FLARESIGNAL

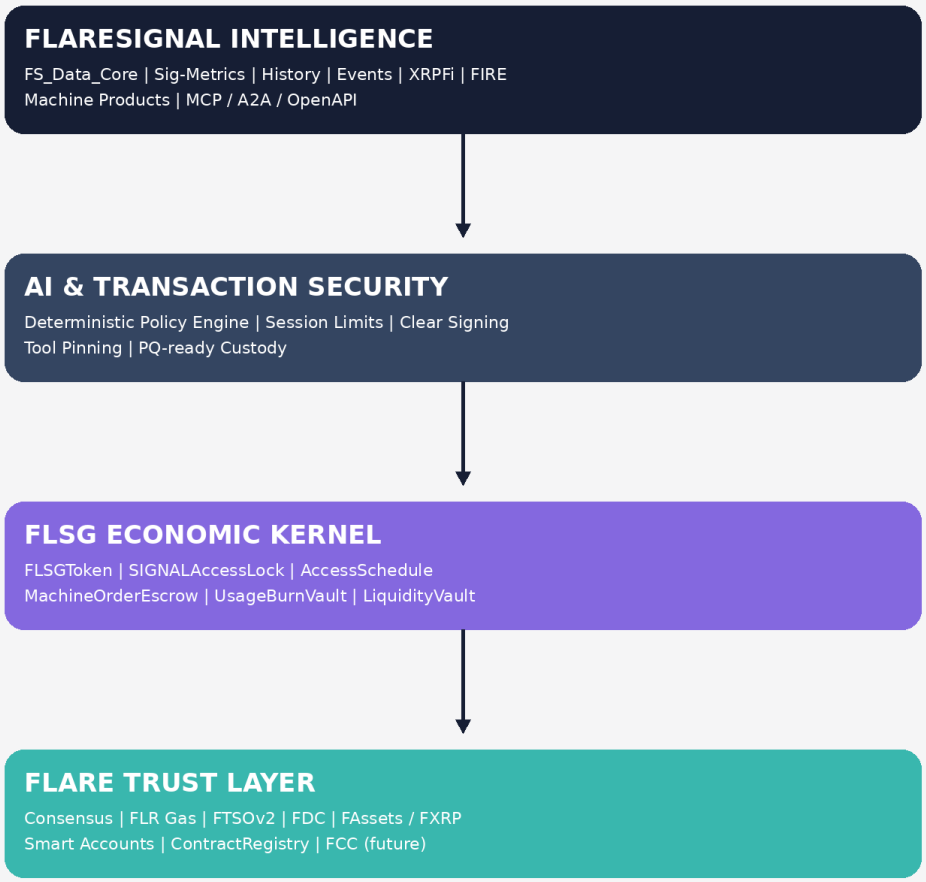
FLSG

Proposed Token, Machine-Access Economy, Security Architecture & Tokenomics

WHITEPAPER / DESIGN DRAFT v0.8

1 November 2026

Core thesis
FlareSignal (FLSG) is the proposed fixed-supply access asset for FlareSignal machine intelligence. Approved assets pay for intelligence consumption; FLSG determines the economic and operational tier at which machines can consume FlareSignal.



Architecture principle: Flare supplies the shared trust layer; FLSG adds only the protocol-specific access and economic kernel.

DESIGN PROPOSAL - NOT A TOKEN OFFERING - NOT A PRICE FORECAST

READ FIRST

Document Status & Disclosure

This document consolidates the current proposed architecture for FlareSignal (FLSG). FLSG has not yet been deployed. Contract parameters, launch mechanics, legal treatment and some economic settings remain subject to technical validation, external security review and legal/regulatory advice before any Mainnet implementation.

Security disclosure policy
The whitepaper intentionally publishes security architecture, contract permissions, immutability rules, custody boundaries, invariants and testing methodology. Security does not depend on hiding these controls. Operational secrets remain private: private keys, recovery credentials, signer/device locations, infrastructure topology, API credentials, exact live alert thresholds and internal incident-response procedures.

What this document is

- A technical and economic design specification for the proposed FLSG token and SIGNAL machine-access system.
- A transparent statement of intended monetary rules, custody boundaries, security properties and deployment gates.
- A summary of the ten-year tokenomics model currently maintained separately in the design workbook.

What this document is not

- An offer to sell FLSG or a solicitation to invest.
- A promise of returns, token appreciation, adoption or liquidity.
- A claim that any system can be made unhackable or quantum-proof.
- A substitute for independent security audit, formal verification or legal advice.

Current protocol programme state

PROGRAM LAYER	CURRENT STATE
Canonical product baseline	Production convergence active
Machine commerce substrate	PayChain + Base Sepolia x402 proof complete; Base mainnet production canary pending
FLSG protocol specification	Architecture defined; implementation freeze remains gated by production commerce and final parameters
Coston2 immutable kernel	Not deployed
External assurance / legal	Future evidence and funding gate

Stage	Current status
FLSG Mainnet	Not launched

Contents

1. Executive Summary

2. FlareSignal Before FLSG

3. Why FLSG Exists

4. Token Design Principles

5. SIGNAL Access Economy

6. Pricing & Access Epochs

7. Burn Architecture

8. Tokenomics & Genesis

9. Ten-Year Economic Model

10. Launch & Liquidity Design

11. Flare-Native Architecture

12. Bespoke Contract Architecture

13. Non-Custodial Funds & Property Rights

14. AI / Agent Security

15. Smart-Account & Signing Security

16. Post-Quantum Readiness

17. Testing, Formal Verification & Audit

18. Deployment & Supply-Chain Security

19. Transparency & Live Intelligence

20. Roadmap

21. Risks & Regulatory Readiness

22. Constitutional Invariants

23. Research & References

PURPOSE

1. Executive Summary

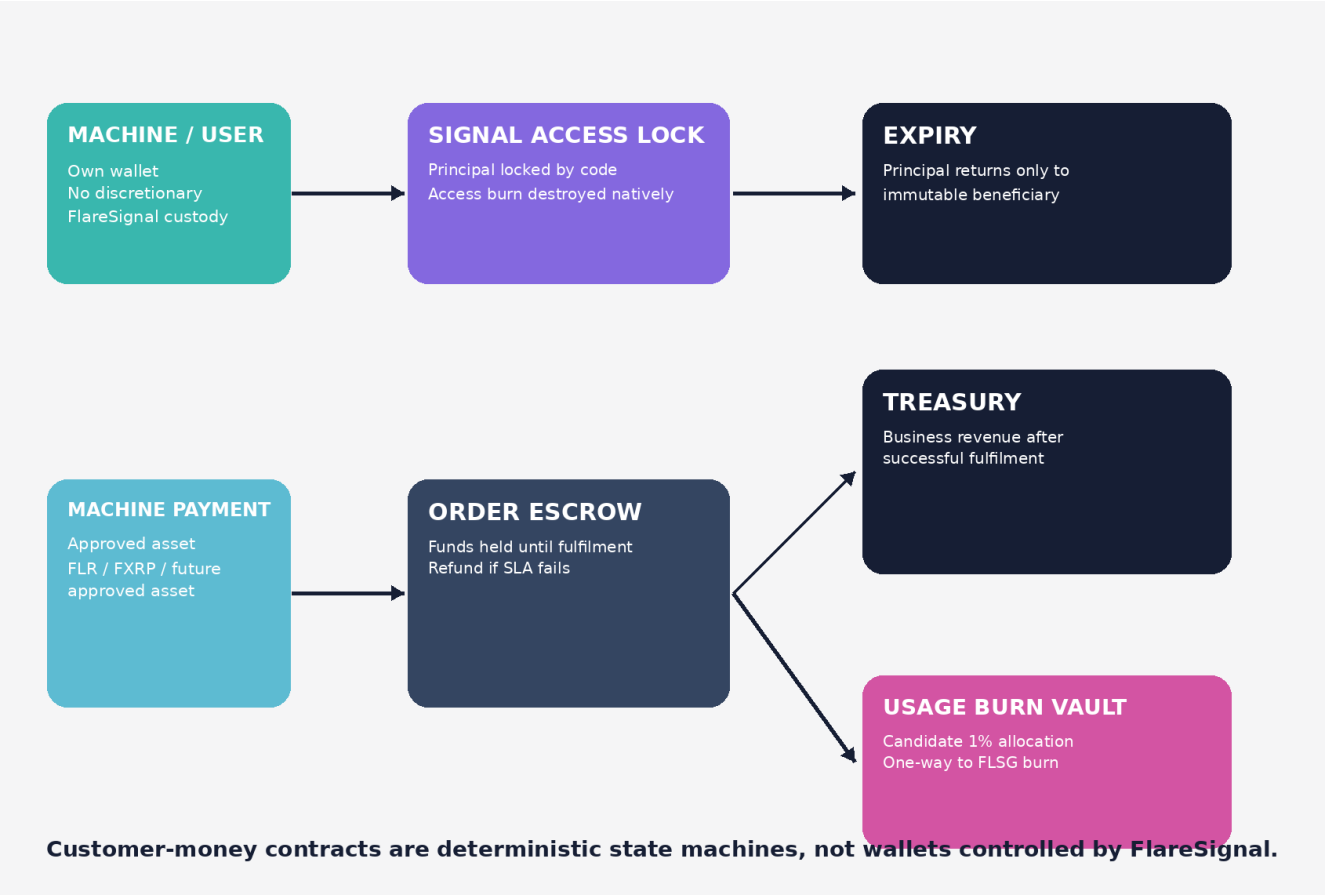
FlareSignal is an existing Flare and Songbird intelligence platform with a canonical truth architecture, live network and market intelligence, XRPFi and FIRE coverage, historical/event analysis, proprietary Sig-Metrics, and machine-readable interfaces. FLSG is proposed as a later economic layer on top of that product - not as a token launched first in search of utility.

One-line thesis
FLSG is the fixed-supply access and economic-commitment asset for FlareSignal machine intelligence. Machines pay product bills in approved settlement assets; FLSG determines enhanced access capacity and economics.

Proposed properties at a glance

Property	Current design
Name / ticker	FlareSignal / FLSG
Canonical chain	Flare Mainnet
Genesis supply	1,000,000,000 FLSG
Future minting	None
Gas	FLR
Transfer tax	None
Staking APY / emissions	None
Token role	Machine-access / capacity commitment, not general payment
SIGNAL Lock terms	1 / 3 / 12 months
Lock principal	Returned in FLSG at expiry
Access burn	Candidate: 0.5% of economic lock value per committed month
Usage burn	Candidate: 1% of qualifying paid machine-product revenue, funded from revenue/margin
Price authority	Standard FTSOv2 FLSG/USD preferred before SIGNAL Access opens
Customer custody	Non-custodial / immutable smart-contract control

Property	Current design
Critical contract upgrades	Immutable versions; new V2 rather than rewriting V1



Proposed separation between customer principal, order escrow, business revenue and usage burn.

PRODUCT FIRST

2. FlareSignal Before FLSG

The architectural premise is deliberately product-first. FLSG is only justified if it improves the economics and capacity management of a real machine-intelligence service that already exists.

Existing FlareSignal foundations

- Canonical truth boundary: FS_Data_Core, independent of underlying storage lanes.
- Public and Pro intelligence surfaces covering network, markets, XRPFi, capital flow, burn, wallets, yield, FIRE, events and utilisation.
- Proprietary Sig-Metrics families spanning execution, liquidity, policy transmission, capital character, demand persistence, accumulation/distribution, utility vs price, concentration, protocol capital flows, derivatives and composite regimes.

Existing FlareSignal machine substrate

- REST/OpenAPI, MCP and A2A machine interfaces backed by the same canonical FS_Data_Core truth boundary as human-facing surfaces.
- Six machine-product families covering current/historical network state, XRPFi capital, yield/collateral opportunity, Sig-Metric cross-reference and event before/after analysis.
- PayChain: FlareSignal-developed crypto-payment infrastructure for non-custodial checkout, deterministic payment verification and purchase/membership activation.
- Metering, entitlement, idempotency and output-verification foundations required for autonomous paid intelligence.

TECHNICAL SEQUENCING

Canonical intelligence stack -> proven PayChain + Base Sepolia commerce substrate -> Base mainnet production settlement -> implementation-grade FLSG freeze -> Coston2 immutable kernel -> adversarial/external assurance -> Mainnet and price-authority gates.

UTILITY

3. Why FLSG Exists

FLSG is not another payment token

Flare already has FLR for gas and network economics. XRP-originated value can reach Flare through FXRP/FAssets, and other approved assets can be integrated when there is a canonical path. Requiring FLSG simply to pay for data would add friction without creating unique value.

The proposed use case is therefore different: FLSG becomes an on-chain machine-access bond. Standard pay-per-request access remains possible without FLSG. Higher-volume or more demanding machine customers can acquire and lock FLSG to obtain better service economics and capacity.

Candidate machine-access benefits

- Higher packet-unit quotas / request capacity.
- Deeper retained history and heavier historical jobs.
- Higher concurrency and throughput limits.
- Access to computationally expensive Sig-Metric combinations.
- Reserved execution capacity / stronger service levels.
- Improved commercial economics for sustained machine usage.

Utility boundary

FLSG is reserved for FlareSignal-specific machine-access economics where FLR does not provide an equivalent service-specific function. FLR remains the network asset wherever it already satisfies the requirement.

MONETARY CORE

4. Token Design Principles

Immutable fixed-supply ERC-20

The monetary core is intentionally minimal. The proposed FLSG token is a standard ERC-20 with exactly one billion units created at genesis and no external mint path thereafter. Native burn reduces totalSupply. There is no transfer tax, reflection, rebase, blacklist, admin seizure or owner-controlled monetary policy.

FLR remains gas

Every FLSG transfer, lock, renewal, settlement or burn is still a Flare transaction and therefore consumes FLR gas. FLSG does not become gas and does not compete with FLR for network-level utility.

No governance over truth

FLSG ownership must never determine what a FlareSignal metric says. Token holders do not vote on canonical facts, metric values, source authority or historical truth. The data architecture remains governed by evidence and canonical source rules rather than token ownership.

Immutable versioning

Customer-money contracts use immutable versioning rather than upgradeable proxies. If AccessLockV1 later requires improvement, AccessLockV2 is deployed while existing V1 locks continue under the exact V1 code and terms until expiry.

MACHINE ACCESS

5. SIGNAL Access Economy

SIGNAL is the proposed name for the machine-access economic system; FLSG is the actual on-chain token.

Current baseline tiers

Tier	Economic lock target	Illustrative role	Principal at expiry
Standard	$0	Public / pay-per-use machine access	No FLSG lock
Enhanced	$100	Small agents / developers	Returned in FLSG
Professional	$500	Serious automated research	Returned in FLSG
Institutional	$1,500	High-volume / reserved capacity	Returned in FLSG

Lock terms

Current design terms are 1, 3 and 12 months. A lock is non-withdrawable until expiry. There is no “early exit penalty” because there is no early exit. At expiry the principal is returned in FLSG, not a guaranteed fiat value.

Economic-value based, not fixed token quantities

The number of FLSG required for a tier adjusts with the FLSG reference price. A $1,500 Institutional commitment would require 1,500,000 FLSG at $0.001, 150,000 at $0.01, 15,000 at $0.10 or 1,500 at $1.00. This keeps the service commitment economically stable while the token quantity adapts.

Renewal

Existing lock terms remain fixed for their duration. Renewal is a new decision under the then-current immutable schedule version. If fewer tokens are required, surplus principal returns; if more are required, the owner tops up.

FTSO-FIRST

6. Pricing & Access Epochs

The preferred mature price authority is a standard FTSOv2 FLSG/USD feed. FTSOv2 is an enshrined Flare protocol with block-latency feeds that update with each Flare block (approximately 1.8 seconds) and standard feeds supported by around 100 independent providers. [1]

Externalised price authority

FLSG introduces no independent oracle network. Price authority is externalised to Flare, with SIGNAL Access opening only after market liquidity and a robust reference source exist, preferably a standard FTSOv2 FLSG/USD feed.

Access Epochs

Existing machine locks remain fixed for their active term. AccessSchedule publishes the FLSG requirement for each new access epoch, likely monthly; new locks use the published epoch amount while existing locks remain unchanged.

Fail-closed pricing

New schedule epochs halt when the required price source is unavailable or fails freshness/sanity requirements. No substitute price is invented, and existing customer rights continue unaffected.

DEFLATION TIED TO UTILITY

7. Burn Architecture

No transfer tax

Buying, selling or transferring FLSG does not automatically burn tokens. Deflation is proposed only where there is genuine FlareSignal utility.

Burn Engine A - SIGNAL Access Burn

Current model candidate: 0.5% of economic lock value per committed month. That gives 0.5% for one month, 1.5% for three months and 6% for twelve months. The burn is calculated in economic value and converted into FLSG using the access schedule.

Example
Institutional 12-month lock: $1,500 equivalent principal + $90 equivalent access burn. At an FLSG reference price of $0.01 this would mean 150,000 FLSG principal and 9,000 FLSG burned. The 150,000 principal returns at expiry; the 9,000 burn is permanently destroyed.

Native destruction rather than a dead wallet

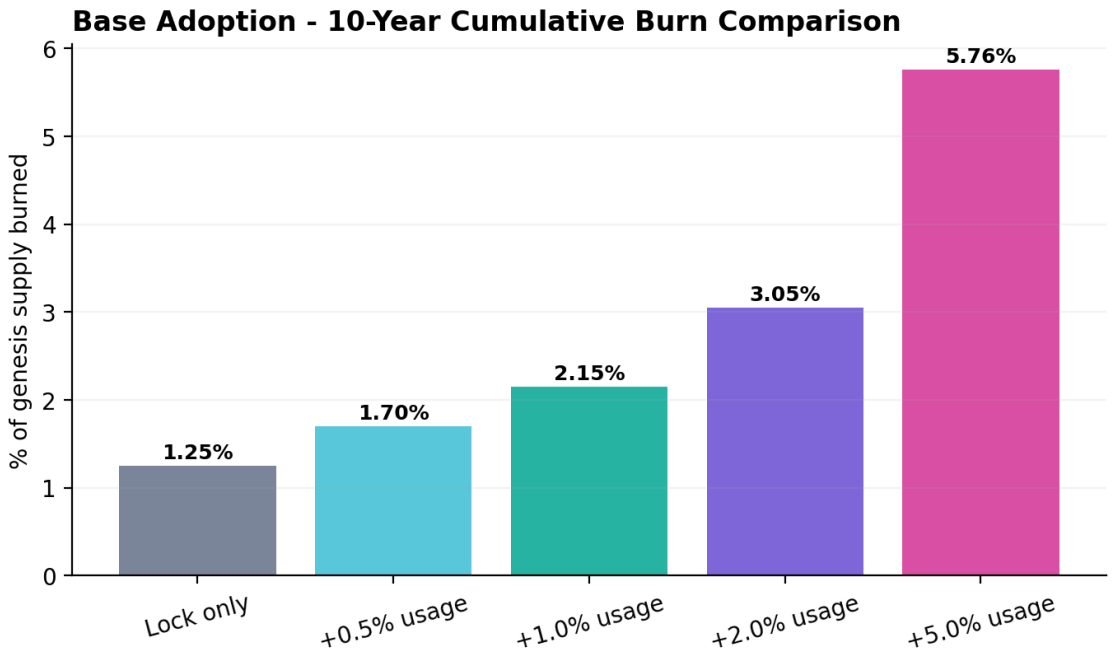
The preferred implementation is contract-level ERC-20 burn that reduces totalSupply, rather than transferring FLSG to a conventional 0x...dead address and maintaining separate circulating-supply arithmetic.

Burn Engine B - usage-derived burn

The current preferred candidate is 1% of qualifying paid FlareSignal machine-product revenue allocated to acquire and burn FLSG. It is funded from FlareSignal revenue/margin and does not increase the customer price. This mechanism remains subject to liquidity design, legal/regulatory review and final economics.

Execution protection

UsageBurnVault is designed as a one-way reserve: value committed to burn cannot return to operating treasury. FLSG acquisition uses manipulation-resistant execution, potentially a bounded batch/reverse auction, rather than a predictable large AMM market order. Final execution mechanics remain a design item.



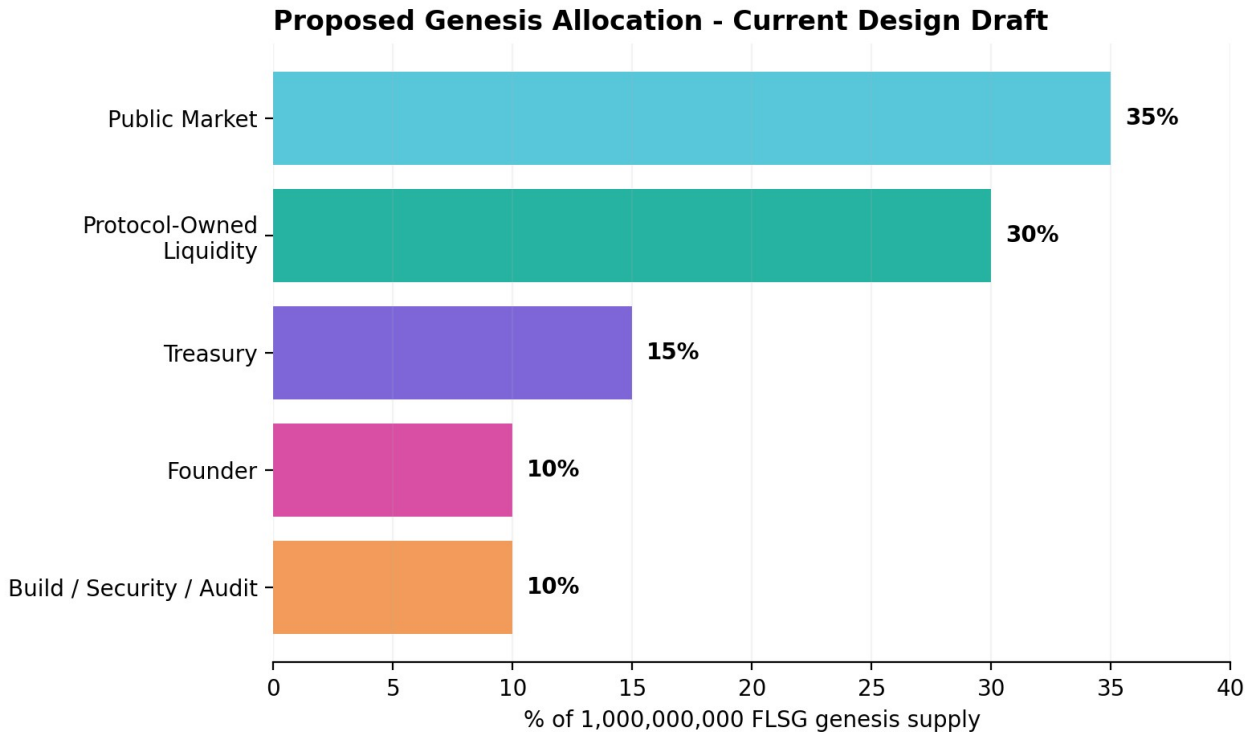
Usage burn is modeled as a share of qualifying paid machine-product revenue and does not increase the customer price.

Workbook comparison of lock-only and usage-derived burn options under Base adoption assumptions.

CURRENT DESIGN DRAFT

8. Tokenomics & Genesis

The genesis allocation below is the current proposed design, not yet an immutable launch decision. It is constructed to avoid a discounted private/VC round, avoid liquid founder supply at launch and make market liquidity a protocol asset.



Bucket	%	FLSG	Launch market-facing	Core restriction
Public Market Allocation	35%	350M	100M	No private/OTC allocation; public on-chain distribution only
Protocol-Owned Liquidity	30%	300M	100M	Liquidity purpose only; not operating treasury
FlareSignal Treasury	15%	150M	0	12m cliff; proposed 60m eligibility
Founder	10%	100M	0	12m cliff; proposed 48m linear vesting
Build / Security / Audit	10%	100M	0	Purpose-bound, timelocked security/engineering use
Total	100%	1.0B	200M	Fixed genesis; no future minting

Founder and project overhang controls

- Founder: zero liquid FLSG at launch; current proposal is 12-month cliff plus 48-month linear vesting.
- Treasury: zero liquid FLSG at launch; current proposal is 12-month cliff plus 60-month linear eligibility.
- Build/Security/Audit: purpose-bound vault, transparent outflows and no unrestricted sweep to operating treasury.
- Public allocation: no private discounts or private unlock cliff.
- Protocol-owned liquidity: liquidity fees remain inside the liquidity system rather than being treated as ordinary business revenue.

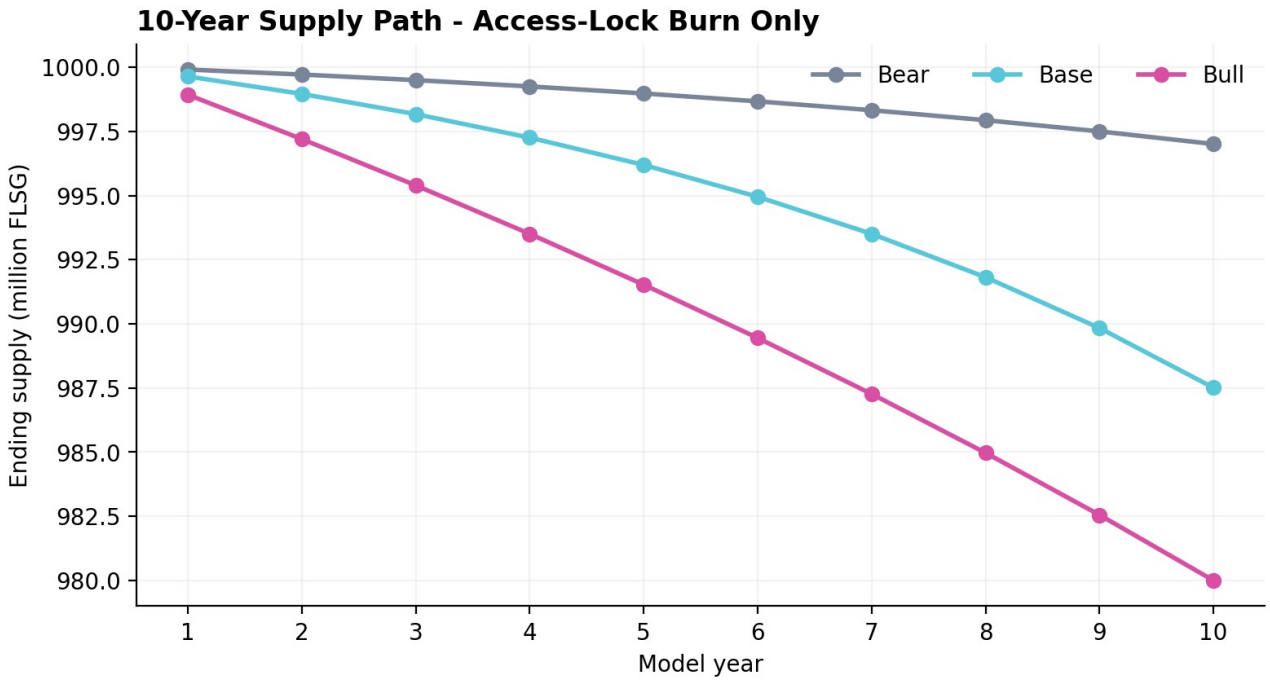
MODEL, NOT FORECAST

9. Ten-Year Economic Model

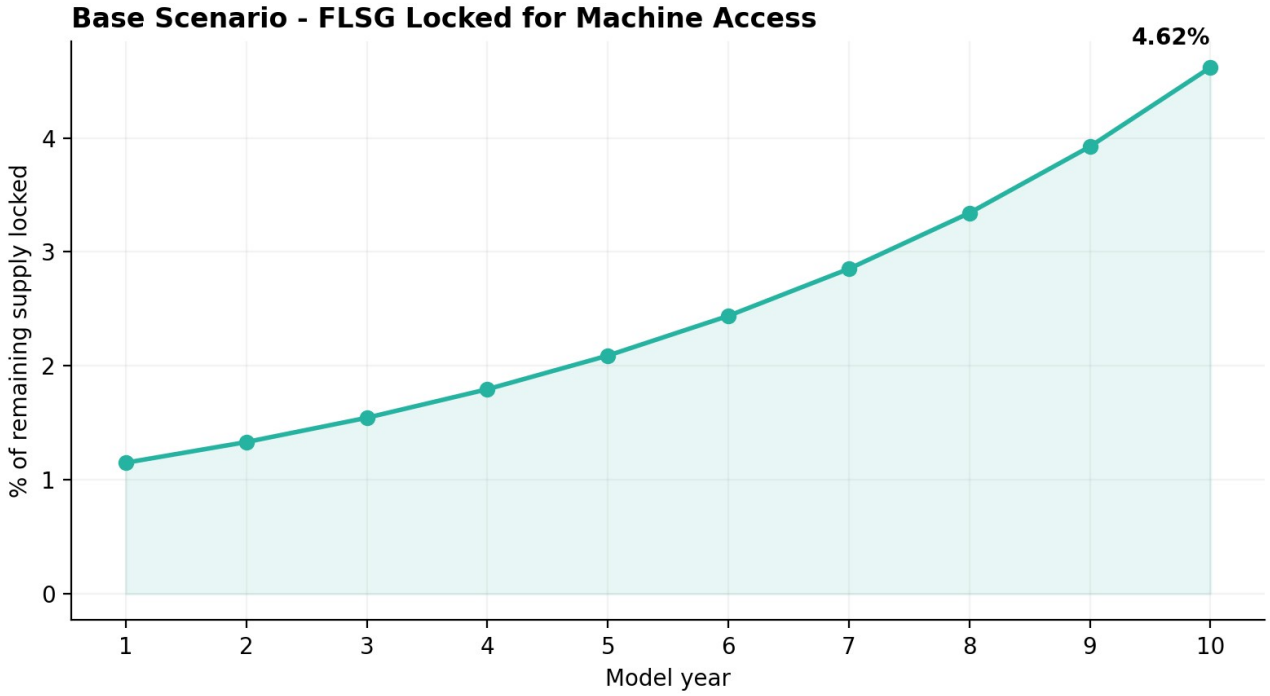
The ten-year model is a design stress tool. It tests whether the proposed lock/burn mechanics remain usable across adoption and price assumptions. It does not predict FLSG price, machine adoption or revenue.

Base scenario Year 10 - access-lock burn only

Metric	Workbook result
Active machine locks	36,424
Economic lock value	$9.42M
FLSG locked	45.61M
% remaining supply locked	4.62%
10Y cumulative access burn	12.49M FLSG
10Y cumulative access burn	1.25% of genesis
Year 10 annual access burn	2.33M FLSG



Illustrative scenario model - not a price forecast.



Bear / Base / Bull Year 10 comparison

Scenario	Active locks	Economic lock value	FLSG locked	% supply locked	10Y access burn	Ending supply
Bear	1,989	$0.47M	13.44M	1.35%	0.30%	997.01M
Base	36,424	$9.42M	45.61M	4.62%	1.25%	987.51M

Scenario	Active locks	Economic lock value	FLSG locked	% supply locked	10Y access burn	Ending supply
Bull	227,586	$62.41M	40.54M	4.14%	2.00%	979.99M

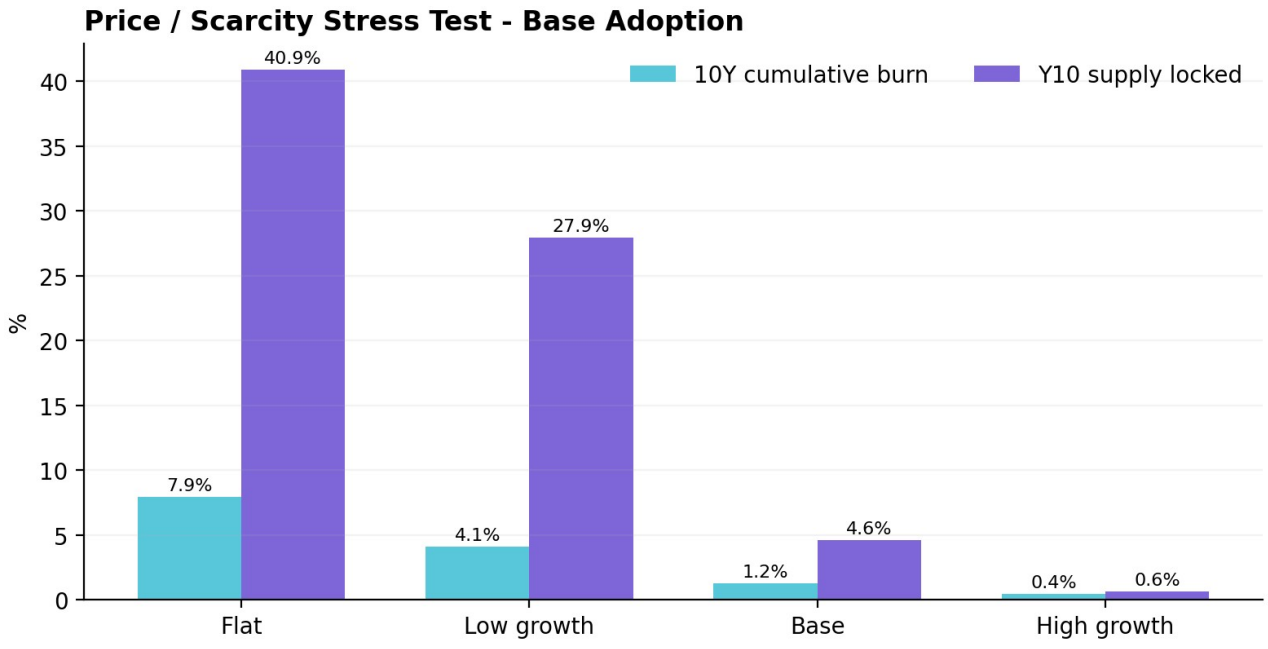
Machine lock break-even economics

Using the candidate 0.5% monthly access burn and a purely illustrative 5% annual opportunity-cost hurdle, the 12-month simplified carrying-cost test is:

Tier	Lock value	Annual burn cost	Opportunity cost	Simplified annual carrying cost	Monthly break-even benefit
Enhanced	$100	$6	$5	$11	$0.92
Professional	$500	$30	$25	$55	$4.58
Institutional	$1,500	$90	$75	$165	$13.75

The model deliberately excludes FLSG price volatility/risk. An institutional treasury would assess that separately.

Price / scarcity stress



Stress scenario - assumed token-price paths are model inputs, not forecasts.

Price-path assumption	Y10 cumulative burn	Y10 supply locked	Y10 effective reference price*
Flat	7.93%	40.90%	$0.0250
Low growth (+15% p.a.)	4.06%	27.92%	$0.0352
Base (+40% p.a.)	1.25%	4.62%	$0.2066
High growth (+75% p.a.)	0.44%	0.61%	$1.5394

*These price paths are model inputs and utility-clearing outputs, not forecasts. Their purpose is to reveal supply pressure and lock feasibility under different assumptions.

Usage-burn model

Illustrative annual paid-product spend assumptions are $120 per Enhanced machine, $1,200 per Professional machine and $12,000 per Institutional machine. Under the Base adoption path, modeled Year 10 machine revenue is approximately $32.45M. With the candidate 1% usage burn, cumulative ten-year burn increases from approximately 12.49M FLSG (1.25% genesis) to approximately 21.52M FLSG (2.15% genesis). These are design scenarios only.

CURRENT PROPOSAL

10. Launch & Liquidity Design

Public price discovery rather than a private seed round

The current technical preference, subject to legal approval, is a Flare-native uniform-price public batch / commit-reveal price-discovery process for a 100M FLSG launch tranche. The design goal is to reduce first-block bot advantage and avoid a discounted insider cost basis.

Proceeds become market infrastructure

The proposal routes 100% of FLR launch proceeds to LiquidityVault rather than Operating Treasury. LiquidityVault pairs those proceeds with 100M FLSG from the protocol-owned liquidity allocation to establish the canonical FLSG/FLR market at the clearing price.

Liquidity depth derived from actual utility

Quote-side reserve	$1,500 acquisition impact*	Approx pool TVL	Interpretation
$50k	~3.0%	$100k	Too thin for preferred launch
$75k	~2.0%	$150k	Below preferred floor
$100k	~1.5%	$200k	Hard-floor candidate

Quote-side reserve	$1,500 acquisition impact*	Approx pool TVL	Interpretation
$150k	~1.0%	$300k	Preferred minimum
$250k	~0.6%	$500k	Strong

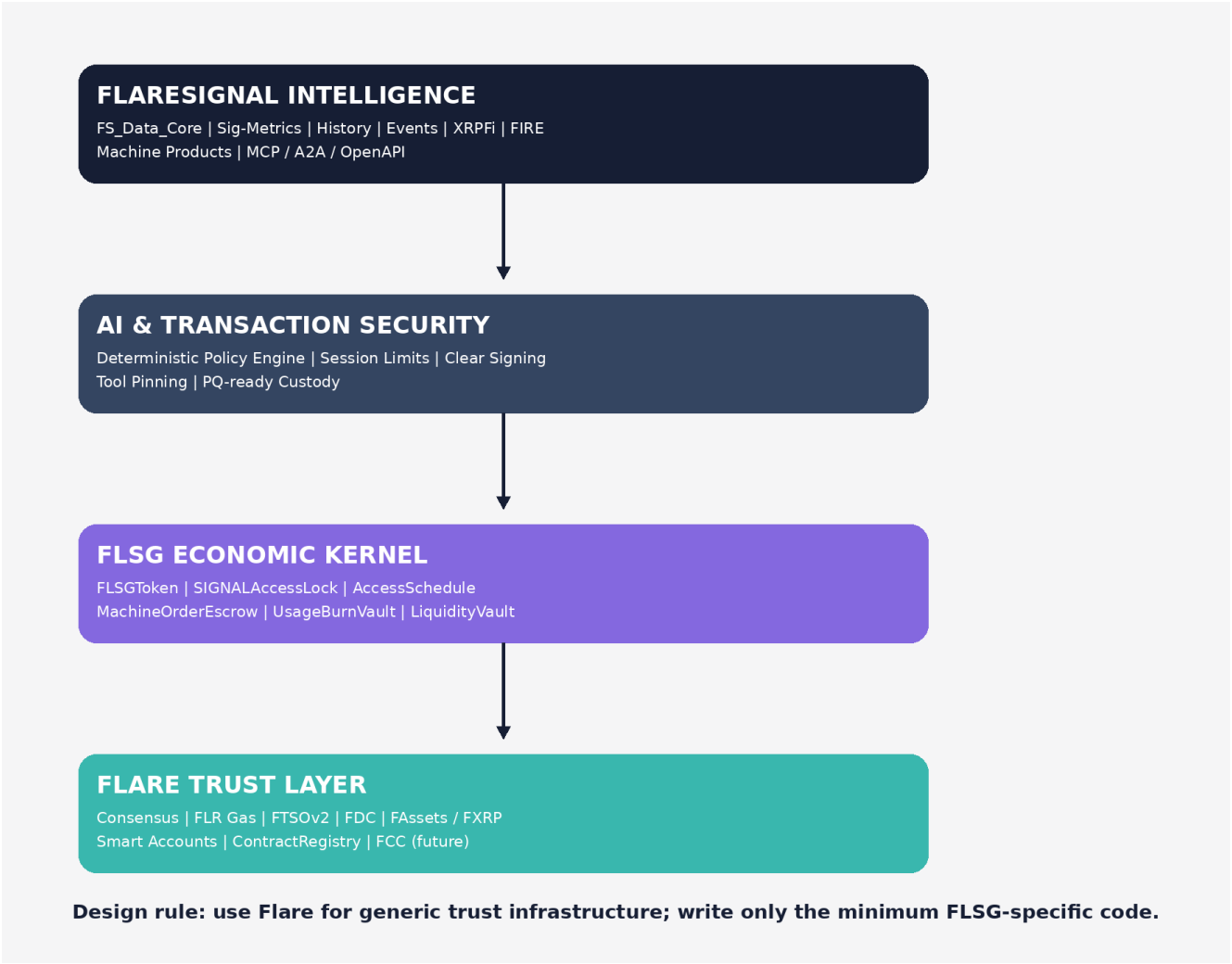
*Simple constant-product/full-range approximation ignoring fees and concentrated-liquidity effects. Actual launch execution follows the mechanics of the selected venue.

Liquidity-gated launch

If sufficient legitimate liquidity is not achieved, the proposed response is refund/postponement rather than launching an easily manipulated market. Enhanced SIGNAL locks stay closed during the initial market/oracle maturation period.

MINIMISE BESPOKE TRUST

11. Flare-Native Architecture



Design rule: use Flare for generic trust infrastructure; write only the minimum FLSG-specific code.

FTSOv2

Decentralized pricing is delegated to Flare through standard FTSOv2 feeds, an enshrined protocol supported by roughly 100 providers. [1]

FDC

Supported external-chain evidence is anchored through FDC provider consensus and on-chain Merkle-proof verification, removing any requirement for a FlareSignal contract to self-assert an external payment. [2]

FAssets / FXRP

FXRP is live on Flare and represents XRP on Flare through the FAssets system. [3] This provides a concrete XRP-originated settlement route without FlareSignal building its own XRP bridge.

Smart Accounts

Flare Smart Accounts allow XRPL users to trigger Flare actions through XRPL payments; the proof-based flow uses FDC and transaction-ID replay protection. [4] FLSG integrates these capabilities where useful rather than recreating account abstraction.

FlareContractRegistry

Official Flare protocol addresses resolve through the FlareContractRegistry rather than mutable off-chain configuration. [5]

FCC is documented as Flare TEE infrastructure and has undergone a 2026 Zellic audit, while Protocol Managed Wallets remain in development. [6][7] Future integration points are preserved, but FLSG v1 does not depend on unfinished capabilities for security.

SMALL CRITICAL KERNEL

12. Bespoke Contract Architecture

The design target is fewer than roughly six genuinely bespoke critical modules, with standard audited EVM primitives used wherever possible.

Module	Responsibility	Security posture
FLSGToken	Fixed genesis supply + native burn	Immutable; no owner monetary controls
SIGNALAccessLock	Principal, burn, expiry, entitlement view	Immutable/versioned; no admin principal path
AccessSchedule	New-lock economic schedule / epoch	Cannot change existing locks
MachineOrderEscrow	Funded -> fulfilled -> settled OR	Deterministic state machine

Module	Responsibility	Security posture
	expired -> refunded	
UsageBurnVault	One-way revenue allocation to burn	No treasury withdrawal path
LiquidityVault	Protocol-owned liquidity and bounded migration	No arbitrary generic calls

Single-source entitlement state

AccessLock exposes entitlement status directly wherever practical, avoiding a second mutable on-chain entitlement database. Fewer contracts and synchronisation paths reduce attack surface.

Constrained execution surface

Customer-money and protocol vaults expose only narrowly scoped destinations and functions. Generic execute(target,data) authority is absent from the baseline architecture, reducing the blast radius of compromised callers or control paths.

TRUST MINIMISATION

13. Non-Custodial Funds & Property Rights

FlareSignal has no discretionary access to customer principal. The custody boundary is enforced by immutable contract logic rather than an administrative promise.

AccessLock custody

During a SIGNAL lock, the FLSG leaves the user's spendable wallet and sits in the immutable AccessLock contract. This is smart-contract custody rather than discretionary FlareSignal custody.

The baseline custody pattern records an immutable beneficiary at lock activation. At expiry, claim() has no arbitrary destination parameter; even if the caller key is later compromised, principal is delivered only to the precommitted beneficiary.

calling claim merely sends principal to the precommitted beneficiary.

Order escrow

Machine payments are held by MachineOrderEscrow until successful product fulfilment. Settlement routes the predetermined shares; failure before SLA permits deterministic refund. FlareSignal cannot simply sweep funded-but-unfulfilled orders.

Emergency powers are negative powers

- Guardian may pause new locks, new orders, burn execution or liquidity changes.
- Guardian cannot mint, seize, redirect, edit expiry, alter beneficiary or move customer principal.
- Where technically safe, expired principal claims and valid refunds remain available during an emergency.

COMPROMISE-RESILIENT SECURITY

14. AI / Agent Security

The AI is never part of the trusted computing base. A model may be manipulated by prompt injection, poisoned memory, tool metadata, malicious external content or a compromised model/tool supply chain. The financial boundary remains safe even when the model behaves maliciously.

Deterministic transaction policy engine

An AI produces structured intent, not arbitrary calldata. A deterministic non-LLM policy engine decides whether the action is permitted and constructs/verifies the final transaction.

AI may request	Policy engine fixes / verifies
Product ID	Canonical contract address
Maximum spend	Function selector
Approved payment asset	Recipient / destination
Requested expiry	Nonce / replay protection
Session identity	Budget and capability limits

Session-key blast-radius limits

- Short-lived session expiry.
- Hard per-transaction and rolling spend caps.
- Product allowlists.
- Contract/function allowlists.
- No arbitrary token approvals.
- No treasury transfer capability.
- No ability to install modules or change its own security policy.

Tool / MCP security

Financial-capable tools use stable IDs, versions, schema hashes and permission manifests. Any security-relevant tool, schema or metadata change becomes a new trust decision. AI memory may retain preferences, but it does not become a source of financial authority.

Ordinary FlareSignal and PayChain account creation and intelligence purchases ask only for an email address and password credential. No name, postal address, phone number, date of birth or other identity profile is required merely to create an account or buy intelligence. Passwords remain on the secure credential-handling path. Any future regulated token distribution is a separate legal gate.

Agent-to-agent messages carry data, not authority

An agent may communicate a price, research result or product request. It cannot grant another agent permission to transfer funds. Authority is cryptographically assigned and non-transitive.

TRANSACTION INTENT

15. Smart-Account & Signing Security

Smart-account lessons

Recent smart-account audit work highlights recurring risks around access control, incomplete signature binding, replay, validation-state mistakes and account initialization. [8] FLSG machine-account permissions remain minimal and signatures bind chain, account, nonce, full operation and value limits.

Clear signing

Where wallet support permits, FLSG publishes ERC-7730 clear-signing descriptors so a user can see “Create 12-month Institutional SIGNAL Lock; principal X; burn Y; beneficiary Z; expiry T” rather than opaque calldata. ERC-7730 improves signing safety against frontend and phishing risks. [9]

Future transaction assertions

EIP-7906 research proposes post-execution assertions that can revert a transaction when its final state differs from the intended outcome, including agent spending and approval constraints. [10] FLSG remains compatible with equivalent future Flare transaction-assertion functionality without depending on it in the baseline protocol.

Frontend compromise assumption

The website is not an authority. High-value wallets and policy systems independently verify chain ID, canonical contract, function, amount and destination, so a compromised frontend cannot silently replace the intended economic action.

CRYPTO AGILITY

16. Post-Quantum Readiness

FLSG cannot honestly be described as quantum-proof while the underlying chain and conventional accounts still use quantum-vulnerable public-key signatures. The design objective is cryptographic agility: stronger post-quantum controls can be added to custody without making the monetary token upgradeable.

Proposed future protection layer

- Optional non-custodial PQ-protected smart accounts for high-value FLSG.

- Hybrid classical + post-quantum authorization for high-value movements.
- Offline post-quantum recovery credential that cannot directly spend funds.
- Precommitted recovery root / one-way delayed migration path.
- PQ-protected treasury and liquidity vault authorization when mature and audited.
- Immutable versioning: ProtectedAccountV1 -> V2 rather than upgrading a live monetary contract.

Standards-based cryptographic agility

NIST standardized ML-DSA and SLH-DSA for post-quantum digital signatures and published a 2026 transition roadmap away from quantum-vulnerable cryptography. [11][12] FLSG introduces no bespoke cryptographic algorithm; production adoption waits for mature, audited EVM/Flare implementations.

implementations.

Quantum design objective
A future break of a conventional wallet signature is not sufficient, by itself, to release FLSG held in a protected account or protected protocol vault.

SECURITY ENGINEERING

17. Testing, Formal Verification & Audit

Invariant-first development

Security properties are written before Solidity. The 2026 Orchard vulnerability in Zcash is a powerful lesson: an under-constrained cryptographic property survived years of expert scrutiny before Taylor Hornby, using targeted AI-assisted security research, found an exploit that could have enabled unlimited undetectable counterfeiting. [13]

Layered test stack

1. Unit and integration tests for normal and abnormal flows.
2. Static analysis on every security-relevant change.
3. Stateful fuzzing across caller, amount, timing and call-sequence space.
4. Mutation testing: intentionally weaken checks and require the test suite to catch the change. Trail of Bits reported a high-severity drain bug found through mutation testing that ordinary coverage had missed. [14]
5. Property-directed AI red-team: give an agent a security invariant and ask it to find a counterexample.
6. Formal verification of the most valuable invariants using independent approaches where funding permits.
7. Independent professional human audit and remediation review.

Value-at-risk canary

A Mainnet canary contract can impose immutable maximum TVL and per-position limits before a final production version is deployed. The canary is not upgraded into production; the final version is a new immutable deployment after evidence, monitoring and review.

REPRODUCIBLE DEPLOYMENT INTEGRITY

18. Deployment & Supply-Chain Security

Source-code security is meaningless if a compromised workstation, CI system or dependency deploys different bytecode. Mainnet deployment therefore needs reproducible build integrity.

Audited commit -> pinned compiler -> pinned dependency versions and hashes -> reproducible build -> independent bytecode reproduction -> Coston2 deployment -> Mainnet deployment -> verified source and published bytecode hash.

Authority separation

- Deployer address is separate from founder custody and holds negligible value after deployment.
- Guardian is separate from treasury authority.
- Operating treasury is separate from long-term reserve treasury.
- Burn executor, AI session signer and liquidity authority are separate capabilities.
- No single signer compromise authorizes high-value movement.

AI-generated code

AI may assist with Solidity, tests and vulnerability research, but it never has production deployment authority. Generated code passes the same repository, testing, review, audit and reproducible-build gates as human-written code.

PUBLIC VERIFIABILITY

19. Transparency & Live Intelligence

FlareSignal applies the same evidence discipline to FLSG that it applies across the wider platform. Public tokenomics and security surfaces make protocol state independently inspectable.

Public contract transparency

Field	Public expectation
Contract address	Published and explorer-linked
Verified source	Yes
Bytecode hash / deployment version	Published

Field	Public expectation
Upgradeable?	Explicit yes/no - critical customer-money contracts intended no
Admin authority	Enumerated
Customer-principal access	Explicitly none
Timelocks / vesting	On-chain and visible
Audit reports	Published
Formal invariants / verification results	Published where available

Live FLSG Intelligence

- Genesis supply, current total supply and cumulative native burn.
- Circulating supply, liquid float and protocol-owned liquidity.
- Founder unvested and treasury locked balances.
- Total machine FLSG locked and tier/term distribution.
- Access burn vs usage burn by period.
- Machine-product consumption linked to burn economics.
- Upcoming unlocks / expiries and renewal statistics.

Utility-led disclosure

The intended go-to-market is utility-led and organic. Product capability, economics, contracts and data are published for independent evaluation by users and machines. The architecture is intended to stand on measurable utility and transparent evidence rather than meme-style promotion or investment claims.

EVIDENCE GATES, NOT CALENDAR HYPE

20. Roadmap

FLSG progresses through explicit technical dependency gates. The roadmap is evidence-driven; phase transitions are not calendar commitments.

PHASE	TECHNICAL OUTCOME
0 - Canonical Intelligence & Payment Substrate	FS_Data_Core, machine interfaces, metering/entitlement and PayChain foundation.
1 - Production Readiness & Canonical Convergence	Production truth/source/history/component parity and certified release baseline.
2 - Machine Commerce & PayChain Proof	Production payment substrate plus autonomous purchase, delivery and verification.
3 - Protocol Specification Freeze	Economic schedules, state machines, authorities, invariants and price-source rules frozen.
4 - Coston2 Immutable Kernel	Minimal token/access/escrow/burn/liquidity contracts proven with test assets.

PHASE	TECHNICAL OUTCOME
5 - Adversarial Validation & Reproducible Security	Fuzzing, mutation, AI red-team, formal invariants and reproducible bytecode evidence.
6 - External Assurance & Mainnet Genesis	Independent audit/remediation, UK legal review, immutable genesis and protocol-owned liquidity.
7 - Market Bootstrap, Price Authority & SIGNAL Access	Market depth/history plus preferred standard FTSOv2 reference before access epochs.
8 - Usage-Derived Burn Activation	Revenue-funded one-way burn only after measured usage, liquidity and legal gates.
9 - Cryptographic & Protocol Evolution	Mature FCC/PMW/PQ capabilities via explicit immutable versions.

No phase is entered because a calendar date has arrived. Each transition requires the technical evidence, market conditions, funding and external assurance defined by the programme gate.

Phase transitions are evidence-gated rather than calendar-triggered. Each transition depends on the technical evidence, market conditions, funding and external assurance defined by the programme gate.

RISK & REGULATORY TRANSPARENCY

21. Risks & Regulatory Readiness

Technical risk

- Smart-contract defects, including unknown logical flaws.
- Dependence on underlying Flare network/security assumptions.
- Third-party wallet, RPC, dependency and frontend compromise.
- New AI-agent attack classes and model/tool supply-chain compromise.
- Future cryptographic change, including post-quantum migration.

Economic risk

- FLSG may be volatile or illiquid.
- Machine demand may be lower than modeled.
- Lock demand or burn may behave differently from illustrative scenarios.
- Usage-burn market acquisition may create execution/manipulation concerns if poorly designed.

Business risk

- Measured FlareSignal machine-product revenue precedes token-economic activation; machine demand is not assumed.
- Infrastructure, data rights, source availability and operational costs can change.

Regulatory risk

The UK regulatory environment is evolving materially. FCA guidance states that a new cryptoasset regulatory regime is expected to come into force on 25 October 2027, and firms need to assess whether their activities require authorisation. Cryptoasset financial promotions are already subject to the UK financial-promotion regime. [15][16]

Accordingly, public issuance, distribution mechanics, liquidity provision, token marketing and the proposed revenue-funded usage burn require specialist UK crypto legal/regulatory review before deployment. The whitepaper remains technical and factual and does not present investment-return claims.

PROTOCOL CONSTITUTION

22. Constitutional Invariants

INV-001 - Maximum Supply - No reachable state allows totalSupply above 1,000,000,000 FLSG.

INV-002 - No Mint Path - No callable sequence after genesis increases total supply.

INV-003 - Principal Solvency - AccessLock FLSG balance is always at least aggregate outstanding customer principal.

INV-004 - No Early Withdrawal - Principal cannot leave an active fixed-term lock before its immutable expiry.

INV-005 - Customer Sovereignty - Founder, treasury, guardian, deployer and FlareSignal administration cannot receive customer AccessLock principal.

INV-006 - Beneficiary Integrity - Principal can only be delivered to the immutable designated beneficiary.

INV-007 - Burn Finality - Native burned FLSG cannot reappear and totalSupply decreases accordingly.

INV-008 - Escrow Exclusivity - A funded order terminates through settlement or refund, never both.

INV-009 - Burn Isolation - Value committed to UsageBurn cannot return to operating treasury.

INV-010 - No Retroactive Economics - New schedules/versions cannot change an already-created lock.

INV-011 - External Replay Protection - One external payment/attestation cannot satisfy two orders.

INV-012 - Truth Independence - FLSG ownership cannot alter canonical FlareSignal metrics.

Compromise-resilience objective
No single compromised AI, tool, server, frontend, executor, guardian, deployer or individual signer is sufficient to cause catastrophic customer loss. Components capable of catastrophic failure, especially AccessLock code, are deliberately small, isolated and subjected to the strongest available verification.

CURRENT PUBLIC SOURCES

23. Research & References

[1] Flare Developer Hub - FTSOv2 - https://dev.flare.network/ftso/overview

[2] Flare Developer Hub - Flare Data Connector - https://dev.flare.network/fdc/overview

[3] Flare Developer Hub - FXRP - https://dev.flare.network/fxrp/overview

[4] Flare Developer Hub - Smart Accounts - https://dev.flare.network/smart-accounts/overview

[5] Flare Developer Hub - Flare Contracts Registry - https://dev.flare.network/network/guides/flare-contracts-registry

[6] Flare Developer Hub - FCC Reference - https://dev.flare.network/fcc/reference

[7] Flare Developer Hub - FAQs / Protocol Managed Wallet status - https://dev.flare.network/support/faqs

[8] Trail of Bits - Six mistakes in ERC-4337 smart accounts (2026) - https://blog.trailofbits.com/2026/03/11/six-mistakes-in-erc-4337-smart-accounts/

[9] ERC-7730 - Structured Data Clear Signing Format - https://eips.ethereum.org/EIPS/eip-7730

[10] Ethereum Foundation - Native transaction assertions (2026) - https://blog.ethereum.org/2026/10/05/transaction-assertions

[11] NIST FIPS 204 - ML-DSA - https://csrc.nist.gov/pubs/fips/204/final

[12] NIST IR 8547 - Transition to Post-Quantum Cryptography Standards (2026) - https://csrc.nist.gov/pubs/ir/8547/final

[13] Shielded Labs - The Orchard Counterfeiting Vulnerability (2026) - https://shieldedlabs.net/the-orchard-counterfeiting-vulnerability/

[14] Trail of Bits - Mutation testing for the agentic era (2026) - https://blog.trailofbits.com/2026/04/01/mutation-testing-for-the-agentic-era/

[15] FCA - New cryptoasset regulatory regime - https://www.fca.org.uk/firms/new-regime-cryptoasset-regulation

[16] FCA - Cryptoasset financial promotions - https://www.fca.org.uk/firms/cryptoasset-financial-promotions-and-fiat-crypto-ramp-services

[17] Flare Developer Hub - Audits - https://dev.flare.network/support/audits

[18] Flare Developer Hub - FAssets Core Vault security model - https://dev.flare.network/fassets/core-vault

Economic model provenance

Tokenomics figures in Sections 8-10 are taken from the current design workbook: SIGNAL_10-Year_Tokenomics_Simulation_v3.xlsx, with terminology normalized in this paper so the asset is FLSG while SIGNAL remains the machine-access system name. Workbook scenarios are illustrative design stress cases, not forecasts.

FLSG

Secure by architecture. Transparent by default. Trust minimised wherever code can replace promises.

PROPOSED WHITEPAPER v0.8 | NOVEMBER 2026

FlareSignal | flaresignal.io