of any ArbiForge-launched token involved. Two Multicall3 calls discover live Uniswap V3 pools (four fee tiers) and V2 pairs (Uniswap, SushiSwap, Camelot); fresh launch pools are kept even before their first trade, when their active liquidity is still zero. Candidate routes — direct, one hop, and mixed venues — are quoted in one call through `ForgeRouter.quote`, in parallel with the KyberSwap aggregator; the higher output wins. No indexer, no backend, and a token is tradable from the block it is created in.

10. Security model

Threat	Mitigation
Liquidity rug	The whole supply sits in one position minted to the ForgeLocker, which has no withdraw, decrease or transfer function.
Honeypot / sell block	No owner, blacklist, pause or transfer-restriction code; the anti-sniper window only limits receivers, exempts the pool and expires after 30 seconds.
Hidden taxes	The transfer hook never changes amounts; the only trading cost is the pool's 1% fee, set by Uniswap and immutable.
Infinite mint / dilution	No mint function; supply is fixed at 1B and can only decrease (burns).
Launch sniping	Atomic creation; the creator's first buy executes before any other order; 1%-per-wallet cap for 30 seconds.
Hidden insider allocation	The creator receives no tokens except an optional, capped (10%) and publicly recorded first buy.
Fee theft	Payouts only go to the stored creator recipient and the platform recipient; only the creator recipient can redirect its stream; splits are immutable per token.
Malicious quote token / reentrancy	<code>ReentrancyGuard</code> on launch, fee collection and swaps; <code>SafeERC20</code> ; balance-difference accounting; approvals reset after use.
Platform admin abuse	Admin cannot touch launched tokens, positions, splits or swaps; fees capped; two-step ownership; multisig recommended.
Front-end compromise	Open-source static site with a strict Content-Security-Policy, mirrorable on IPFS; contract guarantees hold regardless of UI.

Assurance: unit tests against the official Uniswap V3 bytecode (100% line coverage of the launch contracts, both pool orientations), integration tests on a fork of Arbitrum One with ETH-, ARB- and XAUt0-paired launches, and Slither static analysis with no high- or medium-severity findings. An independent audit is planned before significant volume.

11. Business model

- **Launch fee:** free at launch (0 ETH). The operator can introduce a flat ETH fee per launch later, capped on-chain at 0.05 ETH.
- **Trading-fee share:** 20% of every launch pool's 1% fee by default ($\leq 30\%$, fixed per token at launch). Platform revenue grows with the success of the tokens it hosts — the same incentive as its creators.
- **No swap fee on ForgeRouter, no token tax, no supply allocation.** The platform never takes tokens from a creator.

Revenue funds development, audits, infrastructure and ecosystem grants.

12. \$FORGE token (proposed)

 **Draft.** \$FORGE does not exist yet and the platform works fully without it. Parameters below are a proposal for community discussion and may change; nothing here is an offer to sell tokens.

Utility

- **Governance** over the launch fee, the platform share for future launches, pair curation and the grants budget.
- **Revenue alignment:** a share of platform trading fees directed to stakers or buy-back-and-burn, by governance vote.
- **Featured placement** in the explorer for community-curated launches.
- **Access** to advanced launch types (custom-tax pools, vesting, airdrops) as they ship.

Supply & allocation

Fixed supply of **1,000,000,000 FORGE**, launched through ArbiForge itself (zero-capital, liquidity locked forever) to demonstrate the product.



- Community & ecosystem incentives — 35%
- DAO treasury — 15%
- Early contributors — 10% (6-month cliff, 24-month vesting)
- Launch pool (locked forever) — 15%
- Core team — 15% (12-month cliff, 36-month vesting)
- Fair launch / public distribution — 10%

13. Governance

Progressive decentralization: the platform starts under a multisig owner (two-step ownership transfer), publishes every parameter change on-chain, and moves control of the factory's limited admin functions to \$FORGE governance once the token and community exist. Ownerless components — the locker, the router and launched tokens — never need governance.

14. Roadmap

Phase	Milestones
1 • Forge (live)	Zero-capital Uniswap V3 launches, liquidity locked forever, 80% creator fees, first buy and anti-sniper window, RWA & stock pairs, ForgeRouter, on-chain charts, holders, positions, watchlist, analytics, operator console.
2 • Amplify	Independent audit; custom-tax pools on Uniswap v4 hooks; community takeover tooling for abandoned fee streams; token comments; airdrop tools; IPFS mirror.
3 • \$FORGE	Token launch, governance, revenue alignment, community pair curation, builder grants.
4 • Orbit	Arbitrum Orbit chains, Stylus-based templates, cross-chain launches, RWA issuer partnerships.

15. Risks

- **Smart-contract risk.** Despite tests and a minimal design, bugs are possible; the code has not yet been independently audited.
- **Market risk.** Launched tokens can lose all value. With no seeded quote liquidity, early trades move the price sharply in both directions; sellers can only receive what buyers paid in.
- **Pair-asset risk.** Creator fees and pool value follow the pair asset's price; RWA tokens depend on off-chain custodians and legal structures, and issuers can freeze addresses.
- **Dependency risk.** Uniswap contracts, the KyberSwap API, public RPCs and price APIs are third-party infrastructure.
- **Regulatory risk.** Rules on token issuance and trading vary by jurisdiction and are evolving.

16. Legal notice

This document is for information only and describes open-source, non-custodial software. It is not an offer or solicitation to buy or sell any token or security, nor financial, legal or tax advice. Tokens created with ArbiForge are issued by independent users who are solely responsible for them. Statements about future plans, including the proposed \$FORGE token, are forward-looking and may change or not happen. ArbiForge is not affiliated with Offchain Labs, the Arbitrum Foundation, Uniswap Labs, Camelot, SushiSwap, KyberSwap or any RWA issuer mentioned. Asset data was verified on-chain in October 2026 and may change.

Contract addresses for this deployment: see the Documentation page.