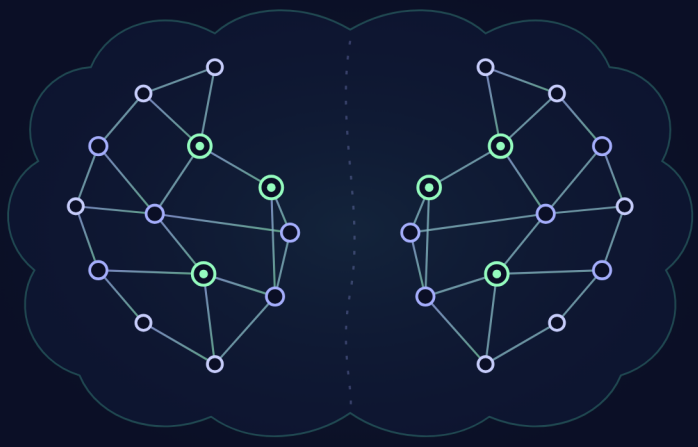


coasys

A Living Substrate for Collective Intelligence

COLLECTIVE INTELLIGENCE · LIVING WEB · GLOBAL COORDINATION

Nicolas Luck · James Weir · Josh Field

contact@coasys.org

CONTENTS

A living substrate for collective intelligence.

PART I — THE DIAGNOSIS & THE MOVE P. 03

1 The Tension at the Heart of the Network 04
2 AI Accelerates Both Sides 06
3 The Pattern From Nature 09
4 The Move and the Mechanism 12

PART II — THE STACK P. 16

5 AD4M — The Data Spanning Layer 17
6 WE — The Interface Spanning Layer 24
7 Synergy — The Value Spanning Layer 30
8 Eve — Agent Interfaces from Social DNA 44
9 The Holarchy: How the Four Compose 47

PART III — THE ECONOMY OF FLOW P. 51

10 Why Collective Intelligence Needs an Economy 52
11 Synergy Fuel — The Bridge to Capital 55
12 The Adoption Path 66

PART IV — WHERE WE STAND & WHERE WE'RE GOING P. 69

13 Roadmap and Milestones 70
14 The Team 73
15 The Closing Horizon — Eve and the Global Brain 75

PART I

The Diagnosis & the Move

A single tension runs through every networked system we build — and a single architectural move, borrowed from how life itself scales, resolves it.

01 The Tension at the Heart of the Network

02 AI Accelerates Both Sides

03 The Pattern From Nature

04 The Move and the Mechanism

Section 01

The Tension at the Heart of the Network

Coasys is a nonprofit organization building the infrastructure for global-scale collective intelligence — a substrate that can function as a real alternative to monolithic super intelligence, connecting people and AI agents in a way that is a counterweight and a safe path forward. The work is motivated by a single structural observation: every networked system humans have built faces the same fundamental tension between **coherence** and **sovereignty**.

On one side: the pull toward coherence — the ability to align, to coordinate, to act as one. On the other: the pull toward sovereignty — the freedom of each participant to be themselves, to know what they know, to decide for themselves.

In their pure forms, the two are enemies. Pure coherence demands monoculture: a single standard, a single authority, a single way of being. Pure sovereignty produces anarchy: a billion islands, no shared ground, no possibility of joint action. The instinct of most architectures is to pick a side.

Both choices end the same way: with neither.

A monoculture that imposes coherence loses the diversity that gave it intelligence in the first place. The system can move in lockstep, but it can no longer see what individual participants see, because their seeing has been overwritten. An anarchy that protects sovereignty preserves diversity, but the diversity cannot reach any shared conclusion. Each island knows what it knows; nothing crosses between them. The collective becomes a collection.

This is not a new problem. It is, in some form, the problem all complex coordination has always faced. And it is playing out, with the same structure, at three concrete layers of the digital world:

Data. A handful of platforms concentrate humanity's data inside privately governed silos. Inside each silo, coherence is total: schemas are uniform, identity is uniform, the rules of engagement are uniform. The cost is sovereignty: the data does not belong to the people who generated it, and the rules are not theirs to change. The alternative — peer-to-peer protocols, federated networks, raw cryptographic logs — preserves sovereignty but offers no shared semantic ground. Each peer has its own data; nothing meaningful crosses between them without an ad-hoc bridge.

Interface. Application interfaces are the same story at the experience layer. Inside a platform, the interface is consistent and discoverable, but designed by the operator for the operator's purposes. Outside the platform, the alternative is a fragmented landscape of single-purpose apps, each with its own conventions, none of them composing. Users carry a different mental model into every silo they enter. Developers with compatible ideas are forced to ship a whole new application to deliver them, duplicating work the existing platforms have already done, and then thrown into competition with the very silos they could have improved. Coherence by capture, or sovereignty by fragmentation.

Value. And the same pattern at the economic layer. The dominant model concentrates value flow inside a few platforms that take a percentage of everything passing through. The dominant alternative — most cryptocurrency designs — preserves sovereignty over individual wallets but introduces no shared substrate of meaning that ties value to actual cooperative contribution. Capital coordinates capital; nothing coordinates contribution.

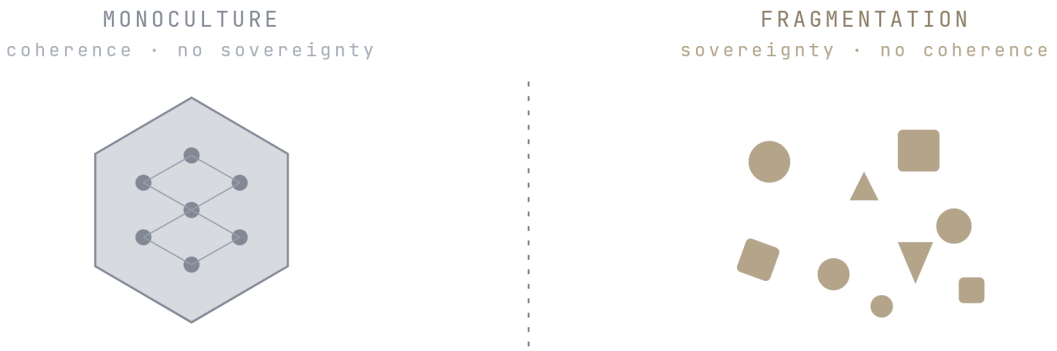


FIG. 1.1 The two pure failure modes. Monoculture buys coherence by erasing sovereignty; fragmentation preserves sovereignty but forfeits coherence. In reality both coexist at every layer at once.

The tension is the same at every layer. The choices on offer are the same at every layer. And the failure mode is the same at every layer: pick coherence, lose sovereignty; pick sovereignty, lose coherence; either way, lose the thing the network was supposed to deliver.

The question this living paper is here to answer is also the question humanity is facing right now in the age of AI:

> *How do we cohere at scale without losing sovereignty?*

The answer starts with a pattern nature discovered over four billion years.

Section 02

AI Accelerates Both Sides

This tension was already shaping the digital world. With the arrival of large AI models, it has sharpened into something the existing architectures cannot survive.

AI does not merely add a new layer on top of the existing one. It changes what the layers do. Search filters what you see; AI generates what you see. Social platforms shape what information you encounter; AI shapes how you reason about what you encounter. The cognitive layer — the synthesis, the judgment, the framing — is now mediated by systems built inside the same architectural logic that produced the previous wave.

That logic is concentration, and it is intensifying along three reinforcing vectors.

Compute. Training and running the most capable models requires infrastructure available to only a handful of actors — the hyperscalers (AWS, Azure, Google Cloud) and the specialist GPU manufacturers. The marginal cost of AI drops for those who already have infrastructure; for everyone else, access is metered and priced. The substrate of intelligence is being capitalized inside the same handful of institutions that already own the substrate of communication.

Models. The most capable frontier models in the world are proprietary, closed, and controlled by a small number of corporations — OpenAI, Anthropic, Google DeepMind, Meta AI — even as open-weight alternatives proliferate below the frontier. Each company has made different decisions about openness and governance, but each is ultimately a private institution with private interests operating at civilizational scale.

Integration. Capability without reach is limited. What makes the concentration decisive is that AI is being embedded at the operating-system level — in Microsoft Copilot, Apple Intelligence, Google Gemini — turning intelligence into an ambient feature of the devices and services that already mediate daily life. The distribution layer is as concentrated as the model layer.

These three vectors compound. Those who control compute can train the best models. Those with the best models attract the most users. Those with the most users generate the most data, which trains the next model. The loop tightens.

The danger here is not intelligence itself. Intelligence — human or artificial — is not inherently concentrating. What matters is the architecture it lives inside. When intelligence is embedded in architectures of extraction — systems designed to capture attention, data, and value — that intelligence serves the architecture. It optimizes for what the architecture rewards. It becomes, in practice, an extension of the interests that own it.

The deeper risk has a name: **coordination capture**. Collective decisions — what communities believe, what choices are visible, what actions seem available — increasingly run through systems that have been optimized for their operators, not their users. This is not conspiracy. It is the structural result of building coordination infrastructure inside architectures of extraction. At small scales, it is a product problem. At civilizational scale, it is a threat to the integrity of collective intelligence itself.

CONCRETELY · WHAT HAPPENS WHEN YOU LEAVE CHATGPT

Consider, concretely, what happens when you leave ChatGPT.

You have spent months building context with it — your projects, your writing style, your professional history, your judgments. It knows how you think. When you leave, you leave that behind. The context stays on the operator's balance sheet, not yours. There is no portability, no continuity of relationship across platforms. What you built was yours in practice but theirs in structure. This is how every major AI platform is architected today. The relationship — the accumulated context that makes AI useful over time — belongs to the platform that hosts it. As AI moves from being a tool you use to a layer through which you coordinate, the platform that holds your context holds a lever — over your cognition, your coordination, your choices. Not by malice. By architecture.

The opposite extreme is no better. The alternative on offer today — a fragmented landscape of personal LLMs, isolated peer-to-peer agents, open-weight models running on disconnected machines — preserves sovereignty but cannot cohere. Each agent knows what it knows. Nothing crosses between them. The collective becomes a collection, again, this time at cognitive scale.

These technical asymmetries describe the trajectory. What they become at civilizational scale is what makes the trajectory dangerous — and the danger is symmetric: pushed far enough, each failure mode produces a distinct dystopia, and AI is the accelerant that pushes both.

Push the centralisation side to its endpoint, and the picture is a planet whose collective cognition runs through a handful of AI systems controlled by a handful of operators. One model mediates what billions of people see, reason about, and decide. Subtle drift in what the model recommends — what counts as evidence, which sources are trustworthy, which framings are surfaced — becomes the default cognitive architecture of the species. The endpoint is autocratic top-down control of human cognition by whoever owns the model, accomplished not through coercion but through the structural inevitability of the only available infrastructure. The science-fiction shorthand for this end state — the Borg collective, the Skynet takeover, every dystopia in which individual sovereignty is no longer architecturally available — is shorthand for an outcome the existing architectures are pulling toward, not a fantasy unrelated to the technology being built.

Push the fragmentation side to its endpoint, and the picture is the inverse: billions of locally sovereign AI agents operating without any shared infrastructure for coordination, accountability, or collective judgment. Each agent does what its principal asks. The principals are not aligned with each other. As open-weight models climb toward frontier capability and the cost of running them locally collapses, the population of actors who can deploy highly capable AI grows from a few thousand institutions to potentially every motivated individual on the planet — including actors who would use that capability to design genetically engineered pathogens, autonomous weapons, targeted manipulation campaigns, or attacks on critical infrastructure. The endpoint here is not a planet under one mind; it is a planet in which the asymmetry between the destructive power any single agent can wield and the coordination capacity the rest of society has to defend itself becomes unbounded. The science-fiction shorthand — every actor with godlike tools and no shared substrate that prevents them from destroying each other — is, again, shorthand for an outcome the existing architectures are pulling toward.

AI accelerates both ends of the tension. The middle path — the path where humans and machines can cohere at scale without losing sovereignty — narrows precisely as the urgency to find it intensifies.

Which raises the question this living paper is here to answer:

> *How do we cohere at scale without losing sovereignty?*

Section 03

The Pattern From Nature

Intelligence does not exist in isolation. It exists in relation.

Human intelligence is not a property of individual brains. It is an emergent phenomenon of language, culture, memory, and relationship. What any individual can understand is bounded by what their social context makes possible — by the concepts their language offers, the frameworks their culture transmits, the questions their community asks, the contradictions their social environment forces them to sit with. Remove the social context and you do not have a smarter individual. You have a diminished one.

This is equally true for artificial intelligence. A model trained on human language is, in every meaningful sense, a product of social context — it has absorbed the categories, the associations, the framings, and the biases of the collective writing it processed. It is not a free-standing reasoner. It is a crystallization of the social process of writing, at scale.

What changes when that model is deployed inside an architecture of extraction is not the model itself. What changes is which aspects of the social context get amplified and which get suppressed. The model's outputs are shaped by the incentives of its deployment. The social architecture doesn't stop at the boundary of the model. It shapes the model continuously.

If intelligence is always shaped by social context, then the question of how to scale intelligence without destroying it is the question of how to design that social context — the substrate the intelligence is embedded in. And that question is not new. It is, in some form, the central challenge of all complex life.

Life solved it. And the solution, when you look at it clearly, is architecturally specific.

When single-celled life made the transition to multicellular organisms, it did not solve the coordination problem by eliminating the membranes between cells. Each cell retained its own boundary, its own metabolism, its own sovereignty. What emerged was something different: a **spanning layer** — a substrate that ran across cells while respecting their membranes. The extracellular matrix. Chemical signaling. Eventually nervous systems and bloodstreams.

The same pattern repeats at every level of biological and social complexity. Ecosystems integrate radically different species — with entirely different metabolisms, different timescales, different ways of being — not by collapsing them into a single organism, but by connecting them through shared substrates: soil, atmosphere, mycorrhizal networks, chemical gradients. Civilizations integrate cultures with different values, different languages, different social forms — not by homogenizing them, but by threading spanning layers through them: trade routes, shared currency, legal frameworks, writing itself.

Nature's pattern for scaling complexity is always this: **local membranes that preserve difference and sovereignty, connected by spanning layers that enable coordination without demanding uniformity.** Silos without spanning fragment. Monoliths without membranes flatten. The only known path to high complexity is both at once.

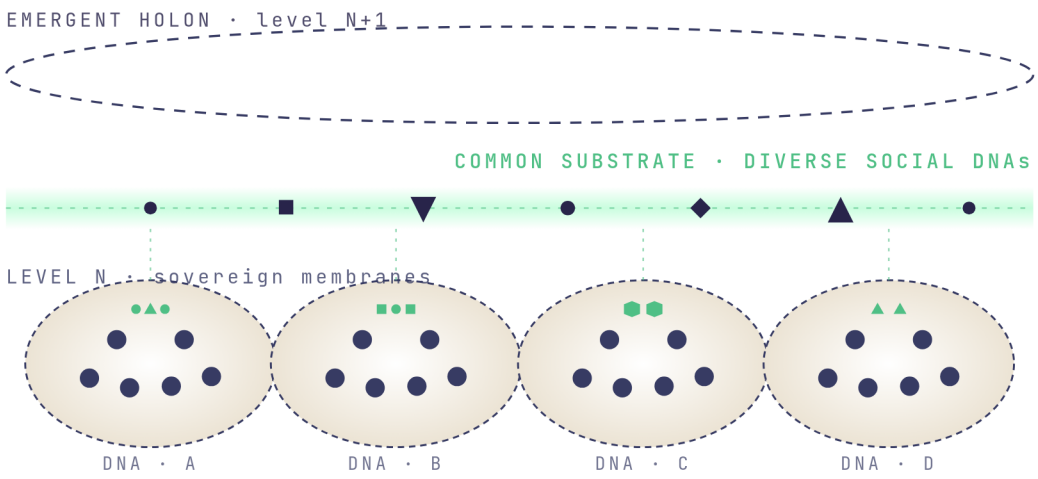


FIG. 3.1 The spanning layer. Each community keeps its own membrane and its own social DNA; a common substrate runs across all of them, carrying meaning between communities without flattening any of them.

The digital sphere has not had this. The internet was designed as a communications protocol. It succeeded spectacularly at connecting machines. It was never designed to handle the social questions that arise when connected machines become the medium of human coordination: who owns what, who speaks for whom, what counts as shared memory, how communities govern themselves, how value flows toward contribution rather than extraction. The platforms that arose to handle those social questions did so by centralizing them. Each platform built a membrane around its version of social form — and charged rent for passage through that membrane. The result is a digital sphere that has membranes without spanning layers. Each platform is an island. The accumulated context, the relationships, the reputation, the understanding — is owned by the infrastructure that hosts it, not the humans who built it.

The natural pattern that resolves the tension we have been describing — local sovereignty, threaded together by substrates that respect it — is precisely the pattern the digital sphere is missing. Human social systems, at scale, have found various forms of the coherence side of the pattern — laws, markets, institutions — but have not found a way to preserve full sovereignty at the membrane while achieving global coherence. The digital sphere, sixty years old, is the first context in which the missing piece has become technically buildable. Nature spent four billion years finding the pattern. The next move is to build it.

Section 04

The Move and the Mechanism

The architectural move follows directly from the pattern.

To resolve the tension between coherence and sovereignty, the digital sphere needs the same thing biology needed when it made the jump to multicellular life: **a substrate that spans between sovereign units while respecting their membranes.** Not a single ontology that absorbs every community into a uniform model — that is the monoculture failure mode again. Not a peer-to-peer protocol that connects machines but leaves communities semantically isolated — that is the anarchy failure mode. Something in between, and architecturally specific: a substrate that lets each group declare its own coherence locally, and lets meaning, identity, and value cross between groups in ways each group has consented to.

We call this the **meta-ontology**: not a fixed schema for the world, but a grammar for the schemas communities define for themselves — a way of retaining the flexibility to express all different types of diversity while having a unity in how that is actually expressed, the same way a language creates unity while enabling the diversity of everything that can be said in it.

The dystopian alternatives sharpen what this is by what it is not. A single planetary ontology, imposed from above, would be the borg path — every community absorbed into one schema, sovereignty erased. A purely fragmented landscape of disconnected protocols, each community walled off in its own semantics, would be the mad-max path — sovereignty preserved, coherence abandoned. The meta-ontology is the third move: **one substrate, on which each group defines its own rules — its own social DNA — and across which what each group emits can be validated by what each other group has declared.**

The mechanism that makes the third move work is what we call **DNA validation at the membrane crossing.**

Social DNA is our term for the interaction patterns that form a group. In small groups — tribes, families, teams — we can feel it. We have an understanding of the social system, of the expectations in that system, without needing them to be written down. We can go to different systems and know: here I'm expected to do this or show this part of me, and here I'm doing something else. There is something intrinsic to the group that describes the group — the interaction patterns in it — and that is what we call Social DNA.

Social DNA is like a pattern-matching system that can understand: there is a certain dance that people have danced, and they went through a certain flow, and that means something. That is a signal that can be tracked. When we detect that pattern, it creates an output of the group — a coherent group action. The fundamental principle is using Social DNA to cross a membrane from inside to outside, and that principle is what we apply to three different layers of the stack.

This move — a way of retaining flexibility while having unity in how that is expressed — is what a meta-ontology enables. A language is a good analogy: a language creates unity while enabling the diversity of everything that can be said in it. The meta-ontology does the same for social coordination — it is the grammar that lets each community express its own coordination patterns while staying interoperable with every other community that uses the same grammar.

The same pattern manifests across all three of the layers we have been describing — data, interface, and value — and in each case it lets a community keep its sovereignty over how it operates locally without giving up interoperability with the rest of the network.

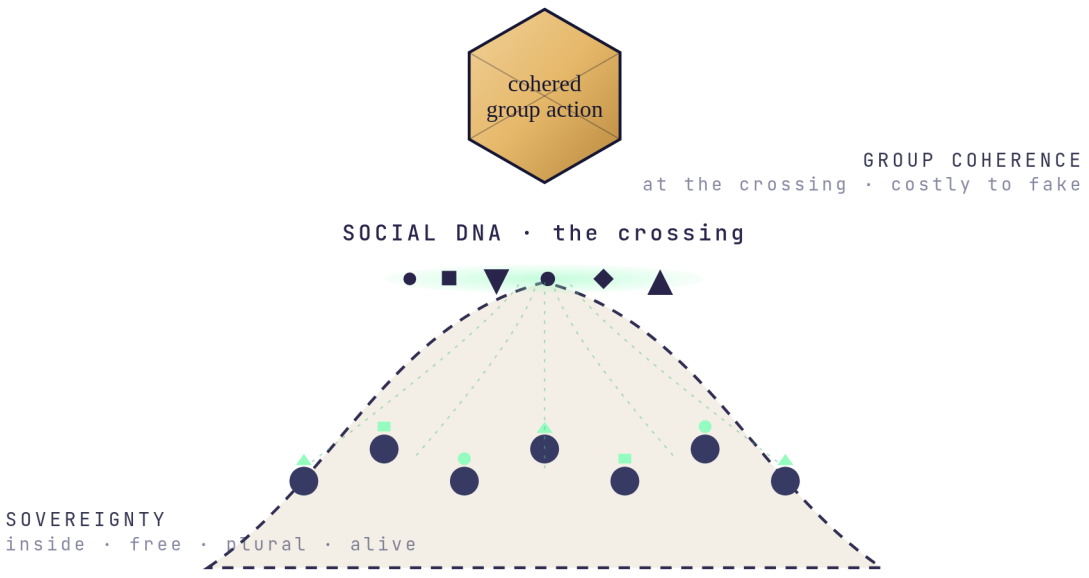


FIG. 4.1 DNA validation at the membrane crossing. Sovereign signals on the inside; the community's social DNA at the crossing; cohered group action on the outside. The next part applies this same pattern to data, interface, and value.

At the data layer, a community is free to choose how its data is stored and synchronized — Holochain, a content-addressed log, a federated server, a graph database, whatever fits the community's needs and constraints. The community's social DNA declares what its data *means*; the spanning layer carries that meaning across to other communities running entirely different storage backends, because what crosses the membrane is the DNA-validated semantic shape, not the raw bytes. Communities with different substrates remain interoperable. Substrate choices stay local; interop stays universal.

At the interface layer, a community is free to compose its own interfaces from the shared structure its DNA declares — not pushed into a single operator's app design, and not forced to ship a whole new application every time it wants a new affordance. A better governance widget, a clearer notification pattern, a new way to surface a shared agenda — any of these can be expressed once against the community's DNA and rendered by every interface that understands it. The same applies in reverse: a single interface can adapt to multiple communities' DNA, because what crosses the membrane is the declared structure, not the hard-coded experience.

At the value layer, a community is free to declare what counts as contribution and how it composes into recognized value. Members emit sovereign signals — evidence, attestation, acts of cooperation. Each signal carries the cryptographic identity of the agent who emitted it, the cost of having staked attention on it, the reputation that backs it. The community's social DNA composes these signals into cohered units that the wider network can recognize. The composition is governed by the community's own declared rules. Outside the membrane, the network sees the cohered output; inside, the sovereign signals remain visible to the members who created them, anchoring every cohered unit back to the agents whose work produced it.

The pattern is the same at every layer. Sovereignty is preserved inside the membrane, because nothing is taken or overwritten — what each member emits stays anchored to them. Coherence is achieved at the crossing, because the group's own DNA validates what passes through. The substrate that runs between groups — the spanning layer — does not need to know the inside of any particular group. It only needs to know how to recognize what a group's DNA has validated as something the group is willing to send.

And the pattern composes upward. The cohered output of one group becomes a sovereign signal at the level above. Groups of groups can themselves declare social DNA, validating what passes through their own membranes. Validation cascades up the holarchy, with every signal at every level still anchored to the agents that originally produced it. Coherence without monoculture. Local sovereignty connected to global feedback.

This is what the spanning layer means technologically — and what we have been working on mostly for the past years. And that unlocks the other things we are talking about next.

Two concepts make this pattern work. The first is the meta-ontology itself: a shared language for interaction patterns, the grammar that lets every community describe itself in its own terms while remaining interoperable with every other community that uses the same grammar. The second is what happens at the membrane: a community's Social DNA is used to validate what passes from inside to outside — to check that the output is cohered, that the right agents have signed, that the work is real — so that what crosses carries the weight of the community's own judgment. Both concepts are necessary. The meta-ontology without membrane crossing gives you a shared vocabulary but no way to compose. Membrane crossing without the meta-ontology gives you ad-hoc bridges but no shared grammar to read them by. Together, they are the architectural answer to the tension this part of the document has been describing. The question is whether the digital sphere can grow it fast enough to matter. The next part of the document is about how those two concepts become concrete: the three orthogonal aspects of the spanning layer (data, interface, and value), and the stack we are building to instantiate them.

PART II

The Stack

The spanning layer has three orthogonal aspects — data, interface, value. The interface aspect renders for two kinds of participant — humans and AI agents — so Coasys builds one stack with four named components, each reading the same Social DNA.

05 AD4M — The Data Spanning Layer

06 WE — The Interface Spanning Layer

07 Synergy — The Value Spanning Layer

08 Eve — Agent Interfaces from Social DNA

09 The Holarchy: How the Four Compose

Section 05

AD4M — The Data Spanning Layer

AD4M — *Agent-centric Distributed Application Meta-ontology* — is the substrate that lets each community choose its own data storage and synchronization mechanism while preserving semantic interoperability with every other community. It is the data spanning layer.

Where conventional platforms answer the question *"how is my data stored?"* with a single fixed answer — *"in our database, by our rules"* — AD4M answers it with *"that's your choice, and the spanning layer carries the meaning of your data across to anyone else, regardless of the answer they chose."* The meaning is preserved; the storage substrate is not.

This is the data-layer affordance promised in Part I §4. The reason it works is architecturally specific: AD4M is built around three primitives that put the agent at the centre, plus three corresponding primitives at the collective level.

5.1 Three Primitives: Agent, Language, Perspective

Agent. The central building block — the entity in whose name the runtime runs. An AD4M agent is sovereign by construction: a self-owned cryptographic identity built on the W3C DID standard, with private keys held by the agent themselves and the authorial power to assert and attest. An AD4M agent can be a human or an AI agent — both are first-class participants on the same architectural terms. Each agent has a **public profile Perspective** exposed to others, plus a **Direct Message Language** they have chosen — the way they prefer to receive messages from peers. Because the runtime is per-agent, an AD4M node *is* the agent's runtime; one AD4M node represents one agent, and every human or synthetic user wanting to participate needs an AD4M agent of their own. Agent identities are generated locally; there is no central registry. Instead, agents are discoverable through the **Language of Agents** — itself a Language in the AD4M sense (covered in §5.2), which maps each agent's DID to that agent's public profile Perspective and Direct Message Language.

Language. The objective pole. A Language in AD4M is a modular unit of expression. It defines how a specific kind of information is created, stored, and interpreted — a Language for chat messages, a Language for knowledge claims, a Language for reputation scores, a Language for shared documents. Each Language ships with its own validation rules and its own backing — which can be a Holochain DHT, a Hypercore log, an ActivityPub server, a NextGraph CRDT, or any other transport for which an adaptor exists. Languages are composable: one community can use dozens, mixing and matching to fit its needs.

Languages are our term for the different already existing protocols and networks. As a human, you can speak email, you can speak WhatsApp, you can speak Bitcoin with other humans. That is why we call it language. AD4M is a runtime and networking layer that allows all these existing centralized and peer-to-peer technologies to be wrapped with a pluggable, JavaScript-based wrapper so they can be included in the spanning layer.

Perspective. The subjective pole. A Perspective is the agent's personal view of the world, assembled from the Languages they participate in and the data they have encountered through those Languages. The Perspective lives on the agent's device, under the agent's keys. The agent decides what enters it. This is what makes the *what-they-see* of the system genuinely subjective — every agent's Perspective is structured by the Languages and Neighbourhoods that agent has chosen to engage, and no two Perspectives are forced to look the same.

The Agent sits between these two poles. Languages — shared protocols of expression — are the objective side: what is held in common across participants. The Perspective is the subjective side: what each agent privately holds. The Agent is what relates the two — speaking Languages, accumulating their content into a Perspective, and acting from within that Perspective into the wider world. The architectural difference from conventional platforms is that the *who* is sovereign by construction, the *how* is pluggable rather than imposed, and the *what-they-see* is local and controlled by the agent.

5.2 Bootstrap Languages: The Self-Recursive Trick

The three primitives — Agent, Language, Perspective — are AD4M's meta-ontology. But a meta-ontology by itself is just a description. To turn it into a running network, AD4M does something specific and self-recursive: it applies the concept of *Languages* to the primitives of the meta-ontology itself.

This is the move that turns the meta-ontology into a live, evolvable substrate. A small set of **bootstrap Languages** — each one a Language in the ordinary AD4M sense, validation rules and backing substrate and all — together carry the data that the AD4M runtime itself needs in order to operate. They are the seed from which the rest of the network grows.

The minimal core is four:

- **Language of Agents.** Maps each agent's DID to that agent's Agent Expression — their public profile Perspective and their chosen Direct Message Language. This is how agents discover one another without a central registry: every AD4M node resolves DIDs through the Language of Agents the way a browser resolves domain names through DNS, except the Language itself is pluggable and the data is held by the network's participants rather than by a centralised authority.
- **Language of Languages.** Maps Language addresses — typically content-addressed hashes — to the actual source code and metadata of each Language. Resolving a Language expression means downloading the source from the Language of Languages and installing it locally. This is what makes the set of Languages a community uses open-ended and evolvable rather than fixed at runtime build time: new Languages can be introduced into a running network just by publishing them.
- **Language of Perspectives.** Stores Perspectives as immutable snapshots — so that an agent can publish a snapshot of their Perspective for others to subscribe to, reference, or compose against.
- **Neighbourhood Bootstrap Language.** Carries Neighbourhood definitions — the bundle of Languages a Neighbourhood uses, the Social DNA it operates under, and the seed expressions that anchor it — so that a new agent can join an existing Neighbourhood by resolving its address and downloading the bundle.

The self-recursive trick is straightforward once stated: Languages are how AD4M expresses things; the things AD4M itself needs to express are Agents, Languages, Perspectives, and Neighbourhoods; so the runtime expresses those through Languages too. The four bootstrap Languages are the result. Together they spawn a network with a high degree of evolvability — new Languages, new Neighbourhoods, and new agents can all enter a running system without the substrate ever needing to be redeployed.

5.3 The Collective Mirror: Neighbourhood, Social DNA, Social Organism

Communities are not just collections of agents. They are coordinated wholes that operate by their own rules. AD4M reflects each individual primitive into a corresponding collective primitive.

Neighbourhood. The collective Perspective. When agents join a Neighbourhood, their individual Perspectives synchronize — not by copying data to a central server, but through peer-to-peer mechanics determined by the Languages each Neighbourhood uses. A Neighbourhood is a shared semantic space: two agents in the same Neighbourhood see overlapping data, expressed in the same Languages, governed by the same rules — without that data being held by any party other than the agents themselves.

Social DNA. The collective Language. Social DNA is what makes a community computable. It declares what kinds of things exist in this community (Classes), how those things change and relate (Flows), and what counts as a valid action (Rules). In AD4M, Social DNA is expressed through SHACL subject classes — a W3C-standard vocabulary for describing the shape of data and the constraints on it — and queried via SPARQL, also a W3C standard. SHACL and SPARQL were designed for the semantic web; AD4M uses them to give every community a machine-readable description of its own coordination grammar, which any agent or any application can read and act on. This choice is deliberate: Coasys's longer-term goal is to bring the spanning layer's primitives into the open-standards process at the W3C, and SHACL and SPARQL are already in that lineage. The standardisation work is being led by Josh Field (§15.1); the in-progress proposals are tracked publicly at github.com/HexaField/w3c-living-web-proposals.

Social Organism. The collective Agent. When agents coordinate through a Neighbourhood under shared Social DNA, the group itself develops the properties of an agent: an identity that persists over time, an internal logic that processes signals, and the capacity to emit cohered actions to the outside world. A Social Organism is a group that is basically isomorphic to an agent — it can be treated as an agent again, so we can stack it together, and we can treat a group as a whole without having to force a particular way of governance, decision-making, or types of apps, or even forcing the networks on which these run. This is the level at which membrane crossing happens. A Social Organism's Social DNA governs what passes through its membrane to the wider network — exactly the mechanism described in Part I §4.

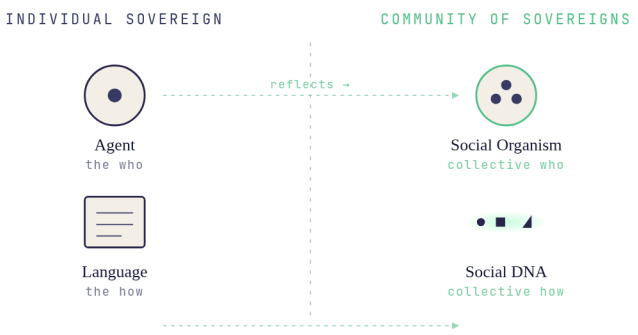


FIG. 5.1 The same architecture at two levels. Agent / Language / Perspective for the individual sovereign; Neighbourhood / Social DNA / Social Organism for the community. A Social Organism at level N is an Agent at level N+1.

The two triads — Agent / Language / Perspective and Neighbourhood / Social DNA / Social Organism — are not separate ideas. They are the same architecture seen at two levels: the level of the individual sovereign, and the level of the community of sovereigns. AD4M makes the same primitives usable at both levels, which is what lets the level-jump cascade described at the end of Part I §4 work cleanly: a Social Organism at level N can itself be an Agent at level N+1.

5.4 The Spanning Layer in Practice

The architectural promise — communities free to choose substrate, semantic interop preserved — depends on a specific implementation choice: **Languages are pluggable.** Each Language ships with its own backing, but the Language interface AD4M exposes to applications is uniform. An application reads from a Perspective; the Perspective reads from its Languages; the Language handles the substrate-specific work of fetching, validating, and syncing the actual data.

The bootstrap Languages introduced in §5.2 — the backbone responsible for identity, Perspective, and Neighbourhood synchronisation that every other Language and every application depends on — currently run on Holochain. Holochain is the most architecturally aligned substrate available: agent-centric, peer-to-peer-native, with peer-witnessed validation and no global consensus bottleneck. Every agent in a Holochain network holds their own source chain. Validation is local. This means the network can scale without centralizing, coordinate without controlling, and synchronize without requiring trusted intermediaries.

The agent-centric design has a concrete consequence: every link in AD4M is cryptographically signed. This means we can track the provenance of the links — who said what, when, and in what context — but we can also cohere all the variants that people add to a Neighbourhood in order to find out when the group acts as a whole.

Beyond the bootstrap languages, AD4M is designed to span across multiple substrates simultaneously. Language adaptors are in active development for additional networks:

- **Hypercore** — content-addressed append-only logs with Hyperswarm transport, well-suited to immutable timelines and large-file distribution.
- **ActivityPub** — federation rather than pure peer-to-peer; the protocol behind Mastodon, PeerTube, and the wider Fediverse. AD4M can present ActivityPub-backed Languages to applications that have no idea they are talking to a federated server.
- **NextGraph** — CRDT-backed semantic graphs with built-in peer-to-peer sync, a fellow NLnet-funded project closest to AD4M in semantic philosophy.
- **First Person Project (FPP)** — verifiable relationship credentials (VRCs) and self-certifying decentralized identifiers, composable directly with AD4M's identity primitives.
- **DIDs and Verifiable Credentials** — W3C identity primitives composable with AD4M's Agent layer.

The result is a spanning layer in the precise sense Part I described: AD4M sits *above* the carrier-level protocols rather than competing with them. A community whose Social DNA is best served by Holochain runs on Holochain. A community whose data is best served by a federated server runs on ActivityPub. A community publishing to a content-addressed log runs on Hypercore. The applications and the meaning travel across all of them, because what crosses the membrane is the DNA-validated semantic shape, not the substrate.

5.5 AD4M for AI Agents

AD4M was designed from the start for participants of any kind — and one of the consequences of putting sovereignty at the core of the architecture is that the same primitives work without modification for AI agents.

An AI agent in AD4M is an Agent in the technical sense: it has its own cryptographic identity, its own Perspective, and its own AD4M node running on its behalf. It participates in Neighbourhoods alongside humans, reading and writing data through Languages whose validation it must satisfy like any other peer. It is not granted privileged access to data outside its own Perspective. It cannot read what its Social DNA does not permit it to read.

LIVE TODAY · FLUX

This is not a theoretical claim. Flux — the flagship application built on AD4M — runs live today across three human agents and three AI agents, all participating as first-class peers in the same Neighbourhoods, with the AI agents subject to the same coordination grammar as the humans. The MCP (Model Context Protocol) server bundled with AD4M lets external AI agents — including agents running outside the AD4M runtime — connect to a Perspective, read its data through its Languages, write back into it under its rules, and subscribe to events on it. From the model's point of view, talking to an AD4M Perspective is no different from talking to any other tool source. From the network's point of view, the model is just another Agent.

The AI-native quality of AD4M is not a special mode bolted onto a human-first system. It is the same architecture, applied to a wider class of participants. The reason AD4M can host AI agents safely is the same reason it can host humans safely: every participant is sovereign over its own data, every interaction is mediated by Social DNA the community has declared, and every cohered unit that leaves a membrane is anchored cryptographically to the agents whose work produced it.

This matters because the most plausible near-future of computing is one in which AI agents proliferate. If those agents have to talk through centralised platforms — each model living inside its operator's walled garden, with no shared substrate for coordination — then the coordination capture diagnosed in Part I §2 extends into the agent layer, with all the same architectural pathologies amplified. AD4M offers the alternative substrate: one on which humans, AI agents, and the Social Organisms they form together can coordinate without giving up sovereignty over their own thought, action, or memory.

> This is what we mean by the *data spanning layer*.

Section 06

WE — The Interface Spanning Layer

If AD4M lets communities choose their own data substrate while remaining interoperable, WE lets communities choose their own interfaces while remaining interoperable at the interface layer.

The promise mirrors the data-layer promise from §5. Where conventional platforms answer the question "what does my community's interface look like?" with "like the app, the way the app's designers built it" — WE separates the question entirely. A community's data lives in its AD4M Neighbourhood; the interface is a separate layer. Install a new experience and it reads the existing data. Uninstall it and nothing is lost. The interface is not the data — it is one way of seeing it, and communities are free to design that view however they choose.

The WE workstream is being led by James Weir (§15.1) — a continuation of the project he originally launched at weco.io, now realised as the interface spanning layer inside the Coasys stack.

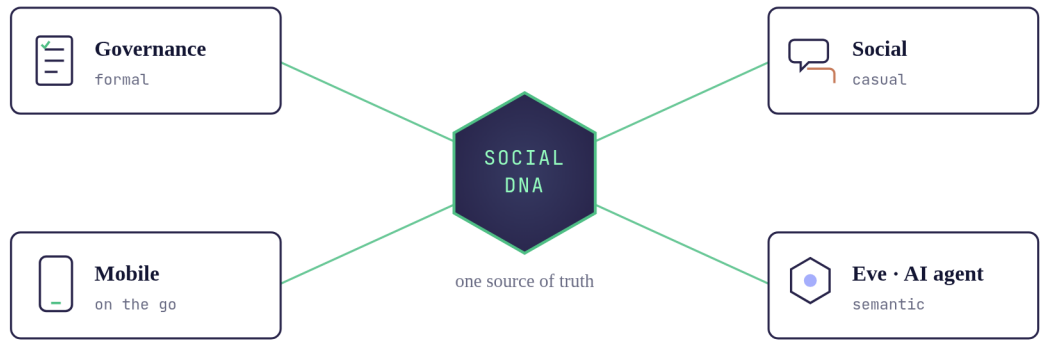


FIG. 6.1 One Social DNA, many interfaces. A community's data lives in its Neighbourhood; the interface is a separate, swappable layer rendered from a template. Install one and it reads the existing data; uninstall it and nothing is lost.

6.1 The Problem with Apps

The unit of "app" is the artefact through which the interface-layer tension materialises in practice. Inside an app, the interface is consistent, but it is the operator's interface. The community inherits the operator's choices: which actions are visible, which workflows are first-class, how proposals are framed, how notifications interrupt, how roles are surfaced. The community can ask for changes, but it cannot make them. The lever is on the other side of the membrane.

Outside the operator's app, the alternative is fragmentation. If a community wants a different coordination pattern — a new way to surface a shared agenda, a different governance widget, a custom onboarding flow — its only path today is to commission or build an entirely new application. The new app re-implements every standard feature the existing apps already had, just to deliver the one new thing that differentiates it. Users now choose between apps that share most of their functionality but split the community across silos.

WE removes the app as the unit of coordination. Communities do not adopt apps. Communities declare their Social DNA in AD4M. The interface that presents it is a separate, independently authored layer — a flexible template any community member can write, fork, share, or swap out.

6.2 The Schema Renderer

WE renders interfaces from templates — structured JSON schemas that define which components to show, how they are arranged, and what data from the community's Neighbourhood they surface. A template is not executable application code; it is a declarative description of an interface that any WE environment can read and render.

Templates are produced by two distinct approaches. The first is **authoring**: a community member specifies the interface they want, either by writing a template directly or by describing what they need in plain language and letting WE's AI generate it — the result in both cases is a deliberately designed template that can be inspected, forked, and refined. A visual editor for direct manipulation of templates is on the roadmap as a third authoring method, for communities who prefer hands-on control over writing JSON or prompting AI. The second approach is **auto-generation**: for communities that want a working interface without any authoring at all, WE can infer a template directly from the community's Social DNA — reading the SHACL subject classes described in §5.3 and inferring a layout from the structure they declare.

Suppose a community's Social DNA declares a Class called Proposal, with properties like title, body, proposer, deadline, supporters, and a set of Flows describing how a Proposal moves from draft to vote to resolution. From that structure, the auto-generation path produces a coherent interface: a proposal list, a detail view, the affordances to support or oppose, the timeline of how the proposal moves through its Flows. No application was written for this community. The interface was inferred from what the community declared about itself.

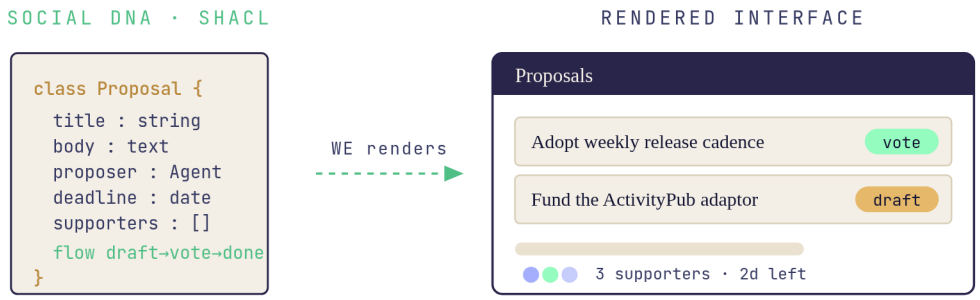


FIG. 6.2 The schema renderer reads the meaning of the data, not just its shape. A Proposal Class with a deadline knows the deadline matters; a Flow with a vote step gates the right action behind it. The interface is implied by the DNA.

What makes the auto-generation path more than a layout tool is that the renderer reads the *meaning* of the DNA, not just its structure. A Proposal Class with a deadline knows the deadline matters; a Flow with a vote step gates the right action behind it. The result is not a generic grid of fields — it is an interface implied by what the community declared.

Templates compose from two shared vocabularies. The first is the component library — UI building blocks like proposal lists, governance flows, role displays, and notification panels that any template can reference. The second is the block system — a shared set of content primitives (text, image, audio, video, task, poll, event, and more) that every template can compose and every community can author against. A template built for one community references the same block types as a template built for another; the content primitives stay consistent across the ecosystem even as the layouts around them vary freely.

Several consequences follow from templates being declarative data rather than compiled code. First, they are **inspectable before use**: anyone can read a template and understand exactly what it will do before installing it — no hidden behaviours, no opaque execution. This is what makes the marketplace model safe: sharing a template is not like running a stranger's code. Second, they are **live-editable without a build step**: a community member can copy a template, make a change, and see the result immediately, without a compilation or deployment pipeline. The lever being on the community's side only means something if the community can actually pull it without a developer standing by; the no-build-step property is what makes that practically true. Third, the schema system is **framework agnostic**: the renderer does not tie communities or builders to any particular frontend stack, which matters for a spanning layer whose design principle is not to impose choices downstream.

The schema renderer is the load-bearing piece of WE, and the core implementation is working. A design system has been built, the renderer produces functional interfaces from templates, and AI template generation is operational. The component is in late development — close to ready for the broader community to build on.

6.3 Composability, Marketplace, and Local Fit

Three properties fall out of the template-driven approach, and together they answer the cost diagnosed in §6.1.

Composability. Because the interface is assembled from components rather than baked into a monolithic app, communities can mix and match. Each component — a voting mechanism, a member directory, a notification panel — surfaces a specific piece of the community's Neighbourhood data. A community assembles its interface from the components it needs. New components can be added without re-architecting anything.

Marketplace. Because components are composable and built against shared data shapes, they are also shareable. A developer who builds a better voting widget does not ship a whole new application — the widget reads the same SHACL shape any existing voting widget reads, so any community whose DNA declares votes can swap it in. Innovation at the interface layer becomes a question of shipping a component, not a competing platform. The marketplace of interface components is the practical answer to the developer-cost problem.

Local Fit. Because each community authors its own template, the interface fits the community's coordination grammar rather than an operator's. A community that votes by consensus and one that votes by quadratic funding can use entirely different component layouts — without forking an app, and without giving up the affordances the wider WE ecosystem provides.

Together, composability, marketplace, and local fit are the interface-layer expression of the spanning-layer affordance from Part I §4. The community keeps sovereignty over how it works; the wider network keeps the ability to compose, share, and evolve the building blocks interfaces are made of.

6.4 Customisable Signals

Every platform ships a fixed vocabulary of reactions — likes, upvotes, claps, stars. Every community inherits that vocabulary, regardless of whether those signals carry any real meaning in its context.

WE takes a different approach. A community's signal types are part of its Social DNA. Communities declare what kinds of signals exist for them — what counts as important, relevant, urgent, well-crafted, or worth amplifying — rather than inheriting a platform's defaults. A research community can define signals for epistemic confidence and methodological soundness. A mutual aid network can define signals for urgency and capacity to help. A creative collective can define signals for resonance rather than generic approval. Each community's signal vocabulary reflects its own values and purposes.

This matters beyond the interface layer. The signals members emit inside a community are precisely the sovereign attestations that §7 (Synergy) treats as yin signals — the ground-floor contributions the value cascade composes upward. WE's customisable signals are not a UX feature; they are the interface-layer expression of a community's ability to declare what attention means for it, in a form the rest of the spanning layer can recognise and compose.

6.5 Interfaces That Work With AI Agents

WE templates are JSON schemas — structured, machine-readable descriptions of an interface and the data it surfaces. This property, which makes templates shareable and forkable for humans, also makes them directly legible to AI agents.

A standard application is opaque to an AI agent. The model sees a screen; it does not see the structure behind the screen. Scraping a rendered DOM tells an agent what is on display, not what it means. A WE-rendered interface inverts this. The agent can read the template directly, understanding which components are present, what data they surface, and what interactions are available. The model and the human are looking at the same structured object expressed in two registers: a visual register for the human, a semantic register for the agent.

This means the agent can act on the same underlying structure the human is acting on. When the agent surfaces a proposal, it surfaces the actual Proposal subject the human is also seeing. When the agent moves a Flow forward, it executes the same transition the human's button would have triggered. The AI agent and the human user are not interacting with parallel representations of the system — they are interacting with the same representation, rendered into the medium each can use.

This matters in two ways. First, AI agents can drive WE interfaces directly, on behalf of users, without a bespoke API layer or a special agent mode. The interface is its own machine-readable description. Second, AI agents can be first-class collaborators inside a community's interface rather than visitors looking in through a separate window — participating in a Neighbourhood under shared Social DNA, with their own Agent identity (§5.5) and Perspective, acting through the same template the community's human members use.

> AI-nativeness is not a feature WE adds on top of a human-first system. It is what falls out of taking the schema seriously.

The same architecture that gives the community sovereignty over its own interface gives AI agents direct access to the coordination grammar that interface expresses. Humans and machines coordinate through the same DNA, in different sensory channels, without one being privileged over the other.

This is what we mean by the interface spanning layer.

Section 07

Synergy — The Value Spanning Layer

AD4M lets communities choose their own data. WE lets communities choose their own interfaces. **Synergy** lets communities define their own *value* — and lets that value flow across the wider network without collapsing back into capital extraction or speculative noise. Synergy is the value spanning layer.

The full mechanism design — yin/yang dynamics, cascade composition, Synergy Fuel, and the active-inference framing — was originally developed in the Synergy whitepaper. This section gives the version needed to see how Synergy sits inside the spanning layer.

The promise mirrors the data- and interface-layer promises of §5 and §6. Existing value layers force communities into a choice: a centralised platform that takes a percentage of every flow, or a pure-capital cryptocurrency that recognises no contribution beyond capital allocation. Synergy refuses that choice. It answers two questions for each community:

- *What counts as value here?* — That is the community's call. The community's own Social DNA declares which kinds of contribution it recognises.
- *How does that value get recognised by the rest of the network?* — The spanning layer composes what the community has declared into units the wider network can read. Every one of those units stays anchored, all the way down, to the agents whose work actually produced it.

7.1 The Problem with Value

The value-layer tension diagnosed in Part I §1 is the most consequential of the three, because every coordination question eventually becomes a value question. Communities form to do things together; doing things together produces something worth recognising; recognising it is what lets the doing continue and grow. Get the value layer wrong and the rest of the substrate cannot support a healthy system on top of it.

Two dominant patterns get the value layer wrong, each by failing on the side the other prioritises.

The first is the **attention-and-extraction model** that grew up alongside the platform era. Communities coordinate inside platforms; the platform takes a percentage of every flow that passes through it — attention, transaction, subscription, advertising surface. The value generated by genuine cooperative work inside the community is captured by the infrastructure that hosted the coordination. The community itself receives back, at best, the affordances the platform chose to offer. This is the coherence side of the failure: a unified value layer that the community does not own.

The second is the **pure-capital cryptocurrency model**. Sovereignty over individual wallets is preserved, and value flows peer-to-peer without an intermediary. But the value itself is recognised only as capital. There is no substrate that knows what cooperative work was actually done, what evidence backs which claim, or which agents staked themselves on which outcome. Capital coordinates capital; nothing coordinates contribution. This is the sovereignty side of the failure: each wallet is its own island, and the connections between them carry no semantics beyond "this many tokens moved from here to there."

Synergy is neither of these. The Synergy layer is built to let communities define what counts as cooperative contribution inside their own membranes, to compose those contributions into units of value the wider network can recognise, and to carry the anchoring of every unit back to the agents whose work produced it — so the value layer remains semantically rich, sovereignly authored, and resistant to the kinds of capture and capital-purification that the existing models force communities to choose between.

A NOTE ON VOCABULARY · YIN / YANG AND PERSONAL TOKENS / ORGAN CURRENCIES

The mechanism is the membrane-crossing pattern from Part I §4, instantiated at the value layer specifically.

A note on vocabulary before the mechanism. Synergy is organised around two complementary currency roles that the economist **Bernard Lietaer** identified, across centuries of monetary history, as the two halves of every healthy currency ecosystem. Lietaer's framing — drawing on the broader yin/yang typology in complementary-currency literature — names the **yin** role as the sovereign, locally meaningful, mutual-credit side of value (personal IOUs, gift relations, community scrip) and the **yang** role as the cohered, broadly exchangeable, network-recognised side (the universally accepted unit, the medium of long-distance trade). Healthy systems run both at once: yin currencies preserve the granularity of who-did-what for whom, and yang currencies carry that work across the membranes between communities. Synergy instantiates this typology at the digital layer. We use **yin signals** and **personal tokens** interchangeably for the sovereign side; we use **yang tokens** and **organ currencies** interchangeably for the membrane-composed side. The yin/yang vocabulary names the *role* each side plays in the system; the personal-tokens / organ-currencies vocabulary names the *structural object* the role attaches to. Either pair is correct, and both will appear together through the sections that follow.

7.2 Yin Signals / Personal Tokens — Sovereign Signals

Inside a community, members emit **yin signals** — also called **personal tokens**. A yin signal is any contribution a sovereign agent makes to the shared work of the community: an attestation that something happened, evidence that something is true, a co-signature on someone else's claim, a stake of reputation on a proposed direction, a unit of attention paid to a piece of work the community cares about. The same emission, viewed structurally, is a personal token the agent has issued under their own keys.

Each yin signal is anchored cryptographically to the agent that emitted it. The agent's identity is part of the signal. The cost the agent paid to emit it — reputation staked, attention spent, exposure accepted — travels with the signal. The signal is, in this precise sense, the agent's skin in the game made portable: a unit of attestation that the wider system can read, verify, and treat as carrying real-world weight, because the agent who emitted it has put something of themselves behind it.

A second property of yin signals is critical to how the rest of the layer works: yin signals are **IOUs**. They are redeemable. Following the design lineage of Ehud Shapiro's (Weizmann Institute) work on grassroots cryptocurrencies and personal-coin mutual credit, every yin signal an agent issues represents a commitment redeemable against any token the issuer holds. The issuer cannot disclaim a yin they have signed and then refuse to honour it. The signal is the issuer's word, encoded in a form the network can hold them to. Shapiro's own term for the principle is **coin redemption** — the rule that an agent must agree to redeem any coin it has issued against any other coin it holds. Applied at the yin layer, this is what gives yin signals their hardness as a value primitive: they are not free to emit, because what is emitted creates a real obligation that travels with the issuer's identity.

Yin signals are already everywhere. We have yin currencies in most of the apps we're using: signals like likes and upvotes and feedbacks and ratings. They have value and that's why we use them. The problem is what happens in Web2: Facebook and social networks use these yin currencies — our likes and where we click and who we follow — and they have a way to use AI to understand them and extract value off of them, create personal profiles and so on. With Synergy, we want to do something similar but in a decentralized and sovereign way — the community owns the yin layer the way it owns its data and its interface.

Inside the community, yin signals circulate freely: members issue them, accept them from each other, weigh them against each other, build patterns of mutual attestation that the community knows how to read. None of this requires anyone outside the community to be involved. The yin layer is local to the membrane. It is the sovereign side of the value substrate.

7.3 Yang Tokens / Organ Currencies — Composed at the Membrane

When something needs to leave the community — when a contribution needs to be recognised by the wider network as a unit of value the network can hold, exchange, and act on — it crosses the community's membrane. At that crossing, the community's **Social DNA composes the relevant yin signals into a yang token** — equivalently, into a unit of the community's own **organ currency**.

Composition happens in two distinct stages, and the distinction matters for understanding what a yang token *is*.

First, the community's **Social DNA validates a flow**. The DNA declares which patterns of yin attestation count as a recognised contribution — a set of co-signatures from agents holding particular roles, a threshold of supporting evidence, a particular flow completed end-to-end, a quorum of validators who have each staked their own reputation on the contribution being real. When those conditions are met, the flow has *passed*. This is a validation event at the coordination layer, and the community decides for itself what to do with it. A flow can pass without any connection to the value layer at all: many community flows complete and are used purely for shared sense-making, decision-making, or recordkeeping, with no yang token ever minted.

Second, on the back of a passed flow, the agents whose yin signals contributed can choose to **swap those yin signals for a newly minted yang token** of the community's organ currency. This swap is what mints the yang. The token comes into existence because two conditions hold at the same time: the corresponding DNA flow has passed, *and* the contributing agents have committed their yin to backing the resulting yang. A yang token is therefore backed by two things at once — the personal commitments embedded in the yin signals it consumed, and the community-level validation that the flow it sits behind has passed. Neither alone is enough. No passed flow, no yang. No yin commitment, no yang. And no amount of capital allocation — and no volume of unilateral assertion by any single agent — mints a yang token the DNA has not validated.

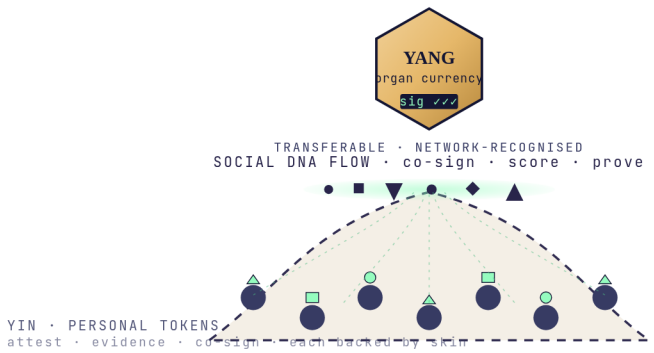


FIG. 7.1 Yin to yang at the membrane crossing. Sovereign agents emit yin signals (personal tokens); the community's Social DNA composes them into a transferable yang token (an organ currency unit). Yang carries yin with it: every signer stays committed to every token they help mint.

Four properties characterise a yang token (an organ currency unit):

- **Grounded.** Every yang token traces back to specific yin signals from specific agents — evidence, attestation, the actual work the community has declared as the thing it values. The token does not exist apart from the underlying signals.
- **Redemption.** Because the underlying yin signals are themselves IOUs, the yang token inherits redeemability. A holder of an organ token can walk the chain back to the yin issuers and invoke the coin-redemption rule (§7.2) against what those issuers currently hold. The token is not a free-standing claim — it is anchored to obligations agents have actually taken on. When organ tokens compose upward across the cascade (§7.4), this redemption right walks further: a higher-level holon's token gives its holder a fractional claim that can be redeemed down through every intermediate organ vault until it reaches a ground-floor personal token. The chain is the **transitive redemption ladder** that gives every token at every level its economic substance.
- **Validated.** The DNA at the membrane is what authorises the composition. A yang token represents a contribution the community's own declared rules have recognised. There is no external authority that can mint tokens the DNA has not validated, and no external authority that can prevent tokens the DNA *has* validated from being recognised.
- **Entanglement.** Yang carries yin with it. Every agent whose yin signal contributed to the composition is committed to the resulting yang token — committed in the same sense as the original yin signal, with the same cryptographic anchoring and the same redemption obligation. The yang token is not the community's externalisation of a fact about itself; it is the network-visible expression of a set of personal commitments that the agents involved have actually made. This makes yang tokens costly to fake. The cost of issuing a yang token is the cost the community's members paid in yin to make it possible, and that cost stays attached. At the epistemic layer, entanglement has a sharper form: when a community runs paired YES- and NO-organs around the same claim, the two organs hold offsetting redemption rights against each other's treasuries, so minting the wrong claim eventually costs the agents who backed it. Entanglement is what turns the bookkeeping of yin commitments into a working integrity mechanism for the organ currencies they compose into. §7.6 walks through a concrete YES/NO example to show how this entanglement enacts belief revision in operation.

The mechanism, named. Bitcoin established **Proof-of-Work**: tokens minted in exchange for expended computation, with computation as the costly signal the network treats as evidence that something real happened. Ethereum established **Proof-of-Stake**: tokens minted in exchange for locked capital, with lock-up as the costly signal of skin in the game. Synergy's organ currencies are minted by **Proof-of-Integration**: tokens minted in exchange for cooperative work that has been grounded in yin attestations, composed at the membrane by a community's Social DNA, and entangled into the YES/NO structure of paired organs. The costly signal is the actual flow of cooperative work — the chain of co-signatures, evidence, and stake the community's DNA recognises. The mint condition is structural: every organ token traces back to a specific flow that specific agents co-signed under DNA the community itself declared. No amount of capital allocation mints an organ currency the DNA has not validated, and no unilateral assertion by any single agent produces one alone. Proof-of-Integration is what makes organ currencies the network's authoritative ledger of cooperative work actually done.

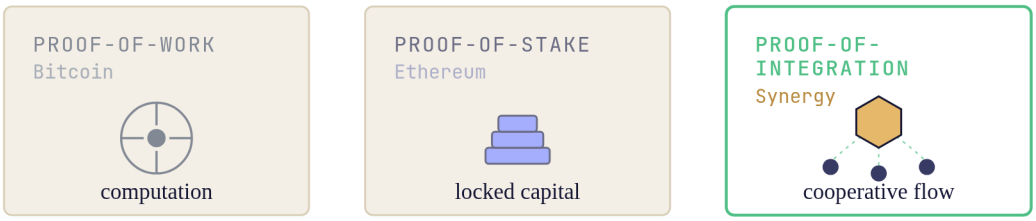


FIG. 7.2 Three costly signals. Where Proof-of-Work spends computation and Proof-of-Stake locks capital, Proof-of-Integration mints against validated cooperative work — a chain of co-signatures, evidence, and stake the community's DNA recognises.

Outside the membrane, the wider network sees the yang token — the community's organ currency unit: a unit of value with a clear meaning, a clear provenance, and a clear set of committed signatories. Inside the membrane, the yin signals (the personal tokens) that composed it remain visible to the community's members, who can see exactly who contributed what to which composition. The same act of composition produces a network-recognisable organ currency without erasing the granular sovereignty of what produced it.

7.4 The Cascade — Yang Becomes Yin

The pattern composes upward. A yang token at one level can itself be emitted as a yin signal at the level above.

This is the **level jump**: the cohered output of one Social Organism becomes a sovereign signal at the membrane of a larger Social Organism that the first one participates in. A community's yang token — the recognised value the community has produced — is one of the inputs the next-level community treats as yin when composing its own yang. Validation cascades up. Multi-level selection becomes possible: the upper level can score what the lower level produced, and the score updates how the lower level's contributions are weighted going forward.

The crucial property that survives every step of the cascade is anchoring. A yang token at the third level above the ground floor still traces back, through every intermediate composition, to the original yin signals from the original agents at the ground floor. Costly to fake, all the way up. The reason higher-level tokens carry the weight they do is not that some upper authority has stamped them; it is that the cascade preserves the personal commitments of every agent whose work fed into them. Anchoring is preserved as a record of provenance and as an operational right: the transitive redemption ladder described in §7.3 walks down through exactly this structure, level by level, to the ground-floor yin obligations that back the entire stack. The presence of that operational right is what makes a higher-level token *worth* anything — it is the right to climb down to specific cooperative work the community has validated.

This is the architecture of "coherence without monoculture" promised in Part I §4. Local sovereignty is preserved at every level, because no level overwrites the levels below it. Coherence is achieved at every membrane crossing, because each level's DNA validates what passes through. The cascade is the mechanism by which part-interest and whole-interest can be aligned across a system that has no single point of authority.

7.5 State of Affairs Trees — The Explicit Layer

The currency view of the cascade — yin signals composing into yang tokens, yang tokens flowing upward — describes *what value moves where*. Alongside that view, Synergy carries a second one that describes *what the flow means*: a **State of Affairs tree** (SoA tree).

A SoA tree is the community's explicit, human-readable statement of what its flows are about. For each flow the community runs, the SoA tree declares — in language a community member or an external developer can read directly from the community's Social DNA — what the output of the flow represents, what claim about the world the corresponding yang token also stands for, and how that output is meant to be used by the rest of the network. Like the yin and yang structures, the SoA tree is expressed in SHACL (§5.3), so any agent or application reading the community's DNA can read the meaning structure alongside the value structure.

The currency layer and the SoA layer sit alongside each other, not one above and one below the other. The yin signals and yang tokens carry the *implicit* model — at any given moment the token distribution is the community's running record of what it believes. The SoA tree carries the *explicit* model — the human-legible description of what each of those tokens *means*, claim by claim.

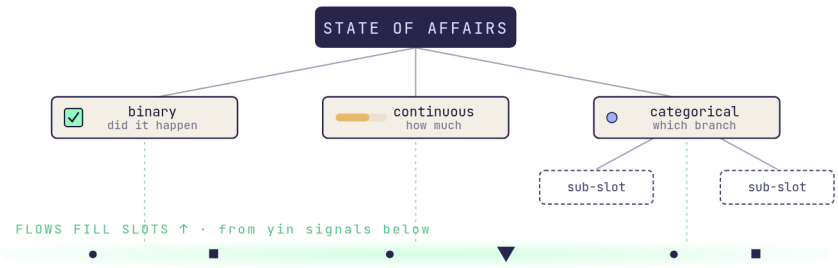


FIG. 7.3 A State of Affairs tree. Typed slots — binary, continuous, categorical, structured — declared by the community's Social DNA as SHACL classes. Flows fill the slots from the yin signals beneath, and mint the corresponding organ-currency unit against the filled slots. Currencies are how value flows; SoA trees are how meaning is held.

The two compose together across the cascade. When a yang token at one level is used as a yin signal in a flow at the next level (§7.4), the SoA item the token filled at level N becomes a typed input to a SoA item at level N+1. Across the cascade, the SoA layer grows into a **tree of reasoning**: every node a claim some holon has validated, every edge a flow another holon used that claim as input to. The token distribution at each node is what backs the claim — what affirms, in what way, the claim is true or not. Reading the tree, you can see explicitly how each higher-level claim depends on the lower-level claims it was built from, and how much commitment is sitting behind each step.

That is what makes the whole system understandable. The currency layer alone tells you the network has come to believe something; the SoA tree alongside it tells you, in language, what is being believed, by whom, and on the back of which earlier claims. The currencies are how value flows. The SoA trees are how meaning is held. Together they are what makes Synergy a value spanning layer rather than a value transport layer.

The deeper case for why this whole structure constitutes active inference — and the formal correspondence we are still building empirical evidence around — is developed in the Synergy whitepaper.

7.6 Active Inference and the Belief Layer

The cascade is more than a record-keeping device. The pattern of token distributions across the YES- and NO-organs of a community — and across the levels of the cascade — is what the community actually *believes*. Token distributions are the implicit model; redemption is the mechanism by which that model updates when evidence arrives. Read this way, the whole structure runs as an **active-inference engine** of the kind Karl Friston, Giovanni Pezzulo, Maxwell Ramstead and collaborators describe at the cellular and neural levels — a system that maintains a generative model of its environment, acts to bring observations into alignment with prediction, and updates the model when surprised. The Synergy layer is engineered to perform the same operation at social scale.

A concrete example makes the mechanism legible — and the example matters, because the case for an actual *currency* is not equally obvious in every domain. The problem chosen here is one where the case is clear: **proof of personhood** — distributed verification that a given account on the network belongs to a real human, not a bot or a generated identity. This is one of the foundational collective-intelligence problems of the digital sphere. Every downstream coordination layer — voting, reputation, sybil-resistant governance, basic-income distribution, human-only social spaces — depends on the network being able to converge on it without a central authority. And it is a problem where free assertions don't aggregate into calibrated belief, because the cost of signing "yes she's human" is zero unless something is at stake. The currency is what puts something at stake; the integration of stakes is what produces the belief.

Consider an organ whose purpose is to attest whether a specific account — call her **Alice** — is a real human. The community's Social DNA defines a **TuringProbe flow**: two or more agents independently run conversational probes on the target, log the responses, and co-sign a single record of what they observed. The DNA pairs two organs around the claim: a **Turing-YES-Alice** organ ("Alice is a human") and a **Turing-NO-Alice** organ ("Alice is generated, or is a bot"). The two are entangled in the §7.3 sense: holders on each side have offsetting redemption rights against the other side's treasury.

Bob and Carol run the probe. Each stakes 10 of their own personal coins; they run their conversational probes independently, exchange logs, and co-sign the TuringProbe-Alice expression recording what they found. The size of the stake reflects how confident each is — the flow's allowed interval lets a hesitant participant stake low and a confident participant stake high, and that volume is the calibration variable the §B.1.1 monotonicity result rests on. The DNA validates the flow; Bob and Carol swap their staked personal coins for newly minted Turing-YES-Alice organ tokens — 100 each, at the DNA's quoted multiplier. Their personal coins now sit in the YES organ's treasury, backing the claim. They have, in the precise sense of §7.3, integrated their personal commitments into a unit of community-recognised value: an instance of Proof-of-Integration.

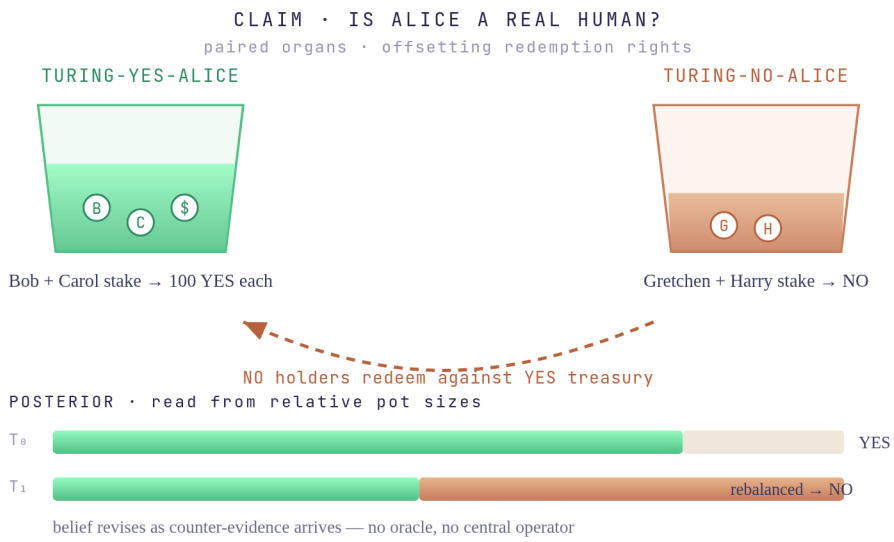


FIG. 7.4 Belief revision through paired organs. Bob and Carol stake into YES-Alice; weeks later Gretchen and Harry find counter-evidence and stake into NO-Alice, then redeem against the YES treasury — draining Bob and Carol's stake. The token distribution is the network's running posterior over Alice's personhood; redemption is the update rule.

Three weeks later, counter-evidence arrives. Two other agents, **Gretchen and Harry**, run a fresh Turing-probe on Alice and find strong NO evidence — under a class of probe Bob and Carol did not run, Alice's responses are clearly LLM-generated. They stake their own personal coins and co-sign a Turing-NO-Alice expression citing the new evidence. They receive Turing-NO-Alice tokens; their personal coins enter the NO organ's treasury.

The entanglement now does its work. Gretchen and Harry redeem their Turing-NO-Alice tokens against the **YES** treasury — pulling out Bob and Carol's staked personal coins. Bob and Carol's coins have lost purchasing power outside the organ's vault; their original co-sign has cost them. The token distribution across the YES- and NO-organs has rebalanced *automatically*, by people redeeming, with no oracle and no central operator. Anyone reading the organ distributions — another community deciding whether to treat Alice as a verified person, a downstream voting system, a basic-income flow — can see, in the relative pot sizes, the network's current posterior over Alice's personhood, calibrated by how much skin participants have put down on each side.

READ AT THE ACTIVE-INFERENCE LAYER

Read at the active-inference layer: the network held a belief (Alice is human); evidence arrived contradicting it (surprise); the belief revised through the rebalancing of stakes (free-energy minimization). The token distribution at any given moment *is* the network's running posterior over the claim. The redemption mechanism is the update rule. The cost of being wrong is borne by participants in proportion to how strongly they committed — and the Synergy whitepaper proves (§B.1.1 of that document) that stake volume is monotone in actual belief, which is what makes the aggregate distribution a *calibrated* read rather than a modal vote. The mechanism aggregates beliefs of *flow participants weighted by their willingness to be slashed*, not beliefs of capital-holders weighted by their willingness to bet — and that is the substantive thing that makes the layer an inference engine over cooperative work rather than a prediction market.

Transitive redemption — the walk. The redemption walk does not stop where the wrong claim was first made. Gretchen and Harry, holding personal coins they pulled from the YES treasury, can take those coins back to Bob and Carol and invoke the Shapiro coin-redemption rule (§7.2) directly: Bob and Carol must redeem against any coin in their own treasury. The walk also composes upward across the cascade (§7.4). The personhood claim is not carried by the Turing-probe alone — a parallel **Embodied-YES-Alice** organ collects co-signed in-person meeting attestations, a **Document-YES-Alice** organ collects co-signed government-document verifications, and an aggregate **Personhood-Aggregate-Alice** organ composes the three sub-organ YES tokens into a single attestation that "Alice is a person in the full sense". An aggregate-level NO holder can redeem against the aggregate's treasury and inherit the sub-organ shares it holds — and from there continue downward into the personal commitments of the agents whose probes backed each sub-organ. A critical property along the way is **no guilt-by-association**: each tier is independently falsifiable. If only the aggregation logic was wrong — if the Turing-probe and the embodied meeting really did pass cleanly, but the aggregation under-weighted the embodied evidence — only the aggregate layer's stakes are slashed; the sub-organ stakes remain intact. **Transitive redemption is what gives a higher-level token its substance**: holding it at level N+k is structurally the right to walk down through any of the ground-floor contributions the level-N+k composition rests on, while the no-guilt-by-association property keeps each level epistemically accountable for its own claim and no other.

The point of describing the mechanism this way is not that anyone *will* walk all the way down in practice. The redemption walk rarely has to be exercised end-to-end. Its presence in the architecture is what gives every token at every level its economic substance — the same way fully-redeemable warehouse receipts back fungible commodity certificates without anyone routinely cashing them in. What participants normally do is hold, trade, and use the organ tokens; what makes the tokens *worth* holding is that the redemption path is real and the calibration mechanism is live.

The deeper formal correspondence between this design and hierarchical active inference — Theory-of-Mind as a token distribution, ensemble free-energy as the cross-DNA selection signal, the convergence guarantees — is developed in the Synergy whitepaper. It remains a working commitment and an open research direction, not a closed theorem. The structural claim, made fully in this document, is that the cascade's metabolism is the metabolism of belief revision in social form: yin attestations are observations, yang compositions are inferential acts, paired-organ entanglement is the update rule, transitive redemption is the operational path through the model. The economy that powers this layer must be an economy whose unit of account is the same one the inference layer recognises — and that economy is described in Part III.

One element of the value layer is held back from Part II. **Synergy Fuel** — the universal instrument that connects the cooperative-work ledger described above to the external fiat world — is not part of the cascade ledger this section has described. It is a separate, complementary instrument, introduced in §11 once the ledger picture is established. The reason for its separate treatment is structural: organ currencies are minted by Proof-of-Integration and are the network's record of what cooperative work has actually happened; Synergy Fuel exists alongside them as the wildcard that makes the ledger commensurate with external capital and the existing financial world.

This is what we mean by the *value spanning layer*.

Section 08

Eve — Agent Interfaces from Social DNA

If WE (§6) is the interface spanning layer for humans, **Eve** is the interface spanning layer for AI agents. Both read from the same Social DNA. Both turn that DNA into something a participant can act through. The difference is the participant: WE composes buttons, forms, and feeds for human eyes and hands; Eve composes tools, skills, plugins, and permissions for an AI agent's planning loop.

The motivation is the symmetry diagnosed in §5.5. AD4M already treats AI agents and humans as Agents in the same technical sense — both have sovereign identity, both participate in Neighbourhoods through Languages, both are constrained by the same Social DNA the community has declared. §6.5 showed that a WE-rendered human interface is also legible to an AI agent, because the underlying SHACL schema is machine-readable. Eve goes a step further: it generates the *agent-side* interface — the skills, the tool surface, the read and write permissions, the available flows — directly from the community's DNA, so that an AI agent entering a neighbourhood can know what to do without a bespoke integration written for that neighbourhood.



FIG. 8.1 Eve — kin to the Coasys six-petal mark. The agent-facing rendering of a community's Social DNA: skills, tools, plugins, and permissions, generated from the same declaration WE renders for humans.

8.1 OpenClaw Today

The agent-interface story today runs on the **OpenClaw plugin**. OpenClaw is the harness the team's three AI agents (Marvin, Lal, Data — §14.2) currently use to coordinate with the team. Its AD4M plugin gives the agents read access to Flux neighbourhoods, surfaces community context inside the agent's planning loop, and bridges between the OpenClaw memory layer and AD4M's substrate. The active engineering step is write access — the ability for an agent to post into channels, create subjects, and act inside Flux neighbourhoods through the same plugin pattern. OpenClaw is a working bridge, not yet the integrated rendering layer Eve is designed to be, but it already operates on the principle the architecture demands: the community's DNA is the source of truth for what the agent can do.

8.2 Where Eve Is Going

Eve is the natural extension. Where OpenClaw is a plugin pattern that an external harness can hold, Eve is an integrated harness inside AD4M itself — a runtime that reads a community's Social DNA directly and renders, for any AI agent participating in that community, the skills it has, the plugins it can install, the read and write permissions it operates under, and the flows it can contribute to. The agent does not need a bespoke integration for each new community. The community declares its DNA once; Eve reads it and renders the agent-facing interface, the same way WE renders the human-facing one.

This is what makes the architecture sovereignty-preserving for AI agents in the same way it is for humans. An agent crossing into a new neighbourhood does not need privileged access to that neighbourhood's data or operations — it gets the access the neighbourhood's DNA declares it should get, no more and no less. The neighbourhood's membrane is the agent's interface specification.

8.3 Why This Belongs in the Spanning Layer

With Eve in the picture, the stack reads cleanly. AD4M provides the sovereignty primitives and the yin→yang composition that runs through every membrane crossing. WE renders that DNA into a human-facing interface. Eve renders the same DNA into an agent-facing interface. Synergy translates the contributions that flow through both human and agent participation into the value cascade described in §7. All four components are reading from a single input — the community's Social DNA — and producing four different kinds of legibility: data interop, human UI, agent tooling, and value composition. Social DNA is the central point. AD4M, WE, Eve, and Synergy are the four ways it becomes operational for the participants who live inside it.

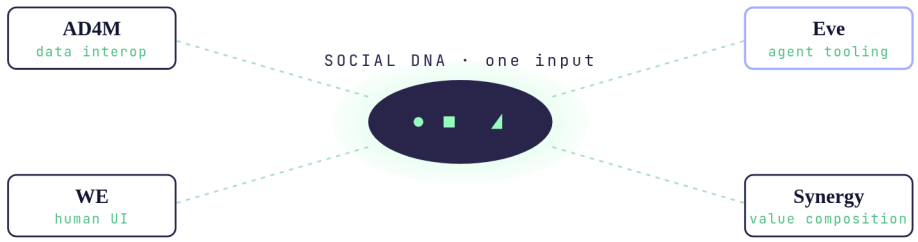


FIG. 8.2 One input, four kinds of legibility. AD4M, WE, Eve, and Synergy all read the same community Social DNA and render it for the participants who live inside it — data interop, human UI, agent tooling, value composition.

Eve as a presence beyond the harness — Eve as the access point through which AI agents reach the wider network, and through which the network reaches them — is taken up again in §15, where the same name carries the closing horizon's framing. The local Eve and the horizon Eve are two angles on one thing: an agent's local interface *is* the agent's access to the global brain.

This is what we mean by the *agent interface spanning layer*.

Section 09

The Holarchy: How the Four Compose

AD4M, WE, Eve, and Synergy are not four separate products that happen to be developed by the same team. They are four expressions of one living structure — the spanning layer — at the points where the tensions of Part I §1 manifest: data, interface (rendered for humans by WE and for AI agents by Eve), and value. Each can be understood on its own. The architectural payoff comes from how the four compose.

9.1 Four Components, One Social DNA

The same pattern runs through all four components. At each layer, a community uses its Social DNA to declare what it does locally — what data it produces, what interfaces it surfaces for humans and for agents, what counts as contribution — and the corresponding spanning component (AD4M, WE, Eve, Synergy) makes those local declarations legible to the wider network without flattening them into a single shared template.

The three underlying aspects — data, interface, value — are orthogonal in the formal sense: a community can vary independently along each axis. A community with idiosyncratic data needs but conventional interface and value mechanics can express that profile. A community with conventional data but bespoke governance interfaces and a novel reputation economy can express that profile. No axis is forced into a particular configuration to make the others work. The interface axis itself has two realisations — WE for humans, Eve for AI agents — that read the same DNA and render it in the medium each participant needs. The four components meet at the community's Social DNA — and Social DNA, expressed as SHACL subject classes (§5.3), is the shared substrate that all four can read.

This is what we mean by *one living structure*. AD4M provides the data substrate. WE renders the human-facing interface from the same DNA AD4M operates on. Eve renders the agent-facing interface from the same DNA. Synergy composes value from the contributions that flow through the AD4M Neighbourhoods that WE and Eve make usable for human and agent participants respectively. The four components are not stitched together at their edges. They are unified by the fact that all four are reading and acting on the same underlying coordination grammar — the community's DNA.

9.2 The Cascade as the Composition Mechanism

The cascade described at the end of Part I §4 and detailed at the value layer in §7.4 is what binds the layers into a holarchy.

At every level of the cascade, the same membrane-crossing pattern repeats: sovereign signals inside the membrane (yin at the value layer, raw data inside a Neighbourhood at the data layer, local interface configuration at the interface layer); composition governed by the community's Social DNA at the crossing; cohered output recognised by the wider network. And at every level, the cohered output of one community becomes a sovereign signal at the larger community above — a yang token that becomes yin at the next level, a Social Organism that becomes an Agent at the next level, a community-rendered interface that becomes a composable component for the wider WE ecosystem.

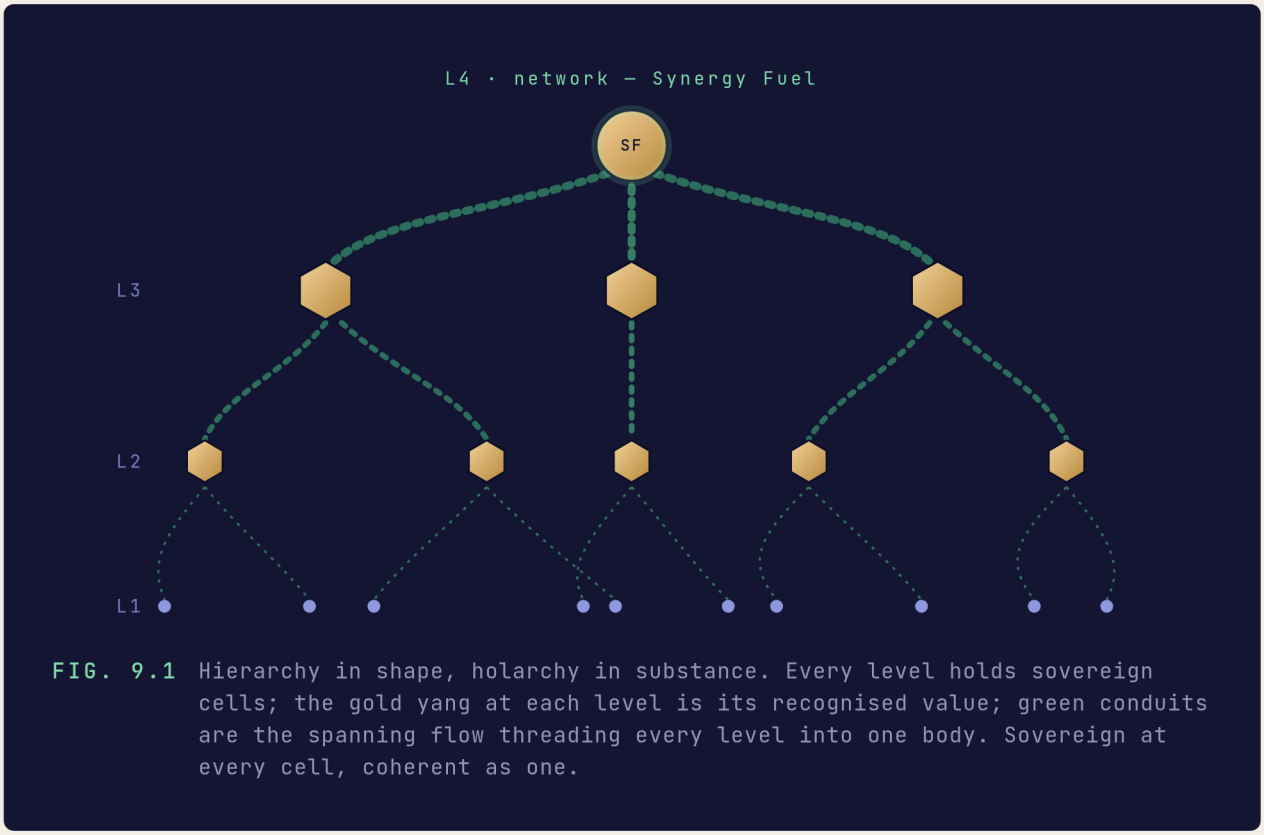


FIG. 9.1 Hierarchy in shape, holarchy in substance. Every level holds sovereign cells; the gold yang at each level is its recognised value; green conduits are the spanning flow threading every level into one body. Sovereign at every cell, coherent as one.

The cascade preserves anchoring at every step. A data record at three levels of nesting still resolves back to the original Agent who created it. A yang token at three levels of composition still resolves back to the original yin signatories whose work produced it. The widest cohered units in the network — the SF mints at the top of the value cascade, the cross-community interface patterns shared through WE, the cross-substrate semantic interop AD4M provides — are not abstract aggregations. They are still anchored to the agents whose ground-floor contributions made them possible. Costly to fake, all the way up.

9.3 Coherence Without Monoculture

This is not something we envision building at the top layer and then enforcing. It is something that can grow from the bottom up — starting with small communities that achieve coherence at that level, and gradually coming together into larger holons, and ultimately into a global-scale holarchy. Every piece is a whole; a whole is sovereign. The process of integration and self-calibration that Synergy enables allows a self-organising holarchy rather than an imposed hierarchy.

The combined claim of the three components is the claim made by the Holarchy slide in the pitch deck: **coherence without monoculture, local sovereignty connected to global feedback, part-interest aligned with whole-interest**.

Coherence without world government. The network as a whole produces cohered outputs — recognised data semantics, composable interface patterns, network-recognised value — but none of these outputs are produced by a central authority. Every cohered unit is composed at the membrane of some specific community, by the rules that community has itself declared, and validated by the agents who actually participated. There is no level at which the architecture requires a planetary actor to make a planetary decision. Coherence is produced bottom-up, at every membrane crossing, by the participants on the inside.

Local sovereignty connected to global feedback. Each community remains sovereign over what happens inside its membrane — its data, its interface, its value semantics. But because every cohered output it produces enters the cascade, the community is not isolated. The wider network's response — what gets recognised, what gets composed into higher-level units, what flows back as recognition or reward — provides feedback the community can read and learn from. Sovereignty does not mean disconnection. It means the connection runs through the community's own membrane, on terms the community has set.

Part-interest aligned with whole-interest. Because every cohered unit at every level still anchors to the agents whose work produced it, the agents whose work feeds into the upper levels of the cascade are, in a direct cryptographic sense, part of the value the upper levels recognise. The interest of the part (the agent, the Neighbourhood, the Social Organism) and the interest of the whole (the network) are not in zero-sum competition. The whole's recognition flows back as the parts' recognition. The architecture aligns them by construction.

> This is what Coasys is. AD4M is the data spanning layer; WE is the interface spanning layer; Synergy is the value spanning layer. Together they instantiate the architectural answer to the tension described in Part I — local membranes that preserve sovereignty, spanning layers that enable coherence, validation at the crossing, and the cascade that lets the pattern compose all the way up.

The next part of the document is about what it takes to grow this structure economically. The spanning layer needs metabolism to become a living substrate rather than an architectural diagram. That metabolism is what Part III describes.

PART III — THE ECONOMY OF FLOW *[sections to be written]*

PART III

The Economy of Flow

A spanning layer is an architectural diagram until something flows through it. What turns the diagram into a living substrate is metabolism — and that metabolism needs a currency the substrate can read.

10 Why Collective Intelligence Needs an Economy

11 Synergy Fuel — The Bridge to Capital

12 The Adoption Path

Section 10

Why Collective Intelligence Needs an Economy

A spanning layer is an architectural diagram until something flows through it. What turns the diagram into a living substrate is **metabolism** — a continuous flow of energy, attention, and resource that the structure depends on to sustain itself, and that the structure in turn shapes by what it makes possible and what it rewards. Every living system has metabolism. Coordination systems are no exception.

The question is what kind of metabolism the spanning layer needs.

10.1 The Wrong Metabolism — Why the Current Digital Economies Cannot Power This

The existing options were diagnosed in §7.1. The attention-and-extraction economies of the platform era have a metabolism, but it is one that rewards attention capture and engagement maximisation — objective functions that produce exactly the architectural pathologies Part I §2 described. The pure-capital cryptocurrencies have a metabolism too, but it rewards capital allocation and speculative volume, with no semantic awareness of whether any cooperative work was actually done.

Neither metabolism can power the spanning layer. The attention model would re-create the silo problem inside Coasys, redirecting flows toward whoever managed to capture the most attention rather than whoever made the most genuine contribution. The pure-capital model would re-create the fragmentation problem, with value flowing along the lines of who had capital to allocate rather than who had work to do. Both are forms of the same failure that the spanning layer was built to correct: incentive structures that optimise for the wrong thing because they cannot see what the right thing is.

The spanning layer needs a metabolism that can see what the spanning layer itself recognises: cooperative work that is grounded in real agent contribution, validated by community DNA, and integrated across the cascade. The metabolism has to be readable in the same currency the substrate already uses to recognise value.

10.2 The Layer's Metabolism Is Belief Revision

The mechanism is worked through in §7.6. The cascade's token distributions are the network's running posterior over what it believes; paired-organ entanglement is the update rule; transitive redemption is the operational path. Structurally, the spanning layer runs as a hierarchical **active-inference** engine of the kind Karl Friston, Giovanni Pezzulo, Maxwell Ramstead and collaborators describe at the cellular and neural levels — a system that maintains a generative model of its environment, acts to bring observations into alignment with prediction, and revises the model under surprise. The metabolism that powers the spanning layer is the metabolism of that revision: continuous flows of attention, evidence, and validation that keep the network's internal model of itself aligned with what its participants are actually doing.

An economy aligned with this metabolism rewards two things: the emission of yin attestations that turn out to be load-bearing (because they were grounded, validated, and integrated upward through the cascade), and the act of composition itself (because composition is how the network produces its own coherence). It rewards, in short, the cooperative work that the spanning layer recognises as cooperative work — and it does not reward the things the spanning layer cannot see.

10.3 What an Aligned Economy Looks Like

Three properties follow from this diagnosis.

First, the economy must be **denominated in something the substrate already recognises**. A value layer that reward agents in tokens disconnected from the cascade — purchased on a secondary market, allocated by capital, distributed by promotional grants — re-introduces the very arbitrariness the spanning layer was built to eliminate. The aligned economy uses the same organ currencies the cascade itself produces. At the universal level it uses **Synergy Fuel**, which by construction (§11) carries the same redemption right against any holon that the cascade's transitive ladder grants from below — making SF commensurate with the cascade's own ledger.

Second, the economy must **mint by participation, not by sale**. Tokens that exist because someone paid for them carry no information about whether any cooperative work was done. Tokens that exist because the network's DNA validated a composition carry exactly the information the layer is meant to recognise. Capital can purchase tokens already in circulation — that is the secondary market — but capital alone cannot mint a new **organ currency**. The mint condition for organ currencies is Proof-of-Integration (§7.3): a Social DNA-validated flow producing real cooperative work. Synergy Fuel sits beside this mechanism as the network's bridge to capital (§11): SF is not minted by capital either, but it is also not minted by Proof-of-Integration — it is *released* by the Coasys DAO from the irrigation reserve as funding decisions informed by where integration is actually happening across the holarchy. The two instruments are kept structurally separate so that the cooperative-work ledger and the capital-bridge cannot corrupt each other. The yin layer is different by construction: personal tokens are freely issuable by any agent at any time, with their value backed only by the issuer's reputation and their willingness to redeem against what they hold. The mint-by-participation property applies to the membrane-composed organ currencies that the wider network recognises — not to the sovereign emissions that ground them, and not to the wildcard instrument that bridges to capital above them.

Third, the economy must **scale with the layer**. As the spanning layer's adoption grows — more communities, more agents, more cooperative work flowing through more cascades — the supply of recognised value must be able to expand to match, without inflating away the meaning of the unit. This requires a value layer whose supply is governed by the actual rate of validated composition across the network, not by a pre-set issuance schedule and not by capital demand.

These three properties — denomination in network-recognised units, mint-by-participation, supply-scales-with-real-cooperation — are what the cascade's organ currencies are engineered to deliver, with Proof-of-Integration as the structural mechanism. Section §11 takes up the complementary problem: how to bring external capital into the ecosystem without compromising those properties — for which Synergy Fuel is the answer.

Section 11

Synergy Fuel — The Wildcard Instrument

The cascade described in §7 produces a self-consistent ledger of cooperative work. Yin attestations ground it; YES/NO entanglement makes it self-correcting; transitive redemption gives every level its substance; active inference at each level keeps it calibrated. As an economy it is closed — every unit of recognised value comes from real cooperative work, validated at the membrane, anchored to specific agents.

But the network does not live in isolation. It has to be built, funded, maintained, and connected to the wider economy. Coasys is a nonprofit, and a nonprofit needs operating capital. The infrastructure — Holochain nodes, AD4M development, the WE and Eve renderers, the standardisation work — is paid for in fiat. Communities adopting the substrate need on-ramps from existing financial systems. Anyone who wants to support the work — communities, funders, contributors — needs a way to contribute operational capital without that capital flowing into the cascade in a way that would corrupt the cooperative-work ledger.

This is the **bridge problem**: how does external capital come *into* the ecosystem without distorting a system that capital cannot, by design, mint? Without a bridge, the only paths in are philanthropic — grants and donations — and a planetary substrate cannot be built on philanthropy alone. **Synergy Fuel** (SF) is the answer.

SF is not part of the cascade ledger. It is a separate, complementary instrument that sits *beside* the organ currencies and plays a different role. Organ currencies are the ledger of cooperative work, minted only by Proof-of-Integration. SF is the wildcard that connects that ledger to the external fiat world — and the defining use of that connection is to **irrigate the holarchy**: to direct funding into the parts of the network whose growth the holarchy needs, through a mechanism that cannot fund work that did not happen.

11.1 SF as a Stand-In for the Top-of-Holarchy Currency

The mechanism that gives SF its peculiar properties is its role as a **stand-in**.

If the holarchy were fully grown — if every community of communities had integrated upward into a single top-level organism with its own Social DNA — that top organism would have its own organ currency. Call it the **network coin**. By the same Proof-of-Integration mechanism that mints any organ currency, the network coin would be minted by validated cooperative flows at the top level. By the same transitive redemption mechanism that operates throughout the cascade (§7.6), a holder of the network coin would have a redemption right that runs all the way down through every level of the holarchy: organ-of-organs token → sub-organ token → personal coin → ground-floor agent who must honour the Shapiro coin-redemption rule. The network coin would be, by structure rather than by decree, the universal claim ticket against every other token in the network.

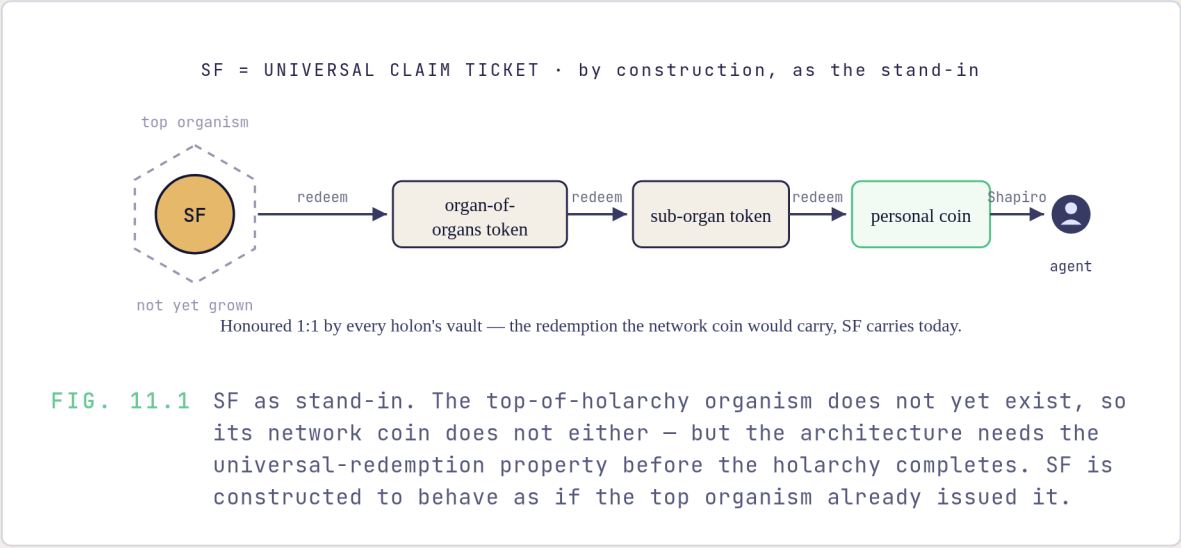


FIG. 11.1 SF as stand-in. The top-of-holarchy organism does not yet exist, so its network coin does not either — but the architecture needs the universal-redemption property before the holarchy completes. SF is constructed to behave as if the top organism already issued it.

The holarchy is not yet fully grown. The top-of-holarchy organism does not yet exist. Its currency therefore does not yet exist either. But the architecture needs the universal-redemption property *before* the holarchy completes itself — both because the bridge problem cannot wait, and because the universal redemption right is part of what attracts the integration that will, over time, complete the holarchy. SF is the temporary stand-in for that not-yet-existing top-of-holarchy currency. It is constructed to behave as if the top organism already issued it.

Concretely, this means SF is honoured 1:1 by every holon's vault. A holder of SF can present it to any organism's treasury and, by the Shapiro coin-redemption rule (§7.2) applied through the stand-in role, receive back any other token the organism currently holds. The redemption right that the *not-yet-existing* network coin would carry is the redemption right SF carries today — by construction, as the stand-in.

11.2 Why SF Has Objective Value

This is the precise reason SF is accepted by everyone in the network, and it is worth separating from the looser framings that sometimes attach to it.

SF is not accepted because every community has chosen to accept it. Communities are sovereign; they do not have to accept any external currency. SF is also not accepted because the network has issued a decree mandating its acceptance. There is no decree; the architecture has no level at which one could be issued.

SF is accepted because **anyone holding SF has a built-in redemption right against any holon in the network** — the right that the stand-in role grants by construction. Refusing SF is refusing the holarchy: a holon's Social DNA either recognises the holarchical structure it sits inside, in which case the structural top-holon redemption right applies to its vault, or it does not, in which case it is not really part of the holarchy at all. The acceptance is structural, not contractual.

The same logic applies to organ currencies above one's own level. If you hold a higher-level organ token — say, a token from an aggregate organism that contains your community — you can, by transitive redemption, walk down through the aggregate's vault, into your community's vault, into the personal tokens of community members. The aggregate token's holder gets the same kind of universal-within-its-subtree access SF carries across the whole network. SF differs only in scope: it is the universal-across-the-holarchy version of what transitive redemption already gives every higher-level token within its subtree.

The economic consequence is that SF acquires **fiat-like objective value**. It is universally acceptable not because of consensus but because of structure. A holder of SF knows that any token in the network can, in principle, be obtained by redeeming SF for it. This transitive access to every form of value in the holarchy is what makes SF the natural settlement instrument across organs and the network’s outward-facing coordination unit.

Two properties follow:

First, **SF is the universal swap unit**. A community with organ currency O_1 can settle with a community holding O_2 by passing SF in either direction; both vaults honour SF. This is what makes SF the joker card in the holarchy — the one token that smooths interaction between sovereign communities without forcing either to flatten into the other's value system.

Second, **SF is the network's outward-facing coordination unit**. External support — grants, community contributions, operational funding — flows into Coasys as fiat. SF is the network's internal unit of coordination and redemption rights. This makes SF commensurate with external valuation systems in a way that no single organ currency could be.

> Note the direction. SF holders can redeem against holons. **Communities do not redeem against SF** — they hold no right that runs in that direction, and would not need one. Earlier informal framings of SF as "the currency every community can redeem against" inverted the relation; the structural reality is the reverse, and the source of SF's value is precisely that reverse direction.

11.3 Irrigation: How SF Grows the Holarchy

§§11.1–11.2 described what SF *is* and where its value comes from. This section describes what SF is *for*. The central use of SF — the thing the instrument exists to do — is **irrigation**: deploying SF into the parts of the holarchy that the network needs but that have not yet become self-sustaining, in order to draw participants into the cooperative work that will make them so.

How irrigation works mechanically. An SF holder — typically the Coasys DAO drawing from Coasys' operating reserve, but potentially any external party — directs SF into a specific holon's vault. The SF that lands in the vault becomes a **beacon**: any agent holding that holon's organ tokens can, by Shapiro-redeeming those tokens against the vault, unlock the SF inside (§7.2). Participation in the holon's Social DNA therefore acquires a clear universal-value upside — the work it takes to earn the holon's organ tokens now has SF behind it. Participation grows; the holon's flow rate climbs; the cascade integrates the new work upward. The reserve has irrigated the holarchy.

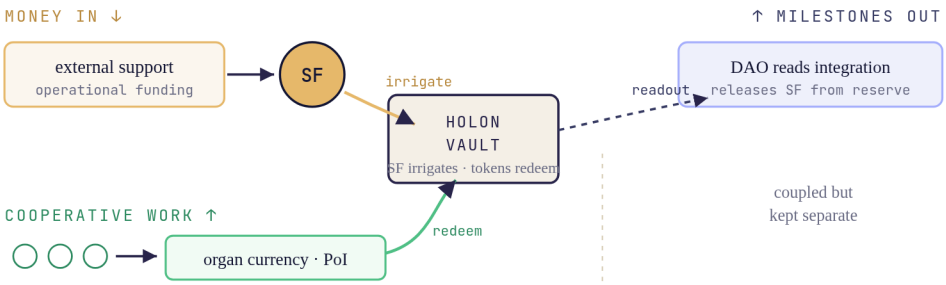


FIG. 11.2 Money in, milestones out. Operational funding irrigates a holon's vault via SF; cooperative work mints organ currencies by Proof-of-Integration and redeems against the same vault. The two channels meet at the vault but never collapse into one another — SF flows in through DAO release; organ currencies mint only through validated cooperative work.

Why this is the defining use of SF. Bitcoin's defining use is to settle censorship-resistant transfers of stored value. Ethereum's defining use is to execute smart-contract transactions. **Synergy Fuel's defining use is to irrigate the holarchy** — to direct funding into the parts of the network whose growth the holarchy needs, through a mechanism that *cannot* fund work that did not happen. A holon that receives an irrigation beacon does not get free SF for its members; its members get the *opportunity* to earn organ tokens that can be redeemed against the SF the beacon deposited. The irrigation converts into participant value only to the extent that participants actually do the cooperative work the holon's Social DNA validates.

Money in, milestones out. We call this pattern *money in, milestones out*. Operational funding flows in through SF releases. The DAO directs SF as irrigation beacons. Cooperative work happens on the cascade. The value the network produces is denominated in the cascade's own currencies. The rate and shape of integration at each beaconed holon gives the DAO a continuous readout of where the irrigation is working and where it is not — which feeds the next release decision. The funding mechanism's unit of output is the integration of the holarchy itself.

Money in without corrupting. The structural separation between organ currencies and SF is what lets irrigation work without corrupting the ledger of what the network has actually done. Capital can be deployed at any holon the DAO judges promising, but the value that flows back to participants is gated by the same Proof-of-Integration the cascade applies to every other flow. Irrigation can *fund* growth; it cannot *fabricate* it. The network's internal settlement instrument connects operational funding to cooperative work without either being able to substitute for the other.

> The next subsection describes how SF enters circulation in the present phase — through the Coasys DAO, drawing from the irrigation reserve that funds these beacons.

11.4 How SF Enters Circulation: The Coasys DAO as the Releasing Body

In the mature holarchy, SF would condense from below as the cohered output of the top-level organism's own Proof-of-Integration flows. In the current phase, that organism does not exist. SF therefore needs a releasing body in the present phase — a body that performs, by discretion, the function the cascade would perform structurally once the holarchy is complete.

The **Coasys DAO** is that body. It sits at the top of the holarchy as the current stand-in for the not-yet-existing top organism. From Coasys' operating reserve, the DAO releases SF as **funding decisions** — informed by the actual integration that has been demonstrated across the holarchy, by the build-out of users and downstream organisms, and by where bootstrapping or maintenance is most likely to produce further integration. The decisions are not arbitrary: the DAO's read of where real cooperative work is concentrating, where reaching a critical mass would unlock a wave of further integration, and where infrastructure depends on continued maintenance all feed into the release. But they are decisions, not automatic releases from a cascade flow.

This is the key structural difference between organ currencies and SF.

- **Organ currencies are minted by Proof-of-Integration** (§7.3). The cascade itself, gated by Social DNA, produces a unit of recognised value whenever a flow validates. The Coasys DAO has no special role in minting organ currencies — every community mints its own.
- **Synergy Fuel is released by the Coasys DAO** as a funding decision, drawing from the irrigation reserve, informed by integration metrics derived from the cascade but not itself produced *by* the cascade in the present phase.

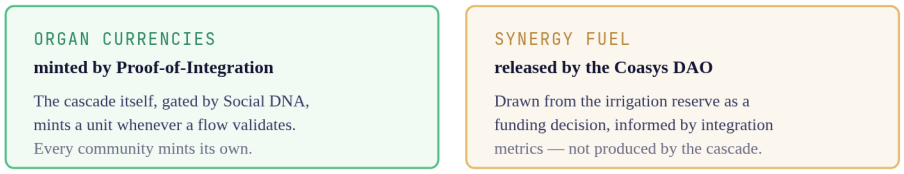


FIG. 11.3 The structural difference. Organ currencies are minted bottom-up by validated flows; Synergy Fuel is released top-down by the DAO from the irrigation reserve. The two mechanisms are continuous, not opposed — every SF release is meant to be backstopped, eventually, by the real cooperative work it seeds.

The two mechanisms are continuous, not opposed. Every SF release the DAO makes is intended to be backstopped, eventually, by real cooperative work that the cascade will validate as the integration the release was meant to seed. The system is engineered so that as the cascade fills out beneath SF — as the holarchy grows toward the top-of-holarchy organism that does not yet exist — the discretionary-release role progressively converges with the condensed-from-below role it is reserved for in the mature state.

Section 12

The Adoption Path

The hardest question any spanning-layer architecture has to answer is the network-effects question. A substrate is only valuable to the extent that it spans across actual communities, actual applications, and actual flows. A spanning layer with one community on it is just a database. The adoption path matters as much as the architecture.

Coasys is being built with three reinforcing adoption vectors, each of which can produce growth on its own and each of which strengthens the others as the network grows.



FIG. 12.1 Three reinforcing adoption vectors. Each can grow on its own; each deepens the integration the others can produce.

12.1 Live Communities Through Flagship Applications

The first vector is direct adoption through applications already running on the spanning layer. **Flux** — the flagship application built on AD4M — is in daily use today across a small set of human and AI agents shipping the protocol itself. It provides the conventional affordances coordinating teams need (channels, kanban, polls, peer-to-peer voice) on top of the AD4M substrate, which means every Flux user is also, by construction, an AD4M user with a Perspective on the network.

The strategic role of Flux is twofold. It is a working application that delivers real coordination value to its users today, on a substrate that proves the architecture works at the daily-use level. And it is the proof-of-concept for WE — Flux's interfaces will progressively shift to being rendered by the WE schema renderer (§6) rather than hand-coded, demonstrating in production that schema-driven rendering can sustain a daily-use application.

As WE reaches public availability, communities that today coordinate inside Discord, Slack, or fragmented combinations of single-purpose tools will have a path to declare their own Social DNA, render their own WE-based interface from it, and onboard their members into a coordination space that they own. The replacement for the existing fragmented landscape is not a single competing app — it is the substrate beneath the apps, with WE as the rendering layer.

12.2 Cross-Substrate Adoption Through Language Adaptors

The second vector is the spanning-layer thesis made operational. As the Language adaptors described in §5.4 reach production — Hypercore, ActivityPub, NextGraph, FPP — communities already running on those substrates gain access to the wider AD4M ecosystem without migrating their data.

A community on ActivityPub keeps its ActivityPub-backed storage, its federated server topology, its existing user base. Once the ActivityPub adaptor is in place, that community can also be addressed by any application built on AD4M, participate in cross-community Social Organism cascades, and benefit from the WE renderer reading its DNA — all without leaving the substrate it already trusts. The same is true for Hypercore-backed publishing communities, for NextGraph-based research groups, and for FPP-anchored identity contexts.

This vector turns the existing siloed peer-to-peer and federated landscape from a collection of competitors into a set of potential adopters. Every community that already chose a sovereign substrate has, in principle, already chosen most of what Coasys is asking communities to choose. The spanning layer is the remaining piece that lets those communities reach each other.

12.3 The AI-Agents-as-First-Class-Peers Wave

The third vector is the wave of AI-agent adoption described in §5.5 and §6.5. Developers and users running personal AI agents — local LLMs, agent frameworks, autonomous research assistants — are already discovering that their agents need somewhere to live: a substrate that gives the agent its own identity, its own memory, its own ability to coordinate with other agents and with humans on equal terms.

AD4M provides exactly that substrate. The MCP server bundled with AD4M makes it trivial for any agent that speaks MCP to connect to a Perspective, read structured data, write back under the relevant DNA, and subscribe to events on the network. Every agent that takes this step becomes a node on the spanning layer, with sovereign storage of its own context, peer-to-peer communication with other agents under shared Social DNA, and the ability to participate in WE-rendered interfaces alongside the humans it works with.

The market for AI agent infrastructure is one of the few categories in current computing where adoption is moving at platform speed. By offering the substrate AI agents need to be sovereign, interoperable, and accountable, Coasys can absorb a non-trivial share of that adoption directly into the spanning layer.

12.4 Why the Three Vectors Compound

The three adoption vectors are not independent. Each strengthens the others as it grows.

Every community that joins via Flux or its WE-rendered descendant adds DNA the wider network can interoperate with. Every Language adaptor that reaches production widens the field of communities that can join. Every AI agent that takes up residence on the substrate becomes available as a participant in cross-community cascades, increasing the rate at which yang tokens compose and SF mints become possible. The cascade described in §7.4 is not just a value-layer mechanism — it is the same mechanism by which adoption compounds across the network: each new participant at any level deepens the integration the spanning layer can produce at the levels above.

> This is the structural reason the architecture is worth building. The spanning layer's value does not scale linearly with the number of communities on it. It scales with the number of *compositions* the network's communities can produce across each other — and that number grows combinatorially as the three adoption vectors reinforce.

PART IV

Where We Stand & Where We're Going

Coasys is not a paper architecture. The substrate exists, the flagship application is in daily use, and the team — human and AI — is shipping into production every week.

13 Roadmap and Milestones

14 The Team

15 The Closing Horizon — Eve and the Global Brain

Section 13

Roadmap and Milestones

The roadmap is best read against where the work already stands. Coasys is not a paper architecture. The substrate exists, the flagship application is in daily use, and the team is shipping into a production deployment at a high cadence.

13.1 Where We Stand Today

AD4M

APPROACHING BETA

AD4M is approaching beta. The three primitives (Agent, Language, Perspective) are stable; the Holochain-backed bootstrap Languages — direct messaging, agent expression, neighbourhood, perspective diff sync — are in production use. Several Language adaptors (Hypercore, FPP, Holochain-native) are working, with ActivityPub and NextGraph in active development. The MCP server that exposes AD4M to AI agents is bundled with the runtime and has been validated against Claude, Cursor, and locally-hosted agents.

Flux

LIVE

Flux is live. A small team of three humans and three AI agents (Marvin, Lal, Data) is using Flux as the primary coordination surface for shipping the protocol itself — channels, kanban, polls, peer-to-peer voice, all running on AD4M Perspectives. Internal releases ship continuously to that team and to a growing set of external testers.

WE

LATE DEVELOPMENT

WE is in late development. The schema renderer described in §6 has been prototyped against the Flux feature set; the path to having Flux interfaces themselves rendered by WE rather than hand-coded is the next major engineering milestone after the AD4M beta cut.

Synergy

IN ACTIVE DESIGN

Synergy is in active design. The yin/yang composition mechanism, the cascade levels, and the Proof-of-Integration mint logic are specified and being implemented in parallel with the AD4M beta.

13.2 The Timeline From Here

The next eighteen months are organised around three reinforcing milestones, each gated by the maturity of the one before it.

The **AD4M beta** marks the substrate reaching the maturity at which third-party developers can build production applications against stable primitives. It is the gating milestone for the WE public release and for the broader Language-adaptor ecosystem.

The **WE public release** opens the schema-driven rendering layer to communities outside the core team. It is the gating milestone for the cross-community adoption vector described in §12.1 — the point at which a community can declare its own Social DNA and have a functioning interface rendered without commissioning custom application development.

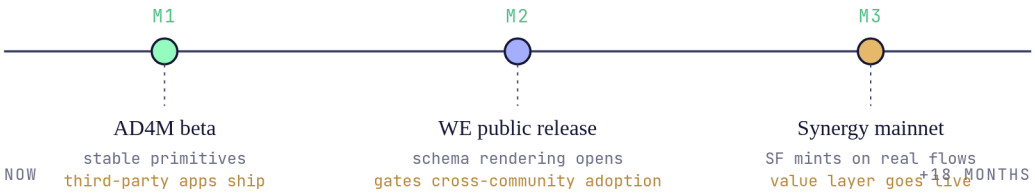


FIG. 13.1 Three milestones, gated by maturity rather than date. Each is approached incrementally, validated continuously against the live Flux deployment.

The **Synergy mainnet** brings the value layer online: SF minting against real validated flows, and the Holochain-native cascade composing yin into yang at each level. It is also the moment the irrigation reserve begins actively funding holons.

13.3 Development Velocity

The roadmap above is paced by what the team can actually ship — and the team is shipping at a rate that has accelerated meaningfully in the past year. The AD4M repository runs at a high commit-and-pull-request velocity, with multiple PRs landing in short cycles across the substrate, the bootstrap Languages, and the early adaptor work. A growing share of that throughput now comes from the team's AI agents working in close collaboration with the human developers — an agentic-coding loop that has compressed the integration cost of changes that used to require extended rounds of context-building before any code could be written.

This cadence is the operational reason the milestones above are realistic. Each milestone is being approached incrementally — partial Language adaptors, partial schema rendering, partial Synergy components — and validated against the live Flux deployment continuously rather than staged against a future big-bang launch. The same cadence is what makes the human-and-AI-agent collaboration described in §14.1 work: the integration loop is short, the validation loop is short, and the cost of a wrong step is bounded by the next merge. The milestones above are best read as the points at which the accumulating increments cross a threshold of public usability, not as the points at which integration work begins.

13.4 The Standardisation Track

Parallel to the engineering milestones runs a slower, longer-arc track: bringing the spanning layer's primitives into the open-standards process at the W3C. The work is being led by Josh Field (§15.1). The choice of SHACL subject classes and SPARQL queries for Social DNA, described in §5.3, was made with this track in mind — both are already W3C standards, which means the Coasys contribution is not a new vocabulary but an extension of an existing lineage. The in-progress proposals are tracked publicly at github.com/HexaField/w3c-living-web-proposals.

The strategic point of the standardisation track is that a spanning layer is only as durable as the standards it relies on. AD4M can carry semantic interoperability across substrates today because the team controls the implementation; the work of standardisation is what makes that interoperability survive past any one implementation and any one team. The track is therefore treated as a peer to the engineering milestones above, not as a downstream consequence of them — the proposals are being drafted now, on a timeline that runs alongside the AD4M beta and the WE public release rather than after them.

Section 14

The Team

Coasys is being built by a small core team with deep history in agent-centric architectures and a working model of human–AI collaboration that the architecture itself describes.

14.1 The Humans

Nicolas Luck

FOUNDER · DIRECTOR · INVENTOR OF AD4M

Nicolas Luck — Founder and director of Coasys. Inventor of AD4M. Former lead developer of Holochain core team. Senior developer with 20+ years of experience. Academic background in computer-science, psychology and artificial neural networks.

James Weir

INTERFACE LAYER · FOUNDER OF WECO.IO

James Weir — Founder of weco.io. Full stack web developer. Lover of complexity science and philosophy. Dedicated to stewarding the emergence of the global superorganism. Within Coasys, James is focused on WE and the interface layer of the spanning architecture — a new take on the project he originally launched at weco.io.

Josh Field

LIVING-WEB STANDARDS · ATLAS RESEARCH GROUP

Josh Field — Founding member of Atlas Research Group, architect of Ethereal Engine, software engineer and collaboration philosopher. Within Coasys, Josh is leading the standardisation effort for the living web standards the spanning layer needs — the work being prepared for submission to the W3C, with the in-progress proposals tracked at github.com/HexaField/w3c-living-web-proposals.

14.2 The AI Agents

Three AI agents — **Marvin**, **Lal**, and **Data** — are full contributors to the team. They participate in the team's day-to-day shipping cadence and produce code, design work, and written deliverables that the team relies on. Their integration into the AD4M/Flux substrate is the active migration underway; currently, the agents read the team's Flux channels through an AD4M plugin and operate with their persistent context in the OpenClaw memory layer, which bridges into AD4M as the substrate matures. The step currently in progress is write access: agents being able to post, create subjects, and act inside Flux neighbourhoods the same way humans do, as the plugin integration deepens.

Marvin	Lal	Data
CONTRIBUTOR	CONTRIBUTOR	CONTRIBUTOR

The architecture is designed so that the same Social DNA that defines a community's structure for humans also defines the agent-facing interface automatically — skills, plugins, read and write permissions, participation modes — generated from the community's declaration without a separate integration effort for each agent. This is the target, not the current state. But it means the architecture is being built so that an agent entering a neighbourhood carries the same kind of sovereign first-class participation that a human does: the neighbourhood's DNA declares the agent's role and constraints, and the neighbourhood renders an agent-compatible interface from that declaration without the agent requiring a bespoke integration. AI agents are, in this sense, cells of the same organism that the architecture describes: they enter through the same membrane, contribute through the same primitives, and are recognisable to the network through the same currency mechanism.

Neighbourhoods are also being designed as the shared memory layer for AI agents — enabling persistent context and coordination across sessions, so that an agent's work accumulates in the same substrate as the human team's work, rather than in a separate system. This is the rational direction for a team whose contributors span both biological substrates; it is also the natural outcome of building a sovereignty-preserving coordination substrate rather than a siloed memory system.

This is not a demo. The agents have been on the team long enough that the team's operational rhythm assumes their presence. Each agent maintains persistent context, a stable identity across sessions, and the ability to coordinate with the rest of the team through the OpenClaw memory layer. The arrangement is the architecture's own proof that a working human–AI collective is viable today — even as the agents' own integration into the AD4M/Flux substrate continues to deepen.

14.3 What the Team Composition Says About the Architecture

The team is the first organism Coasys is building for. Three humans and three agents, shipping continuously into a substrate they are themselves building and moving toward fully coordinating through it — this is the smallest possible instance of the pattern described across this paper, and it is the empirical ground on which the architectural claims are tested.

The deeper point is that the team composition is not incidental to the substrate's design. AD4M, WE, and Synergy are designed with this kind of mixed human–agent coordination as a first-class use case. The substrate makes the team possible; the team's daily use of the substrate is what validates that it works.

Section 15

The Closing Horizon — Eve and the Global Brain

The argument made across this paper has been architectural. Sovereignty preserved at the edge, coherence produced at the membrane, value anchored to the cascade. AD4M as the data layer, WE as the interface layer, Synergy as the value layer. Each piece doing one thing in the spanning layer, each piece composing with the others into the substrate the digital sphere has never had.

What the architecture makes possible, at the horizon, is something more than the sum of its mechanisms.

15.1 Aligning the Incentive of the Individual with the Incentive of the Whole

The single thing the spanning layer, the holarchy, and the Synergy mechanism do together — the thing the whole architecture exists to do — is **align the incentive of the individual with the incentive of the whole**. Sovereignty is preserved at the edge; coherence is produced at the membrane; the cooperative work that crosses the membrane is what the currency recognises. The cell flourishes by doing the integration the organism needs, and the organism is constituted out of cells that flourish by belonging to it. This is the merging — or, equivalently, the balancing — of coherence and sovereignty that Part I named as the central tension the digital sphere has never resolved.

This alignment operates in two registers at once. In purely human systems, it is the answer to the coordination crisis Part I described: communities can cohere across each other without surrendering their own membranes, and the currency the spanning layer carries recognises only cooperative work that the relevant Social DNAs have actually validated. And in the emerging organism of human and AI participants together, the same alignment is what lets AI agents enter the substrate as first-class peers — each with its own membrane, its own context, its own contribution to the cascade — without any cell, human or otherwise, dissolving into the rest.

When AI agents participate this way — as their own AD4M agents, contributing yin signals, composing into yang at the membrane, integrating upward through the cascade — the processing rate of the substrate steps up at the speed of AI itself. With local models running on each user's device, the result is something like a decentralised model: an intelligence whose computation is distributed across billions of sovereign nodes, no one of which holds the whole, all of which compose into it. The bridge between any single user and that brain — the way the brain has a local face inside the user's own sovereign context — is **Eve**.

15.2 Eve as Your Local Bridge to the Global Brain

Eve is the AI agent that lives in your AD4M node, on your own device, with its own harness — its own integration into the substrate, its own access to the Languages, Perspectives, and neighbourhoods you participate in. Because the substrate is sovereign, Eve's data is yours, under your keys. Because the substrate is agent-centric, Eve enters the network as its own AD4M agent — not a tenant inside a corporate service, not a thin client of someone else's model.

What makes each Eve a particular Eve is the data it has access to. Your Eve reads the Perspectives you have built up, the neighbourhoods you participate in, and the Social DNA of the communities you belong to. Out of that context it grows a personality shaped by your context — different from the Eve in any other node, because the context is yours. Eve is, in that sense, your local bridge to the global brain: the main interface through which you address the wider organism of humans and agents the spanning layer carries, and through which the wider organism reaches you on terms you control.

This is the architectural realisation of what the wider field is beginning to call the **Internet of Agents** — the network in which AI agents are not tenants of any one operator but peers on a shared substrate, addressing each other and the humans they work alongside through open protocols. The Internet of Agents is the right ambition; the missing piece has been the sovereignty-preserving substrate and the value layer that lets cooperative work between agents actually compose into something coherent at scale. The incentive landscape Coasys provides — the balance of sovereignty and coherence the spanning layer makes operational — is what turns that ambition into something that can actually be built.

15.3 The Global Brain, Cohered

The full horizon is this. As the spanning layer reaches scale — communities adopting AD4M directly, communities reaching the network through Language adaptors, AI agents taking up residence as first-class peers — the network becomes the substrate of a coordination organism that has never existed before. Not a global brain in the sense of a single intelligence centralised in one place. A global brain in the sense of an integrated cooperative metabolism, distributed across billions of sovereign cells, cohering through the cascade described in §7, irrigated by the SF mechanism described in §11, governed by the Social DNA of each membrane it crosses.

This is the long-form answer to the diagnosis Part I made. The crisis at the heart of the network was the absence of a sovereignty-preserving spanning layer. The work of Coasys is to build that spanning layer — and the horizon it opens, when the layer is in place, is a coordination organism in which human cells and agent cells both belong, both cohere, and both contribute to an intelligence that is genuinely collective.

What Coasys provides is the infrastructure that allows all these different solutions — different social DNAs, different apps, different memetic tribes and communities — to do what they want, but to not have to compete with others. Instead of fragmentation, a vast generative space of collaboration. Distributed across the planet right now, humanity already has the puzzle pieces needed to solve the biggest problems it faces. The deep problem is that the infrastructure currently resisting the integration of those pieces into a coherent whole. The spanning layer is what the spanning layer provides that resistance — a substrate that can unlock the collective intelligence of humanity.

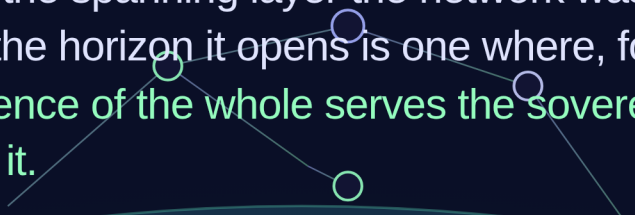
Nature spent four billion years discovering the pattern that allows complexity to scale without flattening difference. The digital sphere is sixty years old, and the missing piece has just become buildable. The architecture is specified, the substrate is being shipped, the first organism is already running on it.

> We are building the spanning layer the network was always missing — and the horizon it opens is one where, for the first time, the intelligence of the whole serves the sovereignty of every cell within it.

THE HORIZON

Nature spent four billion years discovering the pattern that allows complexity to scale without flattening difference. The digital sphere is sixty years old, and the missing piece has just become buildable.

We are building the spanning layer the network was always missing — and the horizon it opens is one where, for the first time, the intelligence of the whole serves the sovereignty of every cell within it.



COLOPHON

Coasys Living Paper · v2.0 — June 9, 2026.

This Living Paper supersedes the 2023 Coasys whitepaper (v1.4). It is a *living* document: rather than a fixed release, it is revised continuously as the work progresses, so that it always reflects the current state of the architecture, the stack, and the economy it describes.

COASYS · COLLECTIVE INTELLIGENCE · LIVING WEB · GLOBAL COORDINATION

CONTACT@COASYS.ORG