sOS Research Lab WORKING PAPER SERIES

sOS Research Lab 2026-07-01 · Barcelona

WP8 – DAO, UILL & Social Machine: Technical Architecture

Series	sOS Execution
Status	Revised v2.0 – Thesis-aligned update (3 dimensions, hybrid concatenation, 6 contract types, tensor factorization, label origin flow)
Version	2.0
Date	2026-06-30 (original: 2026-06-28)
Priority	Critical – Phase 1
Review Mode	Full Redundant Review (Branch A: Pi/DeepSeek → Branch B: Anthropic Opus 4.8 → Synthesis → Revision)
Grounded in	sOS doctoral thesis (Chapters III.13–III.17, specifically §14.1 hybrid model, §14.3 six contract types, §14.4 validation by labels, §15 UILL data collection, §16 Social Machine and three-dimensional model), WP0–WP7, budget architecture, glossary, smart contracts spec v5
The stigmergic conversion (thesis §11.5, §14.4)	Ants use pheromones to coordinate. Humans produce data. When data is classified as labels through the UILL, it functions as human stigmergy – an indirect, mediated mechanism of coordination between actions, in which the trace of an action left on a medium stimulates the performance of a subsequent action (Heylighen, 2016).

Abstract

This working paper specifies the technical architecture of sOS's three core components: the Decentralized Autonomous Organization (DAO) that governs the research program itself, the Unique Information Library Ledger (UILL) that provides personal data sovereignty, and the Social Machine that coordinates human and AI agents through stigmergic information environments. Together, these three components form an integrated governance architecture capable of processing social complexity without centralizing power. This paper defines each

component's structure, data flows, technical dependencies, and integration points. It does not prescribe specific implementations (choice of blockchain, AI models, or smart contract languages) – those decisions belong to the sandbox phase – but specifies architectural requirements that any valid implementation must satisfy.

---

1. Introduction: Why Architecture Matters

sOS is not a product. It is a research program proposing a new class of governance infrastructure. Before writing code, deploying smart contracts, or recruiting sandbox participants, the architecture must be specified with enough precision that:

- Different implementers can build compatible components.
- The design can be verified against theoretical foundations.
- Security, privacy, and decentralization properties can be analyzed before deployment.
- Trade-offs (scalability vs. decentralization, privacy vs. auditability) are explicit rather than discovered at runtime.

This paper defines the *what*, *why*, and *how-they-fit-together* of sOS's three core technical components. The *how-to-build* belongs to the sandbox experimental phase (WP14).

WP6	Barcelona Sandbox	Operational context for architecture testing
WP7	sOS Research Lab	Organizational vehicle; DAO governance defined here
WP9	Legal & Compliance	Regulatory constraints on architecture (complete: AD-SOS-13)
WP10	Tokenomics	Economic layer built on DAO + UILL (complete: AD-SOS-14)
WP14	Experimental Design	Sandbox validation of this architecture (complete: AD-SOS-15)

---

2. Architectural Principles

All three components share five architectural principles derived from the sOS thesis:

2.1 Distribution, Not Decentralization-for-Its-Own-Sake

sOS does not pursue decentralization as an aesthetic or ideological preference. Distribution is an *engineering response to complexity*: when no single node can process all relevant information, governance must be distributed across nodes. The architecture distributes where computation, validation, and decision-making benefit from distribution; it centralizes where efficiency requires it (e.g., cryptographic identity roots, canonical registries).

2.2 Data Sovereignty as Infrastructure

Every individual in sOS owns their data – not as a legal slogan but as a technical property. The UILL enforces this at the protocol level: data cannot be accessed, processed, or transferred without explicit, auditable consent managed through smart contracts. This is not a privacy feature layered on top; it is the foundation.

2.3 Stigmergic Coordination Through Labeled Traces

Governance actions leave digital traces. These traces are structured through a shared label ontology. Agents (human and AI) read the environment of traces and adjust their behavior. Coordination emerges from trace-reading, not from central commands. This is the core mechanism of the Social Machine.

The stigmergic conversion (thesis §11.5, §14.4): Ants use pheromones to coordinate. Humans produce data. When data is classified as labels through the UILL, it functions as human stigmergy – an indirect, mediated mechanism of coordination between actions, in which the trace of an action left on a medium stimulates the performance of a subsequent action (Heylighen, 2016).

Label origin (thesis §15-16): Labels do not appear from nowhere. They originate from the individual through a three-dimensional capture system:

- **Physical dimension:** GPS mobility data, activity spaces (Palmer et al., 2013)
- **Virtual dimension:** Digital activity, social networks, metadata
- **Cognitive dimension:** Emotional states, preferences, stress indicators

These three dimensions feed a tensor factorization / backpropagation model (thesis §16.5) where attributes (X) are related to labels (Y) via functions (β). The model is phenomenological: it improves with more data and more actions. Labels are stored in the UILL as JSON-LD with the sOS vocabulary.

2.4 The Hybrid Model: Concatenation of Human and Machine Domains

The sOS architecture is built on a hybrid model (thesis §14.1): the concatenation of the human domain and the machine domain. The human domain focuses on tasks, deliberation, and decisions. The machine domain performs the links, validates, and calculates interdependencies through the Social Machine. The relationship between both domains is labels and smart contracts.

This is not sequential – it is concatenated. Human validation scores and machine-computed metrics (S^2, variance, match scores) are unified into a single label space. No domain is above the other. They work together in a recursive relationship where each needs the other.

The machine domain does not make decisions. The human domain always makes the final decision. The machine domain provides the links to carry out the task.

2.5 Personal AI Serving Individuals

AI in sOS is personal, not institutional. Each individual has access to AI that understands their context, advocates for their interests, and processes governance information on their behalf. No central AI governs. Personal AIs are diverse by design – different models, different providers, different training – to prevent homogenization of "augmented" decisions.

2.6 Design for Auditability and Revision

Every component must be auditable: data provenance, decision provenance, smart contract logic, AI model behavior. The architecture must support revision without breaking existing coordination – governance rules must be upgradeable through governance itself.

---

3. The DAO: Scientific Governance Infrastructure

3.1 Purpose

The sOS DAO is not a speculative token vehicle or a generic "decentralized community." It is the governance infrastructure for a scientific research program. Its functions are:

1. **Research governance:** proposal submission, peer review, funding allocation, publication decisions.

2. **Resource management:** treasury, token distribution, compensation, grant management.

3. **Protocol governance:** architecture upgrades, smart contract modifications, parameter adjustments.

4. **Membership:** admission, reputation, participation rights, dispute resolution.

3.2 Structural Components

The DAO operates through **six smart contract types** (thesis §14.3), which form a decentralized bureaucracy where the human domain performs tasks and the machine domain provides the links:

3.2 The Six-Function Decentralized Bureaucracy

1. Propose (§14.3.1): Any individual initiates the chain. The proposal includes objectives, benefits, cons (tags), estimated execution time, number of tasks needed, and validation parameters. The proposer may or may not be the executor.

2. Validate (§14.3.2): The core of the system. Multiple humans (≥3) with relevant labels (Yn) score the proposal on a 0-10 ratio scale. The machine computes **S^2 variance** across scores:

- Low S^2 → genuine consensus → validation accepted
- High S^2 → disagreement → expand validators with same labels
- Persistent disagreement → expand label scope, bring in more diverse validators

Validation uses **redundancy for legitimacy**: whoever validates accepts responsibility. Their assessment reliability is itself tracked and validated over time by other validators. The smart contract is fully customizable: thresholds, time windows, validator counts, and asset transfer conditions are configurable per proposal type.

3. Execute (§14.3.3): Requires prior validation to activate. Includes attributes, times, task input (validated), task output (to be validated), and remuneration according to labels. An execute contract can have a linked finance contract.

4. Delegate (§14.3.4): Transfers validation power to another ID with appropriate labels. This is granular representation: *"In environmental aspects (Y_CSR labels), I value what ID x values. In economic aspects (Y_ECN labels), I value what ID z values."* Delegation can be reversed or chained. It enables individuals to participate in governance without expertise in every domain, and allows the system to transition from centralized to decentralized governance.

5. Save/Compute (§14.3.5): Decentralized data storage and machine learning computation. Any ID with stable connection and free space can store fragments of other UILLs, generating compensation. Social Machine computations are distributed across nodes. This replaces centralized servers with peer-provided infrastructure.

6. Finance (§14.3.6): Democratic funding that links validations to asset transfers. Financing is open, configurable by all members, and operates through token allocation smart contracts tied to validated contributions.

3.3 Validation by Labels (§14.4)

Each individual has attributes (X) that relate to labels (Y) via functions (β). Each problem also has its own β function. The matching algorithm finds individuals whose β function aligns with the problem's β function:

```
β(individual) ≈ β(problem) → match → individual validates
β(individual) ≠ β(problem) → no match → delegate or expand search
```

When no single individual matches, the system composes β from multiple individuals whose combined labels cover the problem space. This is the mathematical foundation of collective intelligence: the group's β exceeds any individual's β. Labels are adjusted and created through the validation process – the system learns who is reliable at validating what.

3.4 Voting Mechanisms

The DAO supports multiple voting mechanisms, selectable per proposal type:

Research funding	Reputation-weighted + conviction	Rewards sustained support, resists whale + Sybil
Protocol upgrades	Token-weighted + time-lock + guardian veto	Security-sensitive; stake-weighted with safety override
Membership admission	Reputation-weighted	Expertise matters for peer admission
Routine operations	Simple majority	Efficiency for low-stakes decisions
Emergency actions	Multi-sig council + community veto	Speed + accountability

Note on Quadratic Voting: QV is excluded from the default mechanism set because it is catastrophically vulnerable to Sybil attacks – its anti-whale property inverts when one actor can cheaply mint many identities. Until the proof-of-personhood problem has a production-grade solution (see §9.6), sOS defaults to reputation-weighted and conviction voting, which degrade more gracefully under Sybil conditions. QV may be reintroduced as an optional mechanism if/when Sybil resistance is robustly solved.

3.5 Reputation System

Reputation is non-transferable, multidimensional, and time-decaying. It is NOT a token you can buy.

Dimensions:

- **Research reputation:** earned through peer-reviewed contributions, WP authorship, sandbox participation.
- **Governance reputation:** earned through voting participation, proposal quality, review thoroughness.
- **Community reputation:** earned through constructive participation, mentorship, public engagement.

Decay function: Reputation decays exponentially with half-life per dimension:

- Research: 24-month half-life (scholarly contributions age slowly)
- Governance: 12-month half-life (active participation required)
- Community: 6-month half-life (engagement is perishable)

Reputation is stored as time-bound Verifiable Credentials (VCs) issued by the DAO. Re-attestation occurs at the half-life interval: a participant must demonstrate continued contribution to maintain reputation. This prevents entrenched power while respecting genuine sustained contribution.

Cold-start protocol: Initial reputation is distributed by the founding team to bootstrap participation. This founding allocation is time-bounded: it decays fully within 18 months regardless of activity, ensuring that governance transitions from founder-weighted to community-weighted. The founding team's privileged position is an explicit, time-limited trade-off – not a hidden power grab. This is documented in the DAO constitution and auditable on-chain.

High reputation unlocks: proposal submission rights, review assignments, higher voting weight on reputation-weighted votes.

3.6 Treasury and Token Model

The DAO token serves three functions:

1. **Governance:** voting weight (where token-weighted).

2. **Access:** membership requirement.

3. **Incentive:** compensation for contributions.

Token distribution is merit-based, not purchase-based. Initial distribution: founding team + early contributors (vested). Ongoing distribution: research contributions, governance participation, sandbox participation. No public sale. No speculation.

Treasury is multi-signature with time-locked governance: major expenditures (>5% of treasury) require extended voting period + community ratification.

3.7 Integration with UILL

DAO membership identity is rooted in UILL. Reputation is stored as UILL-linked attestations. Token balances link to UILL identifiers. This means: you cannot participate in the DAO without data sovereignty – your identity, reputation, and participation history are yours, not the DAO's.

---

4. The UILL: Personal Data Sovereignty Architecture

4.1 Purpose

The Unique Information Library Ledger (UILL) is the personal data infrastructure of sOS. It is not a database owned by the system – it is a distributed collection of personal data vaults, each owned and controlled by its individual. The UILL provides:

1. **Data ownership:** cryptographic proof of data provenance and control.

2. **Selective disclosure:** share specific data with specific parties under specific conditions.

3. **Consent management:** smart contracts governing data access, usage, and revocation.

4. **Interoperability:** standardized data formats enabling coordination across vaults.

5. **Portability:** individuals can move their UILL between providers without data loss.

4.2 Structural Design

The UILL vault is organized in four layers, each with specific functions:

Identity Root	DID, Public Key Infrastructure, Recovery Mechanisms	Cryptographic identity foundation
Personal Data	Capacities (labels), Needs (labels), Preferences (labels), History (traces), Assets (tokens), Relations (graph)	Six-category label structure from Google Takeout extraction
Consent & Access Control	Smart Contract Governance: who can access what, under what conditions, for how long, revocation mechanism, immutable audit trail	Programmable data sovereignty
Personal AI Interface	Read (context, capacities, history), Process (governance information), Write (preferences, decisions, votes). AI runs locally or user-chosen.	Individual-AI integration layer

4.3 Relationship to Solid and the Three-Dimensional Capture Model

The UILL is architecturally compatible with Solid (Berners-Lee). A Solid pod serves as the storage layer for a UILL vault. However, the UILL extends Solid in critical ways specified by the thesis (§15-16):

1. **Three-dimensional data capture:** The UILL ingests data from three dimensions – physical (GPS, mobility patterns), virtual (digital activity, social networks), and cognitive (emotional states, preferences). These are not separate databases; they form a unified tensor that is factorized to produce labels.

2. **Tensor factorization / backpropagation:** Attributes (X) from the three dimensions are related to labels (Y) via learned functions (β). The model uses backpropagation techniques (Rosenblatt, 1958; restricted Boltzmann machines, CNNs) to identify the schema of social concepts that define individual and social dynamics. The model is phenomenological: it improves over time as more actions are captured.

3. **Smart contract consent:** Beyond static ACLs, UILL consent is programmable – "share X with DAO Y for purpose Z, for N days, with audit trail." This consent is enforced through smart contracts.

4. **Stigmergic trace generation:** UILL data is structured to produce traces readable by the Social Machine – not just stored, but *active* in coordination.

5. **DAO integration:** UILL identity roots link directly to DAO membership and reputation – identity, data, and governance form a single coherent system.

4.4 Data Standards

The UILL requires standardized data formats for interoperability:

Labels (capacities, needs, context)	JSON-LD with sOS vocabulary	Machine-readable, linked data
Consent policies	ODRL (Open Digital Rights Language)	Standardized, machine-enforceable
Identity	W3C DID + Verifiable Credentials	Interoperable, decentralized
Traces (actions, decisions)	Activity Streams 2.0 extended	Standardized activity representation
AI model descriptors	Model cards + training provenance	Transparency requirements

4.5 Privacy and Security

- **Encryption at rest:** vault data encrypted with individual's keys.
- **Zero-knowledge proofs:** prove properties without revealing data. **Note:** ZK proofs over decaying multidimensional reputation are a research-grade problem (circuit design, trusted setup, revocation). The architecture specifies ZK capability as a target, not a current guarantee. Initial implementation will use selective disclosure of VCs (reveal only the necessary claim) as a pragmatic first step toward full ZK privacy.
- **Data minimization:** smart contracts request minimum data for each purpose.
- **Breach resistance:** even if a vault provider is compromised, encrypted data + distributed architecture limits blast radius.
- **Right to deletion:** sOS distinguishes two layers: (a) **Content deletion** via cryptographic shredding – deleting encryption keys renders personal data permanently unreadable. This satisfies the right to be forgotten for content. (b) **Metadata immutability** – on-chain records that data *existed* and traces *derived from it* are immutable by design. This satisfies auditability. The trade-off is explicit: you can delete your data; you cannot erase the fact that it once existed or that coordination decisions were informed by it. This position is documented for GDPR compliance review in WP9.

4.6 Key Management and Recovery

This is the #1 failure mode for self-sovereign identity systems and must be architected, not deferred.

Key generation: UILL identity roots use a HD (hierarchical deterministic) wallet structure. The root seed generates child keys for: identity authentication, encryption, consent signing, and DAO governance actions.

Social recovery: The individual designates 3–5 guardians (trusted contacts, institutional providers, or a combination). A threshold M-of-N (recommended: 3-of-5) of guardians can jointly authorize key recovery. Guardians never access the key – they authorize a recovery ceremony that generates a new key and transfers identity binding. The old key is cryptographically revoked.

Key compromise recovery: If a UILL key is compromised, the individual initiates a compromise-recovery flow: (a) guardians authorize identity re-binding to a new key, (b) the old key is added to a DAO-governed revocation registry, (c) any governance actions signed by the compromised key during the compromise window are flagged for review, (d) reputation VCs are re-issued to the new key (reputation is tied to the identity, not the key).

Custodial fallback: For users unwilling or unable to manage keys directly, the architecture supports delegated custody through regulated providers (analogous to a bank holding your assets). Custodial providers are subject to DAO-governed standards, audit requirements, and

liability bonds. This is not the recommended path, but it prevents key management from becoming an exclusion mechanism.

---

5. The Social Machine: Human-AI Coordination

5.1 Purpose

The Social Machine is the coordination layer of sOS. It is not a single program or platform – it is the architecture by which human agents, personal AIs, labeled traces, and shared information environments produce emergent governance coordination.

The Social Machine replaces centralized command-and-control with three mechanisms:

1. **Stigmergic traces:** every governance action leaves a trace in the shared environment.

2. **Label-based filtering:** traces are structured through a shared label ontology, making them queryable and actionable.

3. **Personal AI mediation:** each individual's AI reads relevant traces and presents actionable governance information.

5.2 The Stigmergic Loop

The stigmergic coordination mechanism operates through a shared information environment:

1. Shared Environment – Labeled Trace Registry: All agents write traces to a common registry. Examples: "resource tagged available", "policy Y affects community Z", "proposal 42 received 67% support", "streetlight #847 needs repair".

2. Agents read relevant traces: Each agent (human + personal AI) reads traces relevant to their context, capabilities, and current tasks.

3. Behavior modification: Agents modify their behavior based on trace information – tagging resources, voting, proposing, allocating, notifying, delegating.

4. New traces written: Modified behavior produces new traces, closing the feedback loop. The environment becomes the coordination medium – no central controller required.

Shared Environment	Labeled Trace Registry – append-only, queryable
Agents	Humans + Personal AI – read context-relevant traces
Actions	Tag, vote, propose, allocate, notify, delegate
Feedback	New traces → environment update → agents re-read → continuous loop

5.3 The Label Ontology

Labels are the shared vocabulary of the Social Machine. They must be:

- **Sufficiently expressive** to capture meaningful social categories (capacity, need, context, interdependence).
- **Sufficiently constrained** to prevent label spam and ontology capture.
- **Governed collectively** – the label ontology itself is a DAO-governed artifact, not controlled by any single party.

Initial label categories (from the thesis):

Capacities	"has_skill:data_analysis", "has_resource:meeting_room"	What agents can offer
Needs	"needs:childcare", "needs:legal_review"	What agents require
Contexts	"affected_by:policy_zoning_B24", "located_in:district_3"	Situational framing
Actions	"proposed:funding_request_42", "validated:task_17"	What agents have done
Interdependencies	"depends_on:resource_X", "impacts:community_Y"	Relational structure

5.4 Personal AI Architecture

Each personal AI agent:

1. **Reads** from the individual's UILL (capacities, needs, preferences, history).

2. **Scans** the shared trace environment for relevant information.

3. **Processes** governance information: filters noise, identifies interdependencies, simulates outcomes, surfaces trade-offs.

4. **Presents** actionable options to the human – not decisions, but decision support.

5. **Acts** on human instruction: votes, proposes, allocates, delegates, validates.

6. **Learns** from outcomes (with human feedback) to improve future processing.

Critical design constraint: the AI cannot act without human authorization. It is an augmentation tool, not an autonomous governor. This is enforced architecturally – write operations to the shared environment require cryptographic signatures from the human's UILL identity.

5.5 Debate Platform: Decision-Making Space

When stigmergic coordination is insufficient – when values genuinely conflict – the Social Machine includes a structured deliberation layer (thesis §14.5):

1. **Issue framing:** the conflict is formalized as a structured question with labeled context.

2. **Evidence aggregation:** relevant traces, data, and analysis are assembled.

3. **Deliberation period:** structured discussion with moderation and anti-manipulation mechanisms.

4. **Decision mechanism:** voting (see §3.4) informed by deliberation and evidence.

5. **Outcome trace:** the decision becomes a trace in the shared environment, influencing future coordination.

The debate platform is not an alternative to stigmergic coordination – it is the *escalation path* when emergence is insufficient.

5.6 Decision-Class Typology: When Emergence vs. When Vote

A critical architectural boundary: which coordination outcomes can emerge stigmergically, and which require formal DAO action? The architecture defines four decision classes:

C0 – Self-coordinating	No conflict, no resource contention	Pure stigmergy (traces only)	Tagging a resource as available; discovering and using it
C1 – Lightweight coordination	Minor resource allocation, no value conflict	Stigmergy + reputation-weighted signal	Prioritizing which streetlight to repair first
C2 – Governed allocation	Significant resources, potential disagreement	Debate platform → DAO vote	Allocating sandbox budget across research proposals
C3 – Constitutional	Changes to rules, architecture, or rights	Full DAO governance + extended voting + guardian review	Modifying the voting mechanism, label ontology, or treasury rules

Escalation trigger: any participant can escalate a C0/C1 trace to C2 debate by staking reputation. If the debate platform reaches a threshold of participants (quorum: 5% of active members or 15 participants, whichever is lower), it triggers a formal DAO vote. This prevents both over-formalization (not everything needs a vote) and under-governance (contentious issues can't be buried in traces).

5.7 Off-Chain Execution and Verification

Most real governance is off-chain: hiring, physical resource allocation, institutional decisions. sOS must handle this without breaking accountability.

Protocol:

1. A DAO vote authorizes an off-chain action (e.g., "allocate €5,000 to research project X").

2. A designated executor (individual, team, or institutional partner) is assigned with a time-bound mandate.

3. The executor publishes execution evidence (receipts, deliverables, attestations) as signed traces in the registry.

4. A verification pool (randomly assigned reputation-weighted reviewers) validates the evidence within a defined window.

5. If validation passes: the action is marked complete; the executor's reputation is updated. If validation fails: dispute resolution is triggered (see §5.8 below).

This protocol does not eliminate trust – it makes trust *auditable, time-bounded, and revocable*.

---

6. Integration: From Components to System

6.1 How the Three Components Connect

The sOS architecture integrates three core modules (thesis §14-17) on top of shared infrastructure:

DAO	Governance	↔ UILL (identity binding via VC)
UILL	Data sovereignty	↔ Social Machine (trace generation)
Social Machine	Coordination engine	↔ DAO (coordination outcomes → binding decisions)

Shared Infrastructure: DLT (blockchain/DAG for smart contracts), Trace Registry (stigmergic environment), Label Ontology (governed vocabulary), Identity Layer (DID + Verifiable Credentials), Storage Layer (distributed, encrypted).

6.2 Data Flows

Identity binding	UILL	DAO	DID links to membership	VC issuance
Reputation	DAO	UILL	Earned rep → attestation	VC issuance
Governance action	DAO	Social Machine	Vote, proposal → trace	Event emission
Trace generation	Social Machine	Shared Env	Action → labeled trace	Registry write
Trace consumption	Shared Env	UILL (via Personal AI)	Relevant traces → individual	Query + filter
Consent	UILL	Any	Data access permission	Smart contract
Coordination outcome	Social Machine	DAO	Emergent decision → binding	DAO vote or smart contract execution

6.3 Trust Model

- **Trust in identity:** cryptographic (DID + PKI), not organizational.
- **Trust in data provenance:** UILL + trace immutability provides auditable chain of custody.
- **Trust in rules:** smart contracts are open-source, auditable, and upgradeable only through DAO governance.
- **Trust in AI:** personal AI serves the individual; model integrity is attested (§7.7); diverse models prevent population-level homogenization; AI behavior is auditable.
- **Trust in coordination outcomes:** emergent from transparent traces + auditable rules; legitimacy derives from process, not authority.

---

7. Technical Dependencies and Choices

7.1 Blockchain / DLT

sOS requires a DLT for: smart contracts (DAO governance, UILL consent, token), immutable audit trails, and decentralized identity anchoring. Architecture requirements:

Smart contract support	DAO governance, UILL consent, token logic
Low transaction costs	Must not exclude participants by gas fees
Decentralization	No single entity can censor or control
Upgradeability	Contracts must be upgradeable through governance
Interoperability	Must connect to identity standards (DID, VC)
Energy efficiency	Proof-of-Stake or equivalent; PoW unacceptable

Current candidates (to be evaluated in sandbox): Ethereum L2 (Optimism, Arbitrum), Polygon, Tezos, or a dedicated app-chain. The choice is an implementation decision, not an architectural commitment.

7.2 Identity Standards

- **W3C DID** (Decentralized Identifiers) for UILL identity roots.
- **W3C Verifiable Credentials** for reputation attestations, membership proofs, and label assertions.
- **DIDComm** for secure peer-to-peer communication between agents.

7.3 Smart Contract Languages

sOS smart contracts must be:

- Formally verifiable (security-critical: treasury, consent, voting).
- Upgradeable through DAO governance.
- Auditable by third parties.

Candidates: Solidity (EVM ecosystem, largest auditor pool), Michelson (Tezos, formal verification), or Move (resource-oriented, strong safety). Decision belongs to sandbox phase.

7.4 AI Models

sOS does not build its own foundation models. It integrates existing models under strict constraints:

- **Diversity requirement:** the system must support multiple model providers and architectures to prevent single-model capture.
- **Local-first preference:** where technically feasible, personal AI should run locally or on user-chosen infrastructure.
- **Transparency:** model cards, training data provenance, and behavior audits are required for any model integrated into sOS.
- **Small model viability:** the architecture should work with models small enough for personal devices (7B–70B parameter range) rather than requiring centralized supercomputing.

7.5 Storage

- **UILL vaults:** encrypted personal storage (Solid-compatible pods, IPFS, or personal servers).
- **Trace registry:** append-only, content-addressed (IPFS or comparable), indexed by label and context.
- **Label ontology:** version-controlled, DAO-governed, published as linked data.

7.6 Trace Query and Index Layer

This component was a hidden load-bearing dependency in the original draft. It is now specified as a first-class architectural component.

The trace registry (§7.5) is content-addressed storage – it has no native query layer. Yet Personal AI agents must "scan the shared trace environment for relevant information" (§5.4). Connecting these requires a **trace query and index layer** with the following properties:

- **Index by:** label, context, timestamp, source reputation, decision class (C0–C3).
- **Query types:** real-time feed (new traces matching filters), search (historical traces by criteria), aggregate (counts, trends, reputation-weighted signals).
- **Trust model:** the index layer is a *trusted service* – it can omit, reorder, or delay traces. Mitigations: (a) multiple independent index providers can be queried, (b) index results are cross-checked against the append-only registry for completeness, (c) the index layer is open-source and DAO-governed, (d) reputation-gated index providers with slashing for provable omission.
- **Latency target:** new traces should be queryable within 5 seconds of publication for C0/C1 coordination; 60 seconds for C2/C3 governance. This is a Phase 1 validation target, not a Phase 0 guarantee.

7.7 AI Model Integrity Attestation

Personal AI models integrated into sOS must be attested for integrity:

- **Model cards** (standardized metadata: architecture, training data, evaluation results, known biases).
- **Training provenance** (data sources, fine-tuning process, alignment methodology).
- **Behavioral audit** (third-party evaluation on governance-relevant tasks: bias detection, information filtering transparency, decision-support accuracy).
- **Open-source preference:** closed-source models are permitted but subject to additional audit requirements and user warnings. Open-weight models that can be independently verified are the recommended default.

The DAO maintains a **model registry** listing attested models. Personal AI must use models from this registry. Models can be challenged and delisted through DAO governance.

---

8. Threat Model

This section consolidates attack surfaces that any implementation must defend against. It is a minimum viable threat model for Phase 0.

8.1 Sybil Attacks

Attack: One actor creates many fake identities to manipulate reputation-weighted or conviction voting, dilute governance, or extract resources.

Defense: sOS does not claim to solve Sybil resistance. It adopts defense-in-depth:

- Reputation-weighted voting degrades gracefully: fake identities start with zero reputation; earning enough to impact governance requires sustained activity, which is itself Sybil-deterring.
- Quadratic voting is explicitly excluded until proof-of-personhood is solved (§3.4 note).
- Phase 1 sandbox will evaluate proof-of-personhood mechanisms with Gitcoin Passport as the initial candidate for low-stakes contexts.
- For high-stakes governance (C2/C3), additional verification (video call, institutional attestation) may be required.
- Acknowledged limitation: this is the hardest unsolved problem in decentralized governance.

8.2 Reputation Collusion

Attack: Groups of participants mutually inflate each other's reputation through reciprocal validation, peer-review rings, or coordinated voting.

Defense:

- Reviewer assignment is randomized and anonymous during review.
- Reputation-gated write operations limit spam accounts.
- Statistical anomaly detection on reputation graphs (clique detection, reciprocity outliers).
- Reputation decay ensures collusion gains are perishable.

8.3 Trace Poisoning

Attack: Bad actors flood the trace registry with false labels to manipulate emergent coordination.

Defense:

- All traces are cryptographically signed by a UILL identity.
- Traces can be challenged; challenged traces are deprioritized pending verification.
- Reputation-gated trace amplification: traces from low-reputation sources get lower visibility.
- Rate-limiting on trace publication per identity.

8.4 Oracle / Real-World Boundary

Attack: Physical-world claims ("streetlight broken," "resource delivered") are false, but the system treats them as true.

Defense:

- Physical claims require multi-party attestation (≥2 independent validators).
- Executors stake reputation on truthfulness – false claims cost reputation.
- Random verification audits with economic penalties for false reports.
- Acknowledged limitation: this is the hardest engineering problem in decentralized governance.

8.5 AI Perception Control

Attack: A compromised or biased personal AI filters governance information to manipulate its human's decisions.

Defense:

- AI model integrity attestation (§7.7) requires model cards, training provenance, and behavioral audits.
- User-visible transparency: the AI must show *why* it surfaced certain traces and filtered others, with override capability.
- Diversity is a systemic defense (prevents population-level homogenization), not an individual defense – this is explicitly stated.

8.6 Delegation Capture

Attack: Governance power re-concentrates as participants delegate votes to a small number of influential delegates.

Defense:

- Delegation is time-bounded (max 6 months, renewable).
- Delegation is revocable at any time by the delegator.
- Delegates' voting records are public and auditable.
- Reputation-weighted limits on delegations per delegate (max 5% of total voting weight).

8.7 Smart Contract Upgrade Capture

Attack: Attacker captures DAO governance and upgrades smart contracts to drain treasury or modify consent rules.

Defense:

- All contract upgrades require: (a) extended voting period (≥14 days), (b) supermajority (≥66%), (c) mandatory security audit published before vote, (d) time-lock between approval and execution (≥7 days) allowing guardian veto.
- Guardian council (elected by DAO, rotating) can veto upgrades within the time-lock window.
- Critical contracts (treasury, consent, identity binding) have higher upgrade thresholds than operational contracts.

---

9. Positioning Against Prior Art

sOS is not built in a vacuum. This section positions the architecture against established systems and identifies what is genuinely novel.

9.1 Comparison Matrix

Governance model	Reputation + task-based	QV + grants	Token voting	N/A (data layer)	Reputation + conviction + stigmergy
Identity	Ethereum address	Passport (PoP)	Ethereum address	WebID + Pods	DID + UILL (data-sovereign identity)
Reputation	Domain-specific, earned	Trust bonus	Token-based	N/A	Multidimensional, decaying, UILL-rooted
Data sovereignty	No	No	No	Yes (pods)	Yes (UILL) + smart contract consent
Coordination	Task marketplace	Grant rounds	Proposals + votes	App interoperability	Stigmergic traces + personal AI
AI integration	No	No	No	No	Personal AI agents (core)
Sybil resistance	Reputation cost	Passport (PoP)	Token cost	N/A	Reputation cost + defense-in-depth

9.2 Key Prior Art

Colony: The reputation-based task governance model is the closest prior art for sOS's DAO. sOS extends Colony by: (a) rooting reputation in UILL (portable, not platform-bound), (b) adding time-decay with defined half-lives, (c) integrating with stigmergic coordination rather than task-claiming alone.

Gitcoin + Gitcoin Passport: The canonical real-world battleground for QV + Sybil. Gitcoin's experience – QV required building an entire proof-of-personhood layer (Passport) to be viable – is the empirical basis for sOS's decision to default to reputation-weighted voting rather than QV.

Ostrom's Commons Governance: Elinor Ostrom's design principles for governing common-pool resources are the intellectual ancestor of sOS's distributed coordination. Ostrom demonstrated that communities can self-govern without central authority or privatization *if* certain institutional design conditions (clear boundaries, collective choice, monitoring, graduated sanctions, conflict resolution) are met. sOS translates these principles into computational infrastructure: labels as clear boundaries, DAO as collective choice, trace registry as monitoring, reputation decay as graduated sanctions, debate platform as conflict resolution. Ostrom is not cited as a competitor but as a theoretical foundation.

Proof-of-Personhood Field: Worldcoin, BrightID, Proof of Humanity, Idena – the entire field is acknowledged as relevant prior art for the Sybil problem (§8.1, §11). sOS does not commit to any specific PoP mechanism; the sandbox will evaluate candidates with privacy trade-offs explicitly documented.

9.3 What Is Genuinely Novel in sOS

Three architectural claims are genuinely novel (not just recombined):

1. **UILL-rooted identity-data-governance fusion:** Identity, personal data sovereignty, and governance participation are the same system, not separate products. You cannot participate in governance without data sovereignty – this is a structural property, not a policy.

2. **Personal AI as governance mediator:** AI serves individuals, not institutions, and mediates between personal context (UILL) and collective coordination (Social Machine). This is not AI-as-oracle or AI-as-governor – it is AI-as-personal-augmentation within a governance architecture.

3. **Stigmergic coordination as first-class governance mechanism:** Traces, labels, and emergent coordination are the default path; formal voting is the escalation path. This inverts the usual DAO model where voting is the default and coordination is ad-hoc.

---

10. Implementation Pathway

> Phase 0 (Months 1–3): Foundation
>
> ├── Select DLT platform (sandbox evaluation)
>
> ├── Implement DID + VC identity layer (UILL root)
>
> ├── Deploy minimal DAO (proposal + voting + treasury)
>
> ├── Define initial label ontology (20–50 labels)

└── Deploy trace registry (append-only, queryable)

Phase 1 (Months 4–9): Sandbox Core

├── UILL personal vault prototype (Solid-compatible)

├── Smart contract consent templates (share, revoke, audit)

├── Stigmergic coordination loop (traces → query → action)

├── Personal AI prototype (single model, basic filtering)

├── Debate platform (structured deliberation)

└── Recruit 20–100 sandbox participants

Phase 2 (Months 10–18): MVP

├── UILL production: encryption, ZK proofs, data portability

├── Multi-model personal AI support (diversity enforcement)

├── Full governance lifecycle (proposal → review → vote → execute → audit)

├── Label ontology v2.0 (community-governed, >100 labels)

├── Smart contract upgradeability framework

└── Third-party security audit

Phase 3 (Months 19–36): Full Platform

├── dApp marketplace (third-party governance applications)

├── Cross-DAO interoperability

├── Advanced AI: outcome simulation, policy impact prediction

├── Scale to 1,000+ participants

└── Formal verification of core contracts

---

11. Open Technical Questions

This architecture intentionally leaves questions open for the sandbox phase:

1. **DLT selection.** Which platform best balances decentralization, cost, and smart contract expressiveness for a governance system?

2. **Scalability of the trace registry.** At what volume do stigmergic traces require sharding or sampling? What is the minimum query latency for coordination-relevant traces?

3. **Label ontology governance.** How does a community collectively govern its shared vocabulary without capture by the most active or powerful participants?

4. **AI diversity enforcement.** How do we verify that personal AIs across the system are genuinely diverse (different models, different providers) rather than all routing to the same API? See model integrity attestation (§7.7).

5. **UX for non-technical users.** How do we make cryptographic identity, smart contract consent, and AI-mediated governance accessible without compromising security?

6. **Sybil resistance.** How does the DAO prevent fake identities from diluting governance without requiring invasive identity verification? See threat model (§8.1) and proof-of-personhood candidates (§9.2).

7. **Economic sustainability.** Can the system fund its own infrastructure (DLT gas, storage, AI inference) without charging participation fees that create exclusion?

These questions define the sandbox research agenda (WP14).

---

12. Conclusion

The DAO, UILL, and Social Machine are not three separate projects. They are three components of a single governance architecture:

- **The DAO** governs the rules, resources, and research.
- **The UILL** gives individuals sovereign control over their data and identity.
- **The Social Machine** enables coordination to emerge from labeled traces and personal AI, without central command.

Together, they form an architecture for governance under complexity: distributed where distribution handles complexity better, centralized where efficiency requires it, transparent where trust must be earned, and personal where power must remain with individuals.

This paper is a specification, not an implementation. The sandbox will test it. The DAO will govern its evolution. The architecture will be revised in response to evidence – that is the scientific commitment of sOS.

Revised draft – Branch A (Pi/DeepSeek generative) + Branch B (Anthropic Opus 4.8 review) + Synthesis (Pi/DeepSeek arbitration). Revisions applied 2026-06-28: threat model (§8), prior art positioning (§9), QV/Sybil resolution (§3.4), reputation data model (§3.5), key recovery (§4.6), immutability/deletion resolution (§4.5), emergence-vs-vote boundary (§5.6), off-chain execution (§5.7), trace query/index layer (§7.6), AI integrity attestation (§7.7), trust model update (§6.3). Cross-references updated 2026-06-29 (WP9, WP10, WP14 now complete). Steward reviewed 2026-06-29.
