QALS — Whitepaper

The qalarc network: an IOTA-derived, AU$1-backed credit economy

QALS project · qalarc network · 06 September 2026
generated from whitepaper.md · not legal or financial advice

QALS — The qalarc Network

A private IOTA-derived blockchain (Qalnet), its AU$1-backed credit system, compute marketplace, DeFi platform (Qalx), and identity layer

Version 1.0 — September 2026

Abstract

QALS is the native asset and connective tissue of the qalarc network, a family of AI services, trading tools, and agent infrastructure operated in Australia. This whitepaper describes Qalnet, a private Layer 1 blockchain forked from the Apache-2.0 IOTA node (itself of Sui lineage), running Move smart contracts over Starfish/Mysticeti DAG-BFT consensus with sub-second finality. On Qalnet we build four systems. First, a credit platform in which the backed token class, B-QALS, is born one-for-one from Australian dollars in a segregated reserve, burned when consumed, and redeemable at AU$1 less a 0.5% fee — a full-reserve prepaid system whose liability ledger is the chain itself. Second, Qal Compute, a marketplace where GPU jobs are escrowed, metered, attested, and settled as first-class objects, producing tradeable ComputeReceipt NFTs. Third, Qalx, a DeFi platform combining a parity rail for B-QALS, utility price discovery for the growth token G-QALS, and receivables finance against compute escrow — governed by an Oracle Committee, a timelocked Treasury Council, and a revenue-funded FloorVault providing G-QALS an honest, published, ratcheting soft floor. Fourth, Qal ID, a DID/VC identity layer for humans, AI agents, and devices, with chain-enforced spend caps and RFC 9421-signed agent authentication. Supply is fixed at 4,600,000,000 QALS of nine-decimal precision (the doof, 10^-9 QALS). We are explicit about limitations: G-QALS floor coverage begins well below 1% and reaches parity only through scale; the validator set starts as four machines under one operator; and EVM compatibility, absent from the IOTA fork, arrives as a separate, phased stack.

---

1. Introduction and Problem Statement

qalarc operates a family of applications and infrastructure: qalarc.com AI services, tradez.au, doof.ing, endispute.com.au, a Signal/WhatsApp agent hub, and a fleet of GPU compute rigs

connected over a private mesh network. Today these systems share no common ledger. Credits, invoices, data receipts, agent permissions, and compute accounting each live in a different silo — a database here, a Stripe billing account there, a spreadsheet somewhere else.

The consequences:

- **Balances do not compose.** A credit at one app cannot pay for a service at another without an internal reconciliation process that no customer sees and no auditor enjoys.
- **Machine-to-machine payment is impossible.** Classical SaaS billing cannot express "this AI agent pays that GPU rig 40,000 doofs per hour of inference." This is precisely the case the original IOTA project was invented to serve, and precisely the case card-based billing cannot.
- **Provenance is unverifiable.** When an AI system produces a report, a render, or a trading signal, there is no durable, independently checkable record of which model produced it, on what hardware, from which inputs, at what cost.
- **Agent safety is a policy, not a property.** An AI agent with a payment credential is bounded only by the prompt engineering of its operator. Nothing structural stops a runaway agent from spending without limit.

QALS is our answer: one fast, fee-subsidised, object-based chain — forked from IOTA's own open-source node — that lets every qalarc app, device, AI agent, and user share a currency, a data layer, an identity layer, a compute ledger, and a market. The chain is private not because we distrust public networks, but because a private deployment of a public, audited codebase gives us the three things a young network cannot otherwise have: a controlled validator set, subsidised gas for our users, and the freedom to operate a prepaid-credit token model inside a defensible legal perimeter while licensing work proceeds.

The design philosophy throughout is **honesty over narrative**: where a mechanism is strong (the reserve invariant behind B-QALS), we make it structurally enforced and publicly provable; where it is weak (the G-QALS floor, which begins as a fraction of a cent), we publish the number rather than imply a guarantee.

---

2. Background: Lessons from IOTA, 2015–2026

Qalnet is derived from IOTA's node software, and the QALS supply of 4.6 billion units is a deliberate homage to IOTA's own migrated supply. It is therefore worth stating plainly what IOTA was, what happened to it, and what we are taking from the experience.

2.1 The arc

2015–2020: the Tangle era. IOTA launched with an ambitious thesis: a feeless, leaderless directed-acyclic-graph ledger ("the Tangle") would become the backbone of the Internet of Things, with machines paying machines. The implementation was genuinely novel and genuinely troubled. Consensus validity depended for its first nine years on milestones issued by a single Foundation-run **Coordinator**. The ternary hash function Curl-P drew a September 2017 disclosure from MIT DCI researchers; the Foundation's hostile response led UCL's Centre for Blockchain Technologies to sever ties — a lasting reputational scar. A 2018 seed-generator phishing scam stole over AU$10M-equivalent from roughly 85 victims. In February 2020 the Trinity wallet was exploited for roughly AU$2.3M, and the Foundation's only remedy was to **shut down the Coordinator and halt the entire network for four weeks** — the canonical demonstration of the design's centralisation. The promised decentralisation upgrade ("Coordicide," slated for 2021) never shipped in its homegrown form.

2023–2024: Stardust and the EVM L2. The Stardust protocol brought UTXO-based tokenization (first-class NFT outputs, native tokens) and an ISC-based EVM Layer 2 launched May 2024. Shimmer, the incentivised test network, served as staging. **May 5, 2025: Rebased.** IOTA performed the most consequential event in its history: it abandoned the homegrown protocol entirely and rebased the Layer 1 on a Sui/Mysten-derived, Move-based, object-centric design. The old network and its EVM shut down; balances migrated with decimals changed 6 → 9 (every balance multiplied by 1,000); **4.6 billion IOTA migrated at launch**. Delegated Proof-of-Stake replaced the Coordinator permanently, with roughly 50,000 TPS demonstrated and ~400ms average finality. Crucially for our purposes, the entire stack is **Apache-2.0 licensed**, and IOTA's documentation openly acknowledges its Sui derivation.

April 23, 2026: Starfish. IOTA's consensus engine evolved into **Starfish** (whitepaper: IACR ePrint 2025/567), a hardening of the Mysticeti DAG-BFT family that decouples consensus progress from validator synchronisation: lagging validators rejoin without blocking others, and lost data is reconstructed from protocol shards. This property — steady progress on degraded networks — matters enormously for a validator fleet that includes home machines on residential internet.

2026: infrastructure success, token failure. IOTA's trade-infrastructure business — the TWIN network and its government, WEF, and GLEIF partnerships; the ADAPT programme across Kenya, Nigeria, and Morocco — is real and unusually institutional. Yet on September 6, 2026, IOTA trades at US$0.0413 with a market capitalisation of US$191M — rank #137, down 99.27% from its December 2017 all-time high, having set an all-time low of US$0.0311 on July 31, 2026. The 4.6B supply cap was **abolished at Rebased** in favour of dynamic issuance (767,000 IOTA minted per epoch), producing real dilution (~1.5–2% annually at current usage) because fee burning is negligible without volume — and genuine adoption milestones produced essentially zero price response. Meanwhile Shimmer, the staging network, sunsets on **September 30, 2026**, its LayerZero bridge already closed August 29; and the August 29–September 1 Switchboard oracle compromise halted oracle operations across IOTA, Sui, and Aptos — a reminder that ecosystem dependencies fail too.

2.2 What we take from this

Lesson from IOTA	QALS design response
Feelessness as public tokenomics failed; feelessness as a *sponsored UX* works	Qal Pass gas station: users feel no fees because the treasury pays them, inside a network we control
The Coordinator was a single point of failure for nine years	Inherit real BFT consensus (Starfish/Mysticeti) from day one; no coordinator, ever
Speculative tokenomics without usage → dilution and collapse	QALS is first a prepaid service credit, consumed by real work; usage, not speculation, is the demand source
Abolishing the supply cap eroded trust	QALS supply is fixed at 4.6B, minted once at genesis, no protocol inflation
Great infrastructure, failed token: value capture must be designed	B-QALS captures service prepayment dollar-for-dollar; G-QALS captures growth via a revenue-funded floor
Ecosystem dependencies (oracles, staging nets) fail	Small oracle surface, own oracle committee, no dependency on third-party networks beyond optional anchoring
Ten years of engineering is valuable if the license is open	We inherit the audits, test suites, and tooling of Sui/IOTA for free, under Apache-2.0

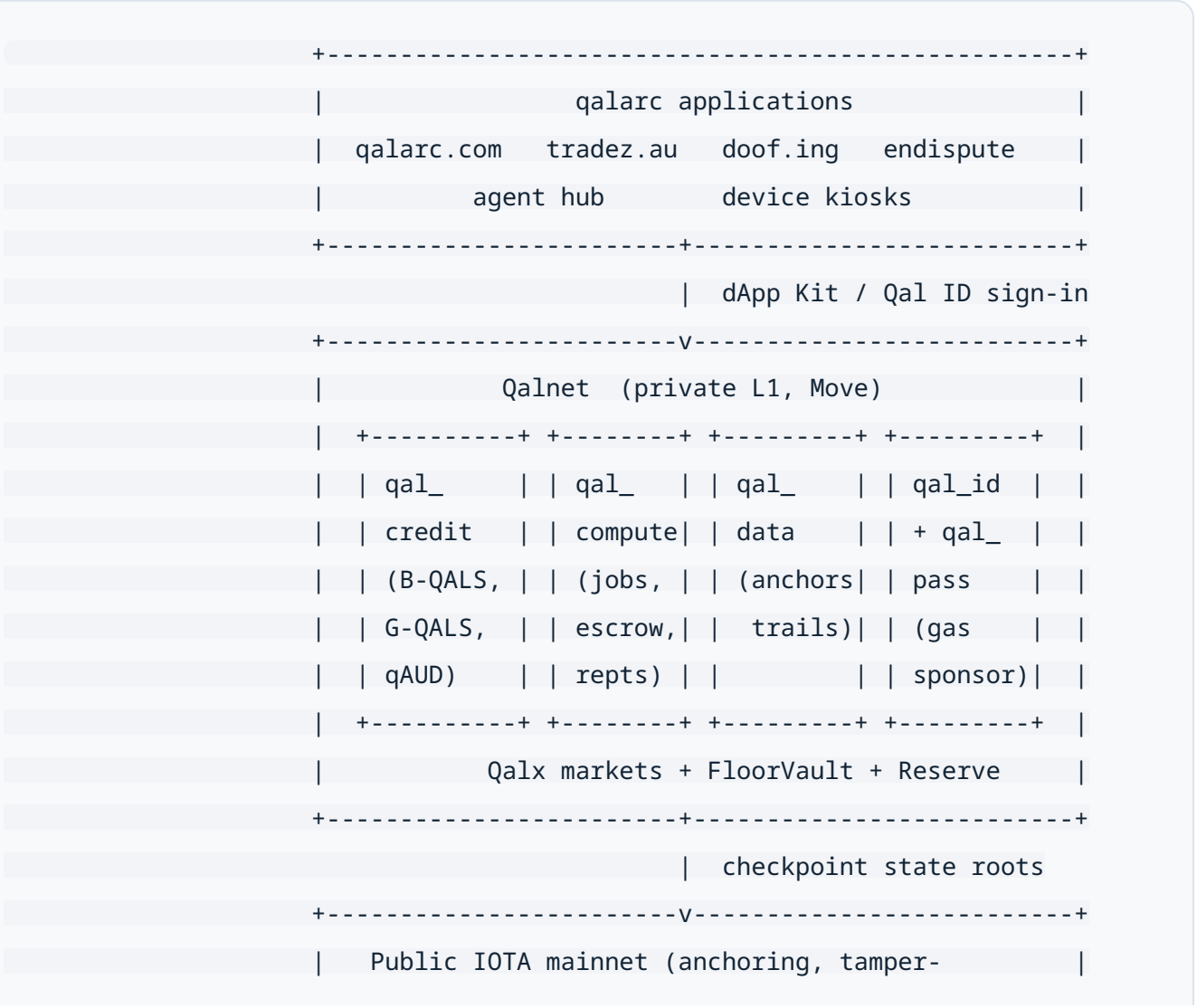
One further, technical lesson drives our phasing: the IOTA monorepo contains **no EVM implementation**. IOTA's EVM was always a separate chain (first ISC/wasp-based, later a parallel L2). A fork of the node inherits Move and the object model — not Solidity. We treat EVM compatibility accordingly: valuable, requested, and *not* free. Section 3.6 covers how we phase it in.

---

3. System Overview

3.1 Architecture

Qalnet is a private/consortium Layer 1 forked from `iotaledger/iota` (Apache-2.0): a Rust node of roughly one hundred crates, executing Move smart contracts over an object-centric ledger with Starfish/Mysticeti DAG-BFT consensus and Delegated-Proof-of-Stake mechanics inherited intact.



```
                      |   evidence; later bridged QALS representation)    |
                      +---------------------------------------------------+
```

3.2 Genesis parameters

Parameter	Value	Rationale
Network	Qalnet (private mainnet); Doofnet (public testnet, later)	Public demo surface without compromising the private perimeter
Native coin	QALS, 9 decimals; 1 QALS = 1,000,000,000 **doofs**	Homage to IOTA's nanos and to doof.ing; a doof of compute is roughly a micro-inference
Total supply	4,600,000,000 QALS, minted at genesis	Homage to IOTA's migrated supply; fixed forever, no protocol inflation
Consensus	Starfish/Mysticeti DAG-BFT, committee = validator set	Instant finality, 3f+1 fault tolerance, lag-tolerant by design
Validators at launch	4 (The Fleet)	4 validators tolerate 1 Byzantine fault; growth to 7 tolerates 2
Epoch length	24 hours	Daily key rotation and committee change
Gas	Low fixed price in QALS; 50% burned, 50% to validators	Burn keeps the ledger tidy; Qal Pass sponsors gas for end users
Storage	Refundable deposits on object deletion	Anchoring data costs a deposit; deleting it refunds you
Anchoring	Qalnet checkpoint state roots posted to public IOTA mainnet	Public tamper-evidence without a full bridge

3.3 The Fleet

The validator set at launch — **The Fleet** — runs on hardware qalarc already owns, plus cloud nodes for geographic spread and public RPC. Consensus traffic travels over the fleet's private mesh (Tailscale), which is authenticated and NAT-free.

Machine	Role
superlocal (96 GB RAM, AMD iGPU)	Validator #1, full node, indexer, dev faucet
qalcachyminirig (GPU node)	Validator #2 and compute provider
bb-mini (mini PC)	Validator #3 — Starfish's lag-tolerance is designed for exactly this class of machine
Cloud VM(s)	Validator #4 and public RPC gateway; a fourth home machine serves as spare witness

We are honest about what this is: at launch, Qalnet is a **four-machine consortium under one operator**. That is appropriate for a private company ledger — it would be indefensible for a public money network — and the roadmap opens the validator set only if and when QALS goes public.

3.4 The object model as a business data model

The inherited object-centric model means every business concept is a first-class on-chain object with an ID, owner, and version history: a `CreditAccount`, a `ComputeJob`, a `DataAnchor`, an `AgentCredential`, a `ComputeReceipt`. Objects can be owned (transferable like NFTs), shared (readable/writable by all apps via consensus), or immutable (permanent records). Escrow becomes structural rather than bookkeeping (Section 6). Deletion refunds the storage deposit, so ephemeral data (job payloads) is cheap while permanent records (invoices, receipts) carry a small forever-deposit — the "pay for data presence" spirit of the old Tangle, reborn in an object ledger.

3.5 Data transfer layer

Application-to-application data moves through a pattern we call **DataAnchor**:

1. The payload — any bytes — stays off-chain (fleet-hosted object storage or the sending machine).
2. An on-chain `DataAnchor` object records the content hash, URI, MIME type, producer signature, and timestamp.

3. Confidentiality uses a Streams-style pattern (per-channel keys distributed via the hub, per-recipient payload keys) — we rebuild the *pattern* from the archived IOTA Streams project, not the code.
4. Every anchor extends an append-only per-subject **audit trail** — compliance-grade "who saw what, when, signed by whom."
5. A light relayer posts Qalnet checkpoint roots to public IOTA mainnet, giving our private data ledger *public* tamper-evidence — the modern replacement for IOTA's old anchoring-to-Bitcoin idea.

3.6 Execution: Move first, EVM phased and separate

Move is the system language of Qalnet. All core contracts (`qal_credit`, `qal_compute`, `qal_data`, `qal_id`, `qal_pass`, Qalx core) are Move packages. The rule of thumb: value-creation logic lives in Move.

EVM compatibility is not inherited from the fork. Verification of the IOTA monorepo confirms there is no EVM implementation in it — IOTA's EVM product is a separate chain and technology line (historically the ISC/wasp stack, whose staging network sunsets September 30, 2026). Our phasing is therefore:

- **Phase 0 (now):** any EVM-facing pilot happens on *public IOTA*, whose EVM L2 and bridging already exist — QALS as a guest asset to prove demand.
- **Phase 1–2:** Qalnet runs Move only. No EVM surface to secure before there is value to protect.
- **Phase 5 (later):** an EVM compatibility layer on Qalnet via a separate stack (a wasp-style ISC emulator or a bespoke adapter), adopted only with dedicated engineering and audit. We do not promise a date.

This is less convenient and more honest than claiming "dual VM from genesis."

3.7 Wallets and feeless UX

Every qalarc web app embeds the dApp Kit ("Connect Qal ID" becomes as ordinary as "Sign in with Google"). Key material lives in Stronghold-derived secure storage wrapped by the OS enclave. **Qal Pass**, our Gas Station, sponsors gas for in-app user transactions from a treasury allocation, so users never see gas. Feelessness inside our own ecosystem was the founding IOTA promise; in a private chain, we can actually keep it.

---

4. QALS and the AU$1 Backing System

This is the centrepiece of the design, and it begins with an honest admission: **"back every QALS at AU$1" for a 4.6 billion supply means finding AU$4.6 billion of reserves.** No project has that money, and any design that implies such backing without holding it is a stablecoin lie of the kind that collapsed repeatedly in 2022. We therefore build the floor **where it can actually be funded**, exploiting the one property prepaid systems have that speculative tokens lack: **QALS is consumed by using it.**

The result is a full-reserve prepaid system with on-chain gift-card economics, wrapped in DeFi rails.

4.1 Two token classes, one reserve

	B-QALS (backed)	G-QALS (growth)	qAUD (Phase 5+)
Born from	AUD paid in (credit top-up)	genesis allocations (team, rewards, compute incentives)	AUD deposited under the licensed phase
Backing	AU$1 per token, held in the Reserve	none; FloorVault soft support	AU$1 per token, safeguarded
Redemption	1 B-QALS → AU$1 − 0.5% fee	not redeemable	1 qAUD → AU$1
Transfer	Phase 1: restricted (internal credit ledger); opens with registration	transferable per legal gates	fully transferable
When consumed	**burned**; its AUD moves Reserve → Operations (revenue)	burned (deflationary)	n/a (payment rail)
Market price	pinned near AU$1 by redemption arbitrage	floats on utility demand; floor via FloorVault	~AU$1

In Move these are two distinct `Coin<T>` types (`b_qals::B_QALS` and `qals::QALS`) plus one shared `Reserve` object. The compiler enforces that B-QALS **can only be minted inside**

`reserve.mint_against_aud()` after an oracle-confirmed AUD deposit, and only burned by the consumption and redemption entry points. Provenance is structural, not bookkeeping.

4.2 The Reserve and the invariant

Structure. AUD is held at a sponsor bank/payment institution in a **segregated client-money account**, never mixed with operating funds. The on-chain mirror is a `Reserve { aud_cents, liabilities_b_qals, last_attestation_ms }` object updated by the Oracle Committee. Because the chain itself records every B-QALS, **the chain is the liability ledger** — anyone can read outstanding liabilities directly.

The invariant — the entire system in one line:

```
AUD_in_reserve >= B_QALS_outstanding x AU$1.00    — at every block, provable monthly
```

The money flow:

```
        AU$1 --credit sale--> Reserve (segregated) --mint--> 1 B-QALS to user
                                    |
        user consumes a service     |  (services priced in QALS)
                                    v
                     B-QALS burned; AU$1 moves Reserve -> Operations (revenue)
                                    |
        user redeems instead        v
                     B-QALS burned; AU$0.995 -> user (0.5% fee -> FloorVault)
```

Three details make this more than a promise:

- **Rewards are born backed.** Marketing, referral, and compute-reward budgets do not mint free credit; they *buy* B-QALS at AU$1 from Reserve issuance. No unbacked issuance ever touches the credit class.
- **Pricing is literal.** Services are quoted in AUD and payable 1:1 in B-QALS. "1 QALS = AU$1 of qalarc credit" is true by construction, not by marketing.
- **Proof of reserves is monthly and public.** (a) a bank/PSP statement with a reconciliation letter; (b) on-chain liabilities readable by anyone; (c) a published dashboard. Quarterly, an independent accountant performs agreed-upon-procedures checks; annually (from Phase 5), a full audit. A reconciliation discrepancy above 0.5% **auto-pauses minting**.

Mint and redeem controls. Minting is rate-limited (Phase 1: at most AU$250,000/day) and gated on a confirmed deposit. Redemptions above AU$10,000 per account carry a 24-hour time-lock (an AML/fraud window). Because the system is full-reserve, a bank run poses no insolvency risk: first-come is still fully served.

Reserve composition. Once the balance exceeds AU$250,000, a tranche may be held in liquid Australian government T-Bills; any yield accrues to the FloorVault, never to operations.

4.3 Worked example

- **Month 1.** Customers top up AU$40,000 → 40,000 B-QALS minted; Reserve holds AU$40,000. AU$28,000 of services are consumed → 28,000 B-QALS burned, AU$28,000 recognised as revenue, leaving AU$12,000 of Reserve against 12,000 B-QALS outstanding. The invariant holds trivially. Separately, a AU$5,000 marketing budget buys 5,000 B-QALS from issuance — those rewards are 1:1 backed too.
- **Month 18 (illustrative).** AU$500,000/month of top-ups at 22% gross margin feed the FloorVault roughly AU$22,000/month plus fees. Against 60 million circulating G-QALS, the floor is **AU$0.0004 per token**. Coverage is honest, tiny, and growing. We show this number publicly because pretending otherwise is how projects die.

4.4 G-QALS and the FloorVault

G-QALS — the growth class, comprising the team, investor, ecosystem, compute-reward, liquidity, and validator allocations — carries **no redemption claim**, and we say so. Its "minimum support" is the **FloorVault**:

- **Funding:** 20% of qalarc service gross margin + 100% of redemption fees + 50% of Qalx trading fees + 100% of compute slash-takings, converted to qAUD/B-QALS and locked in the vault.
- **Protocol floor** = `floor_vault_balance / G_QALS_circulating`, expressed in AUD terms.
- **Ratchet:** governance may raise the floor (48-hour timelock); it can **never** be lowered — enforced structurally in Move via a typestate pattern, not by policy.
- **FloorBot:** an always-on bid on Qalx at `floor − ε`, backed by the vault. The floor is therefore not a promise; it is a visible order on a public book. Sales into the bot refill the vault; the market price may exceed the floor freely.

- **The AU$1 pathway, stated honestly:** if the business scales such that FloorVault coverage reaches 1.0, G-QALS *graduates* to full AU$1 backing and merges with B-QALS (governance vote plus legal review). Until then the dashboard says exactly what is true: *"G-QALS floor coverage: 0.7% → target: parity, funded by revenue, not promises."*

4.5 What we tell people

> Every QALS you buy as credit is backed dollar-for-dollar by Australian money in a segregated reserve — redeemable any time, and burned when you use it. The growth tokens that reward our team and network are not backed; they are supported by a revenue-funded floor that we publish every month.

That is a system a regulator, a bank partner, and a grandmother can all understand — and it is the difference between QALS and every unbacked "one-dollar-pegged" token that came before.

---

5. The Credit Platform

5.1 Three kinds of money

Layer	Instrument	What it is
L1	**QALS** (B-QALS / G-QALS classes)	prepaid service credit and gas
L2	**qAUD** (later)	tokenised AUD claim, 1:1 redeemable, post-licensing
L3	**Metered credits** (doofs)	per-app internal meters priced in QALS-denominated rates

5.2 Core objects

The `qal_credit` package defines three objects:

- `CreditAccount` — one per user or agent: escrowed prepaid balance, membership tier (0 free / 1 plus / 2 pro / 3 whale, NFT-gated), a rolling `spend_cap_epoch`, and spend-to-date this epoch.

- **`Hold`** — a pre-authorisation object, exactly like a petrol-station card pre-auth: a maximum estimated cost is held (not spent) with an expiry timestamp; unused amounts auto-release, so orphaned holds are impossible.
- **`Subscription`** — a constant drip from account to treasury per epoch; pausing or cancelling is one object mutation, with no external billing provider in sight.

A metered AI call, for example, is two small on-chain operations: `place_hold(account, est_cost)` before inference, then `settle(hold, actual_doofs)` after — the difference releases automatically. Gas for both is sponsored by Qal Pass.

5.3 Pricing and metering

Each app runs a small **meter agent** that measures usage (tokens, seconds, gigabytes, jobs) and co-signs settlement. Prices live in an on-chain shared **`PriceTable`** object mapping service keys to doofs-per-unit, updated by treasury multisig: prices are public, auditable, and every change is an event users can subscribe to. One million tokens of LLM context, one GPU-hour, one anchored gigabyte — everything normalises into doofs, so **one wallet pays for everything** and providers become comparable by `price x benchmark-normalised-time`.

5.4 Agent spend caps: the killer feature

Every AI agent's `CreditAccount` carries a hard `spend_cap_epoch`. An agent **literally cannot exceed its allowance** — enforced by the chain, not by prompt engineering. Combined with identity-layer credentials (Section 8), this is what makes agentic commerce safe enough to actually run: the worst-case behaviour of a compromised agent is bounded, provably, at genesis of its account.

5.5 Cross-app surfaces

qalarc.com gets a balance widget and per-agent spend dashboards; tradez.au prices trade credits in QALS and notarises invoices; doof.ing mints NFTs and micro-tips in doofs with sponsored gas; endispute.com.au notarises dispute files and releases escrow to the winning party — on-chain escrow is a natural fit for a dispute platform; devices join via QR `qal:pay` URIs with DID-keyed wallets and tiny caps. Any new qalarc app plugs into the same four contracts on day one.

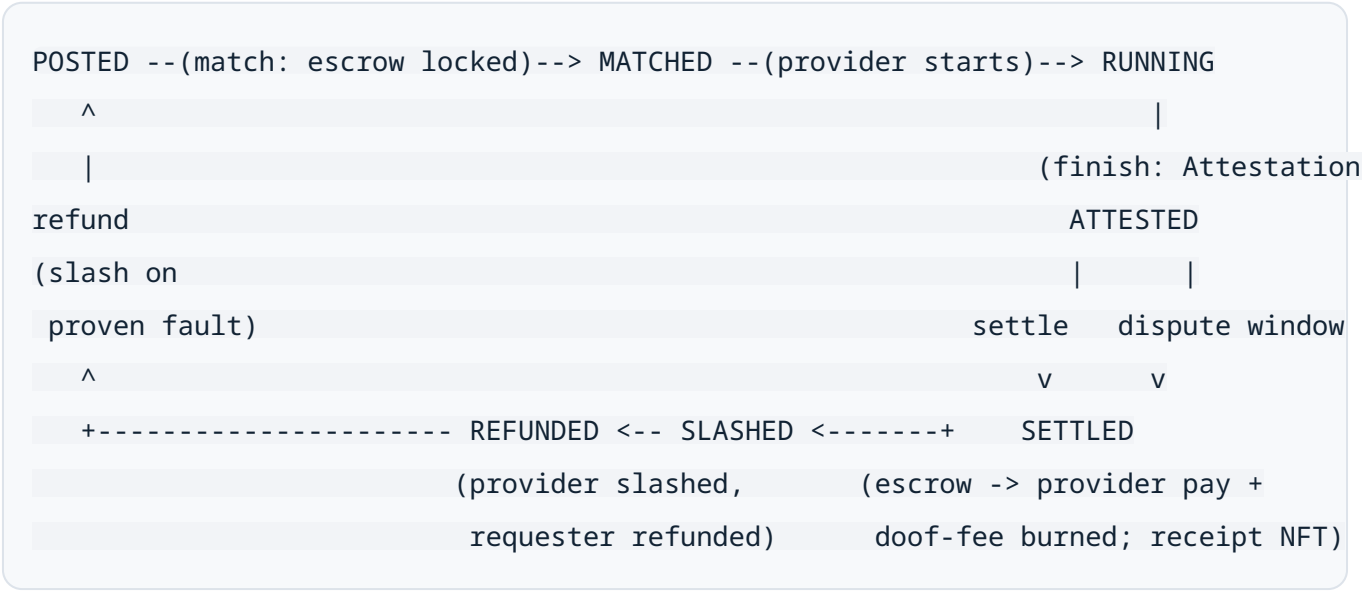
Deferred credit (Zip-style "pay over four fortnights," a reputation object recording on-time settlement, and on-chain `CreditLine` objects with per-epoch interest accrual) is a **Phase 6** item behind Australian Credit Licence work, and deliberately not pre-announced.

---

6. Qal Compute: The Compute Marketplace

6.1 Compute as an object

The original IOTA pitch — machines paying machines for metered work — finally gets adequate data structures in the object model. A **compute job is an object with a lifecycle**, not a row in a database. Escrow is trivial because Move's linear types make QALS inside an escrow un-double-spendable. Proofs are objects too: attestations, benchmark scores, and receipts reference the job by ID. And because finality is sub-second, job state machines never wait minutes for confirmation.



- **Match.** A matcher (off-chain bot, or provider-pull like Akash) calls `match(job, provider)`; escrow transfers from the requester's `Hold`.
- **Settle.** `settle(job, actual_cost, attestation)` releases `min(actual, escrow)` to the provider and returns the remainder. Metering comes from provider-signed telemetry cross-checked against wall-clock — no trusted oracle needed for v1.
- **Slash.** If telemetry contradicts reality (timeout, failed verification re-run), the job resolves SLASHED: requester refunded, provider stake slashed. Reputation is the economic security deposit.

- **Receipt.** Every settlement mints a **ComputeReceipt NFT**: job ID, model and input hashes, GPU-seconds, cost, provider DID.

6.2 Trust machinery

Mechanism	Function
Qal Bench oracle	Fleet-submitted benchmark runs (MLPerf-lite subsets, tokens/second per accelerator class), medianised on-chain into `benchmark_score`
Provider stake	Slashed on proven fault
Optimistic verification	High-value jobs re-run on sampled inputs by a second node; mismatch triggers dispute (~1.1x compute cost, optional)
Attestation signatures	Provider DID signs telemetry and output hashes; a verifier optionally countersigns
Hierarchies accreditation	Devices must hold a `can-run-jobs` accreditation to register; a compromised rig is revocable without touching code
Anchoring	Every attested job's output anchor is checkpoint-anchored to public IOTA

6.3 Tradeable compute

- **ComputeVouchers** — transferable objects redeemable for X GPU-hours at any provider: buy compute for someone else, gift it, or sell it on Qalx. Linear types make double redemption impossible; one-time `receive` consumes the voucher.
- **JobBundles** — aggregate thousands of micro-jobs (e.g., 10,000 inferences) into one escrow and one settlement, amortising state-machine overhead so per-inference billing stays gas-cheap.
- **ComputeFutures** (Phase 4) — `ComputeFuture { deliver_by, gpu_hours, strike }` locks tomorrow's capacity at today's price. With our own rigs as counterparty this is capacity planning, not gambling; it stays internal, with no public derivatives market.

6.4 Tokenomics and provenance

A slice of every settlement is **burned**; the compute rewards pool (10% of supply) pays providers over ten years. As usage grows, burn grows — a Render-style burn-mint equilibrium, simplified

because qalarc is its own anchor tenant: the fleet's LLM proxy and local-model serving consume the marketplace from day one, so it never has a cold-start problem. Real hardware at residential electricity rates (~AU$0.30/kWh) stays profitable for inference-sized work when priced against per-token API rates, which typically carry a 5–20x markup over raw compute.

Every AI artefact qalarc ships can carry a receipt — model hash, input anchors, GPU job, provider, cost, requesting agent DID. endispute can prove "this report was produced by model X on date Y for client Z, unmodified since"; enterprise clients get C2PA-style lineage without trusting us, because they verify the chain. **Compute tracking is AI accountability.**

---

7. Qalx: DeFi Platform and Safety Systems

7.1 Two-sided design

Qalx is the financial layer where B-QALS, G-QALS, qAUD, and compute receivables meet markets. **Inside the wall (Phase 1):** an internal credit ledger and sponsored exchange where apps settle, the treasury manages liquidity, and agents pay each other — no public speculation. **Outside the wall (Phase 5+):** a registered spot venue for QALS/qAUD pairs, KYC-tiered, with the same safety stack.

7.2 Market stack

Layer 1 — the parity rail (B-QALS/qAUD). 1 B-QALS redeems at the Reserve for AU$1 − 0.5%; qAUD redeems 1:1; arbitrage pins the pair within a few basis points. Volume here equals total credit throughput — Qalx earns 1–5 bps on money the business already moves, immune to price volatility.

Layer 2 — utility discovery (G-QALS/qAUD). A constant-product AMM to start (0.25% fee: 20 bps to LPs, 5 bps to treasury), upgradeable to concentrated liquidity. The **FloorBot** posts the always-on bid at `floor − ε`: the floor is a visible order, not a promise. Liquidity is treasury-seeded, then supplemented by protocol-owned liquidity bought with FloorVault excess.

Layer 3 — receivables finance. The differentiated DeFi primitive: lenders supply qAUD/B-QALS into a vault; borrowers post **ComputeReceipt NFTs as collateral**. A receipt is an on-chain, attested, escrowed future payment — LTV 60–80% by provider reputation. The default path is not an oracle guessing "what is this NFT worth": the job's escrow already sits on-chain, and

liquidation simply claims it. Invoice finance (endispute escrowed settlements) uses the same vault pattern — Australia's invoice-factoring market on chain rails.

Qalbook (Phase 5). A central-limit-order-book port of DeepBook v3 (Apache-2.0, verified): shared-object CLOB with limit/post-only/market orders, native flash loans, and epoch-governed fees, with DEEP tokenomics stripped and replaced by QALS. The trigger is flow — more than AU$1M/month of internal volume, or external users post-registration. The port is consensus-critical and carries real engineering risk (framework drift between the Sui and IOTA lineages); it ships only after dedicated Move engineering and an external audit.

7.3 Who watches the money

System	Design
Oracle Committee	3-of-5 multisig (three fleet machines, one cloud signer, one external key). Publishes signed Reserve and FloorVault statements on-chain; any two members can trigger investigation mode
Treasury Council	2-of-3 multisig for parameter changes (fees, floor ratchet, LTVs) plus a **48-hour timelock**; proposals and executions are on-chain objects with event streams
Emergency pause	Council multisig can pause mint/redeem/pools; a pause **auto-expires in 72 hours** unless renewed with a published reason, and every pause triggers a public post-mortem page

7.4 Invariants enforced on-chain, not on paper

```
// The Reserve's entry points enforce, structurally:
assert!(reserve.aud_cents >= b_qals_supply * 100);      // mint gate
mint_against_deposit(...) cap AUD_DAILY_LIMIT;          // rate limit
redeem(...) enforce TIMELOCK_10K;                       // large-redemption window
// FloorVault ratchet: floor_new >= floor_old — enforced by typestate
```

Flash-loan resistance comes from a simple rule: parity logic reads **Reserve state, not pool price**, as the source of truth for mint and redeem. Verification is productised: a live proof-of-reserves dashboard (supply, attested AUD, coverage ratio, history), a job attestation registry with a sampling verifier, identity tiers gating transaction sizes (tier 0: AU$500; tier 1: AU$10k; tier 2: AU$100k+), chain-enforced agent spend proofs, and an audit trail that anchors every

admin action, pause, and parameter change to public IOTA monthly. The audit ladder: inherited IOTA/Sui test suites kept green in CI; one external Move audit of the `qal_*` packages before real value moves; full audit plus a QALS-denominated bug bounty before any external trading.

7.5 Revenue model

Stream	Rate (start)
Parity rail fee	1–5 bps of credit throughput
AMM fee share	5 bps of volume (LPs earn 20 bps)
Redemption fee	0.5% → FloorVault
Receivables vault	1.5–3% flat on advances
Qalbook maker/taker	0–2.5 bps (epoch-voted, Phase 5)
Listing/verification	fixed fee for external compute providers

---

8. Qal ID: Identity and AI Agents

8.1 One DID per actor

Qal ID layers decentralised identity over the existing OAuth login users already have — layer, don't replace. Every actor holds a DID anchored on-chain as a Move object: each app user (optionally), every AI agent instance, each qalarc service, and the company root. Agent DIDs use **multi-level control**: the governance controller is the owning user's or ops team's DID, so a lost agent key never means a lost identity, and a controller can enumerate every DID it controls — an instant inventory of the agent fleet. Domain-linkage credentials bind DIDs to qalarc domains so third parties can resolve that a DID really is ours.

8.2 Credentials for agents

The qalarc identity service issues W3C Verifiable Credentials (Data Model 2.0, SD-JWT default encoding, BBS+ selective disclosure where privacy matters):

- **AgentCapability VC** — what the agent may do: `can-send-signal` with recipient allowlists and rate limits, `can-manage-tradez-jobs`, model and version claims.
- **SpendAuthorization VC** — `can-spend-up-to-X per 24h`, tied to a payment address: an x402-compatible mandate that mirrors the chain-enforced `spend_cap_epoch`. A payment facilitator can verify it without trusting qalarc infrastructure.
- **ModelProvenance VC** — which model (and weights hash) generated an output; attached as C2PA-style provenance stamps on public AI artefacts.
- **UserKYC VC** — selective disclosure of "over 18 / AU resident" without sharing documents.
- **QalRole accreditations via Hierarchies** — the company root accredits per-app issuers, which issue role credentials; revoking an accreditation revokes the whole branch instantly.

8.3 Revocation and authentication

Revocation uses per-credential-type **BitstringStatusList objects on-chain**: flipping a bit revokes an agent's permission or spend allowance with ~2-second finality, verifiable by anyone resolving the DID. High-risk credentials carry short expiry (7 days); provenance credentials are long-lived.

Service-to-service and agent-to-web authentication use **HTTP Message Signatures (RFC 9421)** with DID-anchored keys — the same stack as the emerging **Web Bot Auth** standard (now an IETF working group), where agents sign requests with a `Signature-Agent` header pointing to a public key directory. One code path therefore serves internal service-mesh auth and public-web agent identification, replacing shared API keys with individually revocable per-agent keys. qalarc plans to publish its own Web Bot Auth key directory so its agents are recognised by CDNs as they browse the public web.

Key custody uses Stronghold-derived vaults; rotation and revocation drills are part of operations, not afterthoughts.

---

9. The NFT Layer

NFTs on Qalnet are Move objects (a struct with `key` and `store` abilities) rendered by the inherited **Display** standard, traded through **Kiosk** with on-chain royalty enforcement via `TransferPolicy` — real, chain-enforced royalties on kiosk trades, not marketplace politeness. Gas Station sponsorship makes mints feel free. The family, in shipping order:

1. **Agent Identity Seals** (infrastructure first): one object per agent carrying its public key, model card hash, operator, permission scope, and a dynamic revocation flag mutated by the admin cap when an agent is compromised. Contracts check `seal.is_valid()` before accepting agent-signed messages. Small, quiet, high-leverage.
2. **ComputeReceipt NFTs** — the public debut: one per settled GPU job, with metadata **fully on-chain** (rig, model, watt-hours, duration, job and output hashes) so provenance never depends on external hosting. Machine-minted, instantly legible, and tied to real revenue rather than art-market speculation. They double as the collateral of the receivables vault (Section 7.2).
3. **Doofs** — the generative PFP collection for doof.ing: on-chain trait maps, sponsored allowlist mints, on-chain randomness, a burnt mint-cap to finalise the collection, dynamic traits that mutate as holders participate, and 5% kiosk-enforced royalties. Doofs double as avatars across qalarc sites.
4. **Membership tier NFTs** — Leaf/Bark/Canopy tiers gating credit limits and discounts; credits drawn down mutate the object, so the subscription is its own audit trail.
5. **AI-art with signed provenance** — model, version, prompt hash, seed, and the generating agent's Ed25519 signature as on-chain attributes: collector-grade anti-slop provenance for agency work.
6. **Digital Product Passports** (later) — per-item objects with immutable origin plus mutable service history, aimed at Australian exporters facing EU product-passport regulation.

---

10. Tokenomics Summary

Total supply: 4,600,000,000 QALS, fixed. Minted once at genesis. No protocol inflation — validator rewards are paid from a dedicated allocation, not from emissions. This is a deliberate inversion of IOTA's choice to abolish its cap at Rebased; a fixed supply with burn-on-usage is the deflationary mechanism, and it only bites when the network is actually used.

Allocation	%	QALS	Vesting / use
Ecosystem and user rewards	25%	1.15B	faucet, cashback, community
Treasury (qalarc)	20%	0.92B	4-year programmatic; funds Qal Pass sponsorship
Team and founders	15%	0.69B	4-year linear, 1-year cliff
Investors and partners	10%	0.46B	3-year linear, 1-year cliff
Compute rewards pool	10%	0.46B	paid to compute providers over 10 years
Liquidity and Qalx seeding	10%	0.46B	LP positions (QALS/qAUD and pairs)
Validator subsidy	5%	0.23B	epoch rewards to The Fleet and future validators
Airdrop to qalarc app users	5%	0.23B	identity-gated via Qal ID — also the identity pilot

Two classes carry this supply. **B-QALS is not allocated from these buckets at all** — it exists only as minted against AUD deposits (and budget-purchased rewards), so the credit class's supply equals its reserve, always. The buckets above are the G-QALS class, whose value thesis is utility demand plus the revenue-funded floor, never a redemption claim. A whimsical tertiary unit, the **trit** (1 QALS = 3^20 trits), exists purely to honour IOTA's ternary past in documentation and merchandise.

The legal posture in Phase 1 is a **prepaid service credit**: buy QALS, spend them on qalarc services, inside a closed loop. Public trading, listing, and external transferability are deliberately deferred until the registration and licensing work concludes. This document is not legal advice, and no public sale should be conducted without a written fintech legal opinion.

---

11. Roadmap

Phase	Scope	Exit criteria
0 — Pilot (weeks)	QALS as a guest asset on public IOTA (Move and its EVM L2); demand proof with existing tooling	real usage data; go/no-go on the fork

Phase	Scope	Exit criteria
1 — Qalnet fork (1–2 months)	Fork the node; custom genesis (4.6B to treasury, Fleet validator keys, 24h epochs); `qalnet-dev-1` devnet with faucet and explorer; Move-only	devnet stable; first DataAnchor payload from the agent hub
2 — Core contracts and apps (2–3 months)	`qal_credit`, `qal_data`, `qal_compute`, `qal_id`, `qal_pass`; The Fleet live (4 validators over the private mesh, cloud RPC); metering wired into the agent hub's LLM calls; tradez and doof integrations; closed beta, identity-gated	real jobs and credit flows settling on Qalnet daily
3 — Markets (months 3–6)	Reserve + redemption desk + proof-of-reserves dashboard; Qalx AMM + FloorBot; vouchers tradeable; NFT seals and ComputeReceipts; Doofs mint	invariant provable monthly; Qalx volume > AU$1M/month internal flow
4 — Open surfaces (months 6–12)	Doofnet public testnet with faucet; minirig/bb-mini as providers; Qal Bench oracle; optimistic verification; JobBundle batching; external compute providers with stake and accreditation	external providers earning; receipts financing live in the vault
5 — Regulated expansion (12+ months)	qAUD issuance under the licensed/registered phase; transferability of B-QALS; Qalbook CLOB (audited); anchoring and bridging to public IOTA; T-Bill reserve tranche	registered venue operating with the same safety stack
6 — Credit and beyond (later)	Qal Instalments behind Australian Credit Licence work; reputation objects and credit lines; device/POS pilots; EVM compatibility layer as a separate, audited stack	deferred credit products, not pre-announced

---

12. Risks and Honest Limitations

We list these not as disclosures required by convention, but as the numbers and structural facts a serious reader needs.

The G-QALS floor starts tiny. The FloorVault is funded from margins that start small: at illustrative month-18 scale, the floor is AU$0.0004 per G-QALS, with coverage below 1%. Parity

requires either a multi-billion-dollar business or deliberate small-float discipline. We publish coverage monthly precisely because it will be unimpressive for a long time.

Qalnet is initially centralised. Four validators, three of them home machines on residential internet, all ultimately under one operator. This suits a private company ledger and nothing larger. Starfish's lag-tolerance mitigates outage risk; it does not mitigate governance concentration.

The fork inherits risk with the code. We inherit ten years of Sui/IOTA engineering and their audits, but inherited audits are not audits of our genesis, our Move packages, or our configuration. An external audit of `qal_*` packages is required before real value moves; a full audit and bug bounty before any external trading. Upstream moves quickly — fork maintenance is a standing cost, and confining our changes to genesis plus Move packages is the mitigation.

EVM is not free and not inherited. The IOTA repository contains no EVM implementation. Any claim of day-one dual-VM capability would be false; the EVM layer is a separate, phased, audited project (Section 3.6).

Parity depends on scale. The parity rail and FloorBot economics only matter when credit throughput is material. Qalx starts as a thin internal settlement layer, and its receivables vault depends on compute demand that qalarc's own workloads must initially supply.

Oracle and committee trust. The Reserve mirror is only as trustworthy as the 3-of-5 Oracle Committee and the monthly bank attestations behind it. The chain proves liability; the bank proves the asset. Between attestation dates, trust — not math — covers the gap, which is why discrepancy guards and auto-pause exist.

Regulatory risk is the highest-severity item. Whether QALS, qAUD, or Qalx activities constitute financial products in Australia is a legal question this document does not answer. The prepaid framing, the phased deferral of transferability and trading, and the refusal to pre-announce credit products are mitigations, not immunities. Nothing here is legal advice.

Liquidity chicken-and-egg on Qalx. An AMM bootstraps from one treasury deposit, but depth beyond that requires volume, and all trades in a pair serialise on one shared pool object — a known throughput ceiling with a known sharding escape hatch, monitorable but real.

Ecosystem dependency. Anchoring ties us lightly to public IOTA; a hypothetical IOTA failure would cost us tamper-evidence, not funds — the fork is self-contained under Apache-2.0 — but the Shimmer sunset of September 30, 2026 and the August 2026 Switchboard compromise both demonstrate that ecosystem services disappear, and designs should not lean on them.

Key management for agents remains hard. Stronghold storage, multi-level DID control, caps, and revocation drills reduce but do not eliminate operational risk. The chain-enforced spend cap bounds the blast radius; it does not prevent the explosion.

---

13. Conclusion

IOTA spent a decade proving two things: that the machine-economy problem is real, and that a speculative public token is a poor way to fund its infrastructure. The engineering that survived — an object-centric Move ledger with DAG-BFT consensus, gas sponsorship, identity products, and a decade of audits — was open-sourced under Apache-2.0, where any serious operator can inherit it.

QALS is that inheritance, deployed where its economics actually work: inside a real business with real demand. Every element of this system follows the same discipline. B-QALS is backed because it is born from deposits and burned by consumption, with an invariant anyone can check. The G-QALS floor is honest because it is a visible order funded by revenue, published monthly, starting small. Agent spending is safe because the chain enforces caps rather than hoping prompts will. Compute is tradeable because escrow is a linear type and provenance is an object. The market layer is safe because its invariants are code, its committee actions are timelocked, and its pauses expire.

The measure of QALS will not be a price chart. It will be whether a GPU-hour, a million tokens of inference, a dispute settlement, and an agent's daily allowance all become objects on one ledger — and whether that ledger can be audited by anyone, including the people who do not trust us. That is the standard we have set.

Appendix: Glossary

Term	Definition
QALS	The native asset of Qalnet; fixed supply of 4.6 billion, 9 decimals
doof	The smallest unit of QALS: 10^-9 (one billionth); metering granularity for services

Term	Definition
trit	Joke/heritage unit: 1 QALS = 3^20 trits; honours IOTA's ternary era
Qalnet	The private/consortium Layer 1 blockchain forked from the IOTA node
Doofnet	The public testnet of Qalnet (faucet, demos)
B-QALS	Backed QALS: minted 1:1 against AUD deposits, burned on consumption, redeemable at AU$1 − 0.5%
G-QALS	Growth QALS: genesis-allocation token class; gas and utility; supported by the FloorVault, never redeemable
qAUD	Tokenised Australian-dollar claim, 1:1 redeemable; issued in the licensed phase
Reserve	Segregated client-money AUD account plus its on-chain mirror object; backs all B-QALS
FloorVault	Revenue-funded vault (margins, redemption fees, trading fees, slash takings) backing the G-QALS soft floor
FloorBot	Always-on Qalx bid at `floor − ε`, backed by the FloorVault
Oracle Committee	3-of-5 multisig publishing signed Reserve/FloorVault statements on-chain
Treasury Council	2-of-3 multisig with 48-hour timelock governing protocol parameters
Qal Pass	The gas-station service sponsoring user transactions for a feeless in-app UX
Qal ID	The DID/VC identity layer for users, agents, services, and devices
Qal Compute	The compute marketplace: escrowed, metered, attested GPU jobs
ComputeJob	On-chain object tracing a job through POSTED → MATCHED → RUNNING → ATTESTED → SETTLED/SLASHED/REFUNDED
ComputeReceipt NFT	NFT minted at job settlement; on-chain provenance and vault collateral
ComputeVoucher	Transferable object redeemable for GPU-hours; single-use by linear type
Qal Bench	Oracle medianising fleet benchmark scores per accelerator class

Term	Definition
Qalx	The DeFi platform: parity rail, utility AMM, receivables vault
Qalbook	The Phase-5 central limit order book (DeepBook v3 lineage)
CreditAccount	Per-user/agent object holding escrowed prepaid balance, tier, and epoch spend cap
Hold	Pre-authorisation object holding (not spending) an estimated maximum cost with auto-expiry
PriceTable	On-chain shared object mapping service keys to doofs-per-unit prices
DataAnchor	On-chain object anchoring a payload's hash, URI, and producer signature
The Fleet	The validator set: superlocal, qalcachyminirig, bb-mini, plus cloud nodes
Doofs (NFTs)	Generative PFP collection for doof.ing
Web Bot Auth / RFC 9421	Standards for cryptographically signed HTTP requests identifying AI agents
SD-JWT / BBS+	Selective-disclosure credential formats used by Qal ID
Starfish	IOTA's hardened DAG-BFT consensus (ePrint 2025/567), inherited by Qalnet