

# ARCHAI PROTOCOL WHITEPAPER

## Programmable ownership for land, housing, and community infrastructure

Version 0.2 – June 2025

### Disclaimer

*This document is for research and design discussion. It does not constitute an offer to sell securities, financial advice, or legal advice. Real-world deployments require jurisdiction-specific legal structuring, licensing, and compliance.*

---

### Table of contents

1) The thesis.....	3
Why now (the missing enabling stack).....	3
2) The problem.....	4
2.1 Ownership in real estate is structurally exclusionary.....	4
2.2 Housing markets are dominated by rentier dynamics.....	4
2.3 Buildings are opaque, so risk is mispriced.....	4
2.4 The coordination problem: land + community + infrastructure.....	4
3) Design principles (non-negotiables).....	5
3.1 Legal reality first.....	5
3.2 Yield first, not appreciation first.....	5
3.3 Anti-speculation must be engineered.....	5
3.4 Composability at the primitive level.....	5
3.5 Continuous attestations, not annual audits.....	5
3.6 Progressive decentralization with hard guardrails.....	5
4) System model: what Archai actually is.....	6
4.1 Archai is a network of “Vault Institutions”.....	6
4.2 Archai separates “use” from “ownership” (crucial).....	6
4.3 Archai is a policy engine for many civilizational models.....	6

5) Protocol architecture.....	7
5.1 Layer 1: Legal wrapper (Mass-native entity).....	7
5.2 Layer 2: Vault contracts (capital + rights engine).....	7
5.3 Layer 3: Governance module (polycentric).....	7
5.4 Layer 4: Commons + network treasury.....	8
5.5 Layer 5: Evidence layer (HomeOS + attestation streams).....	8
5.6 Layer 6: Arbitration and dispute resolution.....	8
6) Vault economics: stake-to-live, stake-to-own, and earned equity.....	9
6.1 Vault balance sheet: it's finance, admit it.....	9
6.2 Two primary stake products (clean segmentation).....	9
A) Resident Stake (stake-to-live).....	9
B) Capital Stake (stake-to-build / stake-to-yield).....	10
6.3 Earned equity: from first principles.....	10
6.4 Rent: discount as “coupon”.....	11
6.5 Surplus: the sacred object.....	11
7) Liquidity and exit without rentier speculation.....	12
7.1 Three exit paths (each with constraints).....	12
7.2 Redemption gates: preventing bank runs.....	12
7.3 “Non-speculative” as a pricing rule.....	13
8) Governance and control surfaces.....	13
8.1 Governance must be polycentric.....	13
Resident Council.....	13
Capital Council.....	14
Trustee/Guardian.....	14
8.2 Constitutional constraints (hard-coded).....	14
8.3 Evidence-based governance.....	14
9) Legal + compliance: Mass-native smart assets.....	15
9.1 The Mass integration is not optional.....	15
9.2 Smart Assets carry legal modules.....	15
9.3 Attestation streams are the regulator interface.....	16
9.4 Legal wrappers are modular (jurisdiction SDK).....	16
10) Security, risk, and solvency.....	16
10.1 Threat model (real, not crypto-theater).....	16
10.2 Controls (minimum viable institutional safety).....	17
10.3 Insurance and underwriting (network moat).....	17
11) Token model: \$ARCH and vault assets.....	17
11.1 Two token categories.....	18

11.2 Distribution design (principles, not arbitrary numbers).....	18
11.3 Regulatory posture.....	18
12) Roadmap.....	18
Phase 1: MVP (single vault).....	18
Phase 2: Multi-vault registry + portability.....	19
Phase 3: Commons layer + underwriting.....	19
Phase 4: Jurisdiction SDK (legal modules).....	19
Phase 5: Resident and operator interface.....	19
13) Appendix A: The Archai stack in one diagram.....	19
14) Appendix B: Glossary (minimal).....	20

---

## 1) The thesis

The built environment is the largest asset class humans maintain, yet its ownership rails are archaic: high friction, low transparency, rentier capture, and political brittleness.

Archai is the **protocol layer** that turns land and buildings into **programmable, arbitration-enabled, on-chain/off-chain hybrid institutions**, where:

- people can **enter via staking** (not only buying),
- **earn ownership** over time and contribution,
- **surplus is allocated by explicit constitutional logic**, and
- the whole system stays **compliant, auditable, and jurisdiction-portable** through Mass primitives and attestation streams.

Archai is not “a village.” It is the **capital + governance substrate** for many settlements and many civilizational models (Amagi and beyond).

### Why now (the missing enabling stack)

Archai becomes credible because the Momentum stack exists:

- **Mass** makes entities/ownership/consent programmable and enforceable through compliant APIs (the “legal substrate”).

- **HomeOS** makes buildings legible and continuously measurable (the “operational substrate”), and that data directly improves valuation and risk models for tokenized assets.
- **Programmable legal stacks / SEZ modules** make jurisdictional portability possible (the “jurisdiction substrate”), with Smart Assets carrying compliance and producing attestation streams regulators can verify programmatically.

In our view, this is the difference between “tokenizing PDFs” and building a real institutional layer.

---

## 2) The problem

### 2.1 Ownership in real estate is structurally exclusionary

For most people, “ownership” means a mortgage or nothing. Entry requires large down payments, creditworthiness, and legal sophistication. Exit is illiquid and expensive. Transaction costs are brutal.

### 2.2 Housing markets are dominated by rentier dynamics

The default incentives reward:

- holding scarce assets,
- restricting supply,
- harvesting rent,
- and capturing appreciation driven by policy and scarcity.

If you want abundant, well-maintained, community-aligned housing, you must **change the default incentive package**.

### 2.3 Buildings are opaque, so risk is mispriced

Traditional real estate suffers from chronic information asymmetry. HomeOS flips this: buildings can self-report occupancy, energy, maintenance, and revenue streams—creating a continuous operational evidence layer that tokenization actually needs.

### 2.4 The coordination problem: land + community + infrastructure

Land ownership is not just finance. It encodes:

- governance,

- cultural norms,
- maintenance commitments,
- collective investment decisions.

Today, these are handled by slow legal processes and weak social trust. Archai's premise is that they can be expressed as **programmable institutional logic**.

---

### 3) Design principles (non-negotiables)

Archai is not a “crypto project.” It is institutional infrastructure. So it must obey institutional physics.

#### 3.1 Legal reality first

A “token” is not a secured asset unless it maps to enforceable rights. Archai vault assets are always paired with a **real legal wrapper** (SPV, trust, foundation, coop) that holds title and signs a protocol-aligned charter. This must be made explicit and primary.

#### 3.2 Yield first, not appreciation first

We treat built assets as **cash-flow machines** (housing, services, utilities, data, commerce). Appreciation is not the core promise.

#### 3.3 Anti-speculation must be engineered

“Non-speculative” is a set of constraints:

- pricing rule,
- transfer constraints,
- premium capture rules,
- exit windows and gates,
- and governance limits.

#### 3.4 Composability at the primitive level

Mass provides primitives (entities/ownership/identity/consent/etc.). Archai composes them into “built asset institutions.” This is exactly how Momentum describes the ecosystem architecture: Archai is the tokenization layer for HomeOS buildings that integrate natively with Mass primitives.

### 3.5 Continuous attestations, not annual audits

A system this complex needs an evidence layer. SEZ Legal Stack implements: Smart should generate cryptographically secured, legally enforceable attestation streams so compliance can be verified programmatically without exposing private data.

### 3.6 Progressive decentralization with hard guardrails

Governance starts with safety: trustees, emergency powers, upgrade controls. It decentralizes only as mechanisms prove stable.

---

## 4) System model: what Archai actually is

### 4.1 Archai is a network of “Vault Institutions”

Each Archai project is a **Vault**:

- a set of on-chain contracts,
- bound to an off-chain legal wrapper,
- holding a specific portfolio of land/buildings/infrastructure,
- issuing stake positions and ownership claims,
- distributing surplus according to encoded policy.

### 4.2 Archai separates “use” from “ownership” (crucial)

Most housing systems bundle:

- right to occupy (use),
- financial exposure (ownership),
- and governance power.

Archai treats them as separable primitives:

1. **Use Rights**: the ability to live/work in a unit under certain terms.
2. **Stake Positions**: time-locked capital with defined redemption and yield.
3. **Ownership Units**: equity-like claims on the underlying entity.
4. **Contribution Credits**: earned equity via labor/service, verified by attestations.
5. **Governance Power**: parameterized, scoped, and constrained.

Separating these lets us engineer “non-speculative” systems while still raising capital.

### 4.3 Archai is a policy engine for many civilizational models

A “civilizational model” is a parameter set + governance constitution:

- rent policy,
- surplus split,
- equity issuance schedule,
- contribution valuation rules,
- secondary transfer constraints,
- commons obligations.

Archai hosts multiple models; vaults select one (and can migrate with governance).

---

## 5) Protocol architecture

### 5.1 Layer 1: Legal wrapper (Mass-native entity)

Each Vault corresponds to a legal entity (or entity set):

- **Asset SPV / Trust** holds title and bank accounts.
- **Operator entity** contracts services (maintenance, property management, construction).
- **Archai Foundation / protocol entity** holds IP, administers upgrades, manages network treasury.

Mass provides the programmable rails to automatically form and maintain these entities and encode governance, ownership tables, identity gating, and consent flows programmatically.

In other words: the legal wrapper is not paperwork. It is a programmable Smart Asset.

### 5.2 Layer 2: Vault contracts (capital + rights engine)

Vault contracts manage:

- deposits,
- stake terms,
- redemptions,
- distribution,
- issuance of ownership units,

- rent discount rules,
- secondary transfer rules,
- reserve accounts,
- liquidation/wind-down logic.

### 5.3 Layer 3: Governance module (polycentric)

Vault governance is not “one token, one vote.”

It is:

- scoped,
- role-based,
- bicameral where necessary,
- with trustee/guardian failsafes.

### 5.4 Layer 4: Commons + network treasury

A portion of surplus flows to the Archai Commons Treasury (v0.1 proposes 5–10%).

This treasury funds:

- seeding new vaults,
- resident subsidies,
- protocol R&D,
- and crucially: **risk underwriting** (insurance).

### 5.5 Layer 5: Evidence layer (HomeOS + attestation streams)

Buildings running HomeOS produce continuous operational data:

- occupancy,
- energy,
- maintenance costs,
- revenue streams.

This reduces information asymmetry and feeds valuation and risk models for tokenized assets.

Separately, the SEZ Legal Stack concept of **attestation streams** applies to compliance: prove required checks (KYC tier, AML screening, transfer restrictions) without revealing underlying private data.



## 5.6 Layer 6: Arbitration and dispute resolution

Every vault must have:

- a defined dispute path,
- arbitration clause,
- on-chain evidence admissibility rules,
- and emergency intervention rights.

Momentum's ecosystem explicitly includes an ADGM-based arbitration court for digital asset and machine-intelligence workflows; Archai vaults should route disputes into such hardened venues by default.

Momentum Monograph - Google Docs

---

## 6) Vault economics: stake-to-live, stake-to-own, and earned equity

This section closes the biggest gap in v0.1: **why lock capital up** and **how returns are actually produced without speculation**.

### 6.1 Vault balance sheet: it's finance, admit it

Each vault is a mini private equity + real estate platform with a capital stack:

- **Senior capital** (bank debt / bonds)  
Lowest risk, capped return, first claim.
- **Resident stake** (time-locked deposits)  
Medium risk, compensated via rent discounts + earned equity.
- **Programmatic equity** (vault ownership units)  
Variable return, receives residual surplus and some appreciation per policy.
- **Sponsor/operator equity**  
Paid for execution, absorbs early risk, earns if the asset performs.
- **Commons stake** (network)  
Receives protocol fee and may provide insurance backstops.

The whitepaper must explicitly model this, because solvency and redemptions depend on it.

### 6.2 Two primary stake products (clean segmentation)

#### A) Resident Stake (stake-to-live)

Residents stake capital for a term (v0.1 suggested 3–7 years).

They receive:

- a **rent discount** (economic benefit now),
- **earned equity** over time (wealth building),
- governance participation (scoped),
- and the right to redeem principal at maturity (subject to safety gates).

This is not “investing.” It is a **capitalized membership** in a housing institution.

## B) Capital Stake (stake-to-build / stake-to-yield)

Non-resident capital providers stake into the vault with:

- defined term,
- defined distribution rules (coupon-like),
- potentially equity kicker,
- no occupancy rights.

This is the instrument for scaling supply without turning the whole system into a speculative casino.

## 6.3 Earned equity: from first principles

Earned equity exists to reward **time, risk, and contribution** instead of mere wealth.

v0.1’s formula is directionally correct (equity proportional to stake \* multiplier / valuation).

But here we refine to a safer, more precise model:

Let:

- $NAV_t$  = net asset value of the vault at time t (audited + attested)
- $S_i$  = stake amount for position i
- $T_i$  = term length for position i
- $R_i$  = risk multiplier (construction > stabilization > mature)
- $C_i(t)$  = contribution score (verified labor/service)
- $\alpha, \beta$  = policy parameters

Define **Equity Accrual Rate**:

$$E_{Ai}(t) = S_i \cdot NAV_t \cdot f(T_i) \cdot R_i + \alpha \cdot C_i(t) \quad E_{Ai}(t) = NAV_t \cdot S_i \cdot f(T_i) \cdot R_i + \alpha \cdot C_i(t)$$

Where:

- $f(T)f(T)$  is convex in time (long-term alignment rewarded),
- $R_i R_i$  is bounded (no infinite multipliers),
- $C_i(t)C_i(t)$  is capped per epoch to prevent gaming.

Equity accrues continuously or in epochs, minted as **Vault Ownership Units**.

## 6.4 Rent: discount as “coupon”

We formalize:

- Let Base Rent be computed from:
  - operating costs,
  - reserves,
  - debt service,
  - targeted maintenance SLA,
  - and affordability policy.
- A Resident Stake provides a **discount** that functions like the “interest payment” on their locked capital.

This is economically coherent: the vault is borrowing from residents; residents are compensated through reduced rent + earned equity.

## 6.5 Surplus: the sacred object

Define:

$$\text{Surplus} = \text{NOI} - \text{DebtService} - \text{Reserves}$$

Then allocate surplus by policy:

- **Resident / staker distributions:** 50–70% (v0.1)
- **Reinvestment:** 20–40% (v0.1)
- **Protocol fee to Commons:** 5–10% (v0.1)

We add one missing bucket:

- **Insurance / contingency buffer** (can be part of reserves or funded via protocol treasury)

Surplus allocation is not “governance vibes.” It is the economic constitution.

---

## 7) Liquidity and exit without rentier speculation

This is where most “tokenized real estate” fails.

### 7.1 Three exit paths (each with constraints)

1. **Redemption at maturity (primary)**
  - Stake positions redeem at maturity at a deterministic rule:
    - principal less fees,
    - subject to solvency and redemption queues.
2. **Secondary transfer of stake rights (bounded)**

We formalize:

- transfers only to whitelisted (KYC/KYB) recipients,
  - premium cap relative to NAV,
  - any premium above cap is routed to Commons (anti-speculation capture),
  - mandatory disclosure of vault state.
3. **Cross-vault portability (network feature)**

As Archai evolves into a registry, positions can be portable:

    - stake exits one vault and enters another under rules,
    - enabling mobility without liquidation cascades.

### 7.2 Redemption gates: preventing bank runs

Real assets are illiquid. If you promise instant liquidity, you’re lying.

Archai uses explicit gates:

- max redemption per epoch,
- variable queue times tied to liquidity,
- emergency halts when reserves breach threshold.

This is not anti-user; it is solvency.

### 7.3 “Non-speculative” as a pricing rule

A vault can choose among policy modes:

- **NAV-bounded mode:** ownership units trade within a tight band around NAV; excess premium is taxed away.
- **Affordability-first mode:** occupancy rights are non-transferable or transferable only at cost.

- **Growth mode** (allowed but explicit): partial appreciation can accrue, but with caps and commons capture.

The key is: speculation is never the default. It's a deliberate configuration.

---

## 8) Governance and control surfaces

v0.1 led with “stake-based, resident-weighted, trustee-checked.”. We make this enforceable.

### 8.1 Governance must be polycentric

Different decisions require different stakeholders.

Recommended structure:

#### **Resident Council**

Controls:

- community rules,
- local service priorities,
- cultural programming,
- some budget allocations.

Voting weight: occupancy + contribution (not raw capital).

#### **Capital Council**

Controls:

- leverage limits,
- reserve policy,
- large capex,
- refinancing terms,
- issuance policy.

Voting weight: stake and ownership units (bounded).

#### **Trustee/Guardian**

Controls:

- legal compliance,
- emergency actions,
- fraud response,
- enforcing constitutional constraints.

Trustees are not “centralization”; they are the interface to legal reality.

## **8.2 Constitutional constraints (hard-coded)**

Certain parameters must be upgrade-resistant:

- maximum leverage,
- minimum reserves,
- affordability floors,
- anti-speculation constraints,
- disclosure requirements.

Governance cannot vote to steal the treasury or lever the vault into oblivion.

## **8.3 Evidence-based governance**

Governance proposals must reference:

- attested operating metrics (HomeOS),
- audited financials,
- risk reports,
- and predicted impacts.

This is where “building as programmable institution” becomes real.

---

# **9) Legal + compliance: Mass-native smart assets**

## **9.1 The Mass integration is not optional**

Mass provides programmable primitives:

- entities,

- ownership structures,
- financial instruments,
- identity,
- consent.

Momentum Monograph - Google Docs

Archai vaults should be implemented as a specific composition of those primitives:

- the SPV/trust is a Mass entity,
- stake positions and ownership units are Mass-native ownership tables/tokens,
- transfers are governed by identity gates and consent workflows,
- distributions use Mass fiscal rails (accounts/wallets/banking integration).

## 9.2 Smart Assets carry legal modules

SEZ Stack expresses the crucial design principle: a Smart Asset should encode jurisdiction, transfer restrictions, identity tier requirements, consent mechanisms—so compliance is intrinsic, not a separate spreadsheet.

For Archai:

- Each vault declares its jurisdictional module.
- Transfers compute the intersection of permissible actions (e.g., KYC tiers, residency rules, securities exemptions).
- The asset enforces constraints programmatically.

## 9.3 Attestation streams are the regulator interface

Regulators and auditors don't want dashboards; they want verifiable compliance.

Attestation streams prove requirements were met without exposing private data (e.g., “all holders meet KYC tier X”).

Archai uses attestation streams for:

- holder eligibility,
- transfer compliance,
- treasury flows,
- operating metrics integrity.

## 9.4 Legal wrappers are modular (jurisdiction SDK)

v0.1 mentioned modular templates (Cayman foundation, Swiss Verein, Panama company).

The correct framing:

- Create a **Jurisdiction SDK**: a set of deployable legal + compliance modules (entity type, offering exemptions, custody model, dispute resolution path).
  - Vaults instantiate within a chosen module.
  - Modules can be upgraded or migrated with governance and trustee oversight.
- 

## 10) Security, risk, and solvency

### 10.1 Threat model (real, not crypto-theater)

Primary risks:

- title defects / legal disputes,
- operator fraud,
- valuation manipulation,
- oracle corruption (data + appraisals),
- governance capture,
- liquidity cascade (redemption run),
- jurisdictional intervention,
- stablecoin/banking rail failure,
- smart contract bugs.

### 10.2 Controls (minimum viable institutional safety)

- **Segregated accounts** (operating vs reserves vs distributions)
- **Independent appraisal + HomeOS attested ops data** feeding NAV
- **Redemption gates + queues**
- **Role-based access control** for operational actions
- **On-chain + off-chain audit trails**
- **Emergency guardian powers** with transparency constraints
- **Insurance module** (protocol-level underwriting)

### 10.3 Insurance and underwriting (network moat)

v0.1 mentions underwriting network health via insurance mechanisms and a protocol token role.

Make it concrete:



- The Commons Treasury can act as a first-loss backstop for defined event categories.
- \$ARCH can be staked into underwriting pools (risk-weighted).
- Vaults pay risk-based premiums (paid from surplus).

This creates a network effect: more vaults = better risk modeling = cheaper capital.

---

## 11) Token model: \$ARCH and vault assets

v0.1: “The protocol token (\$ARCH) governs the Archai network... stake-weighted voting... access to preferred vaults... claim on commons surplus... underwriting.”

We refine this into a clean separation:

### 11.1 Two token categories

1. **Vault Assets (per-vault)**
  - Stake position tokens (time-locked notes / deposit receipts)
  - Ownership units (equity-like claims)
  - Use rights (occupancy/access claims)

These map to specific legal wrappers and are highly jurisdiction-dependent.

2. **\$ARCH (network token)**
  - Governance of protocol-level parameters and upgrades
  - Underwriting / insurance staking
  - Access coordination (optional, not core)
  - Fee capture from Commons Treasury (if legally viable)

### 11.2 Distribution design (principles, not arbitrary numbers)

- Incentivize early vault bootstrapping (but don't subsidize forever).
- Reward verified contribution (builders, maintainers, operators) more than passive speculation.
- Fund long-term commons (treasury).
- Avoid short-term emission schedules that create mercenary behavior.

### 11.3 Regulatory posture

Vault assets are likely regulated financial instruments in many jurisdictions. That is fine if compliance is embedded (Mass-native) and distributions/transfers are constrained accordingly.

\$ARCH must be treated carefully; if it has explicit cashflow claims, it may be regulated. If it is positioned as governance + underwriting collateral with carefully scoped economics, it may have a cleaner posture. This is jurisdiction-specific; Archai must be modular here.

---

## 12) Roadmap

v0.1 roadmap is directionally correct; we sharpen it into phases with explicit deliverables.

### **Phase 1: MVP (single vault)**

- One legal wrapper
- One vault contract system (stake, redemption, distribution)
- One governance constitution
- One evidence feed (basic HomeOS metrics)
- Basic KYC gating through Mass

### **Phase 2: Multi-vault registry + portability**

- Registry contracts
- Standard interfaces for vault types
- Cross-vault portability primitives
- Redemption queue standardization

### **Phase 3: Commons layer + underwriting**

- Protocol treasury routing
- Grants program for new vault origination
- Insurance pools + risk scoring

### **Phase 4: Jurisdiction SDK (legal modules)**

- Deployable templates per jurisdiction
- Integration hooks to SEZ stack modules
- Attestation stream standards for compliance

### **Phase 5: Resident and operator interface**

- Mobile dashboard

- Resident onboarding flows
  - Contribution verification tooling
  - Proposal and voting UX
- 

## 13) Appendix A: The Archai stack in one diagram

### Value + control flow (simplified)

Residents / Investors

- stake capital into **Vault Contract**
  - Vault Contract coordinates with **Mass Entity (SPV/Trust)**
  - Entity holds title + bank accounts, pays operators
  - Building ops run on **HomeOS**, producing attested metrics
  - Vault calculates surplus, updates NAV, distributes per policy
  - Protocol fee to **Commons Treasury**
  - Commons funds seeding + subsidies + insurance
- 

## 14) Appendix B: Glossary (minimal)

- **Vault:** on-chain capital + rights engine paired to a real legal entity holding the asset.
- **Stake position:** time-locked capital with defined redemption/yield.
- **Ownership unit:** equity-like claim on the underlying entity.
- **Use right:** occupancy/access claim (may be non-transferable or bounded-transfer).
- **Attestation stream:** cryptographic proof stream of compliance/operations.
- **Commons:** shared treasury and seeding/underwriting engine.