

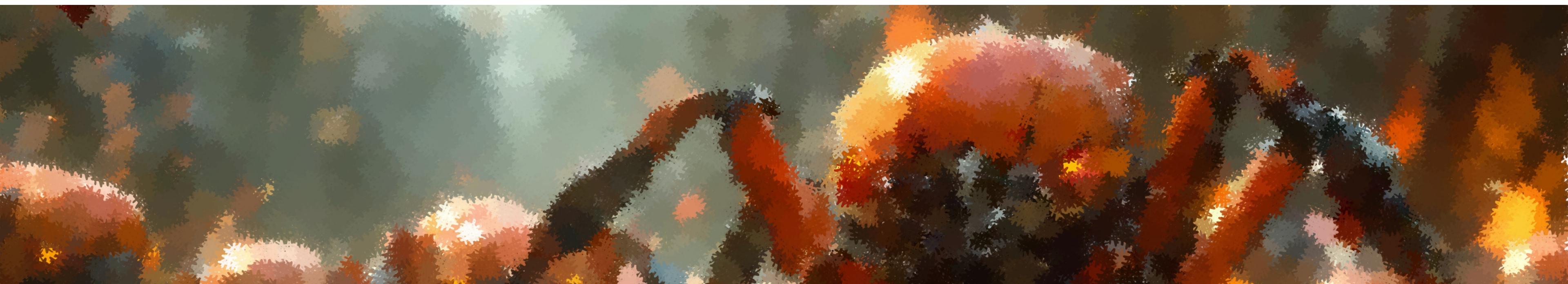
# UNICITY LABS

## off-chain crypto agents

# PROBLEM

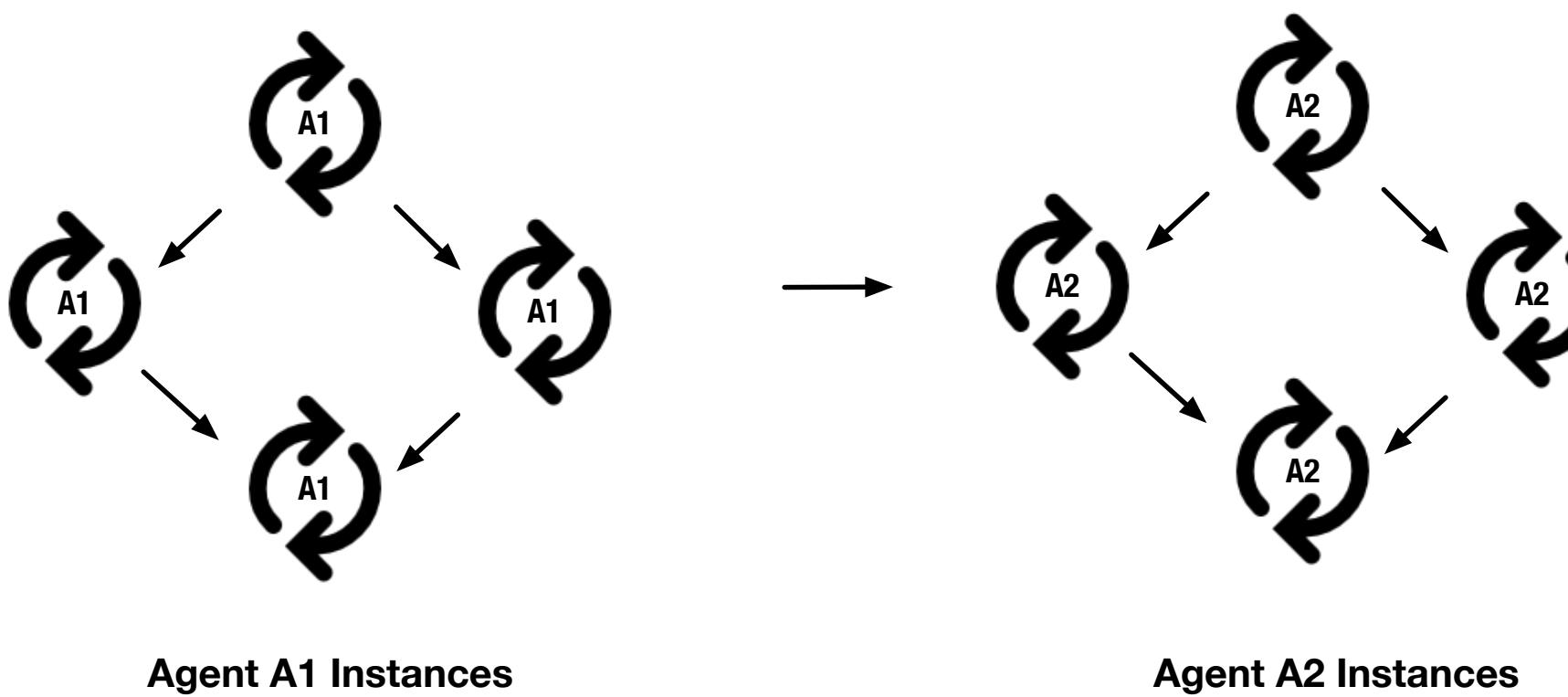
---

- ↗ Current blockchains fail to meet the performance demands of truly global, high-throughput systems.  
No one is running Call of Duty/Fully Self Driving AI models in a smart contract today.
- ↗ Complexity leads to a poor developer experience: smart contract languages, wallets, infrastructure, gas fees etc. raising the bar for mainstream adoption.
- ↗ Now imagine if you could provide an abstraction such that developers could write smart contracts (or "agents") in any language, running in any environment. Agents execute in parallel in local environments with the same verifiability as if they were running on-chain.



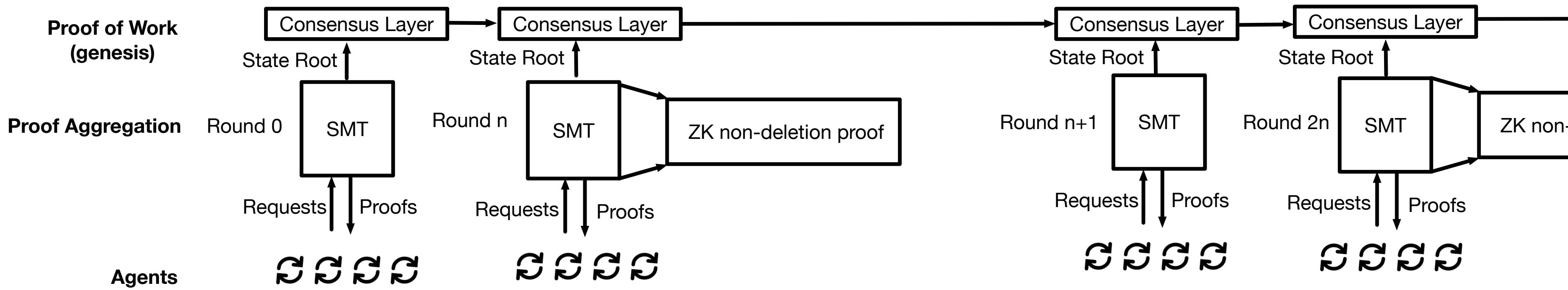
# SOLUTION

- ↗ Unicity is a new blockchain architecture purpose-built for developing and orchestrating off-chain crypto agents



- ↗ An agent is an encapsulation of cryptographically verifiable code and state, e.g. a fungible currency token, an NFT, a smart contract, a game NPC, an AI, or any combination thereof.
- ↗ Agents replicate (interested parties run instances of the agent) and share verifiable state through any medium with other agents.
- ↗ There is no bottleneck as all agent execution happens in parallel off-chain in local environments.

# INVENTION



**DISAGGREGATION** of the proof of uniqueness (non-forking), allowing execution to happen off-chain. When agents need to share state, they request a proof on demand from the Unicity Infrastructure providing a verifiably unique state history (preventing “double spends”)

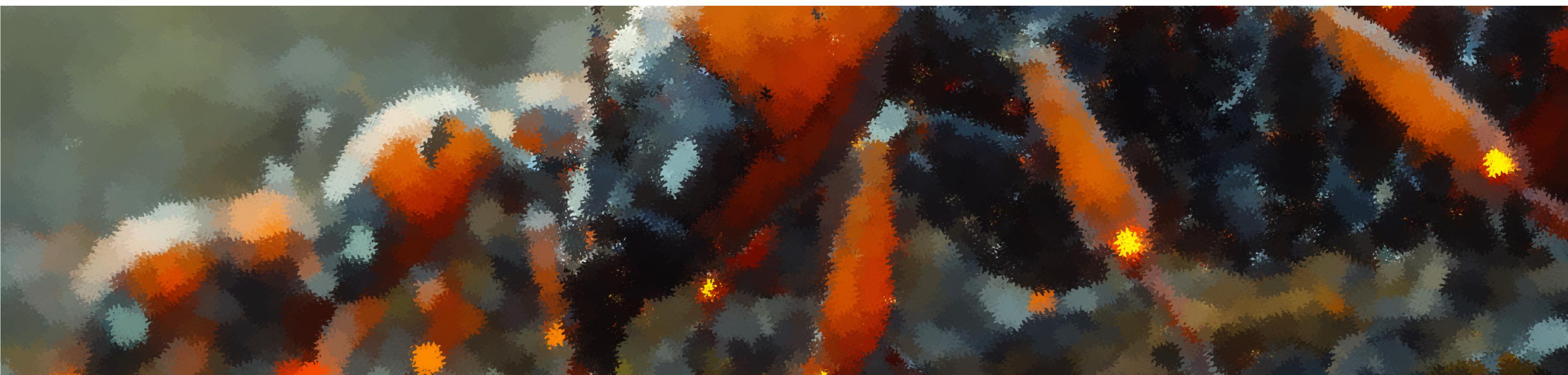
**DECOMPOSITION** of the blockchain. Only single-input UTXOs are allowed in the consensus layer. This ensures decomposition of the ledger as coin histories are independent. Coins are burned, and a burn proof is presented in the agent layer for genesis. No further reference to the blockchain is needed for zero trust verification.

---

# EFFICIENT SHARED STATE

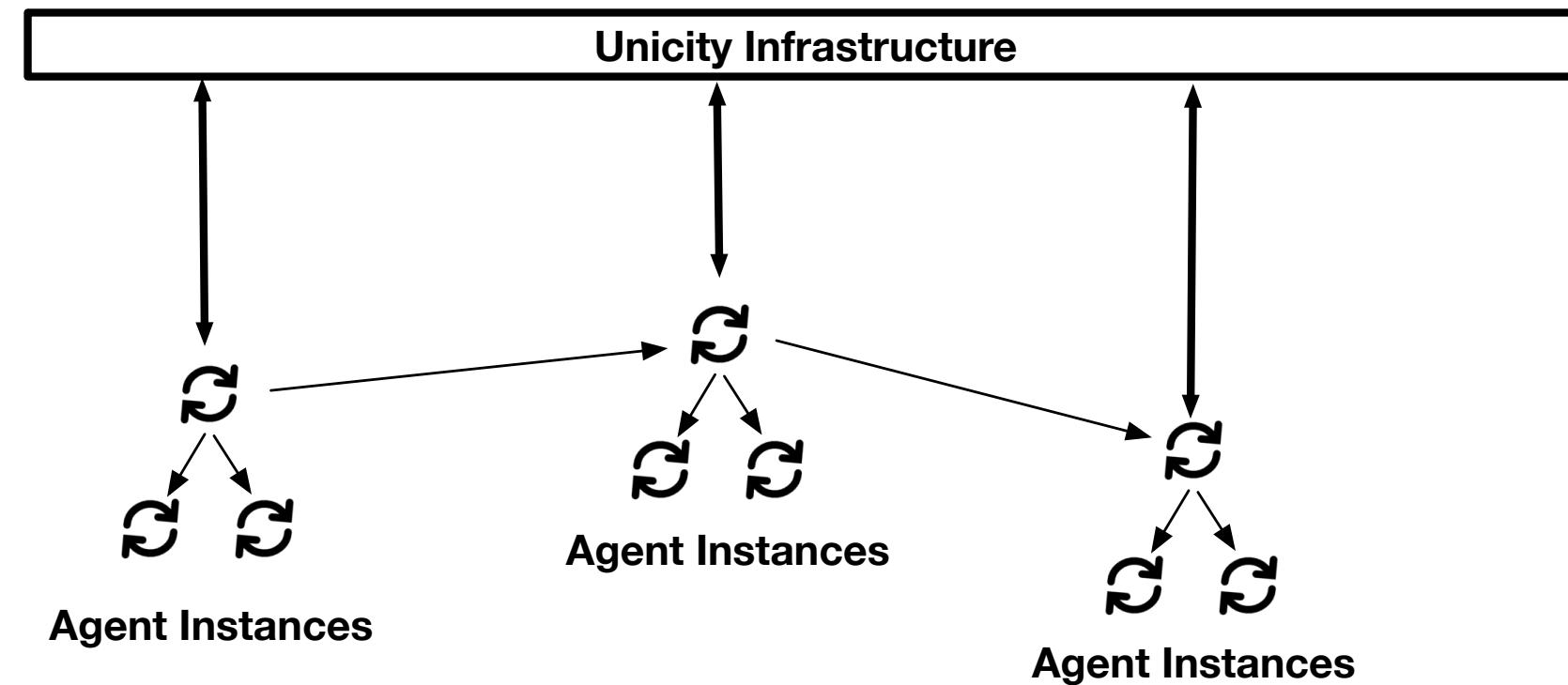
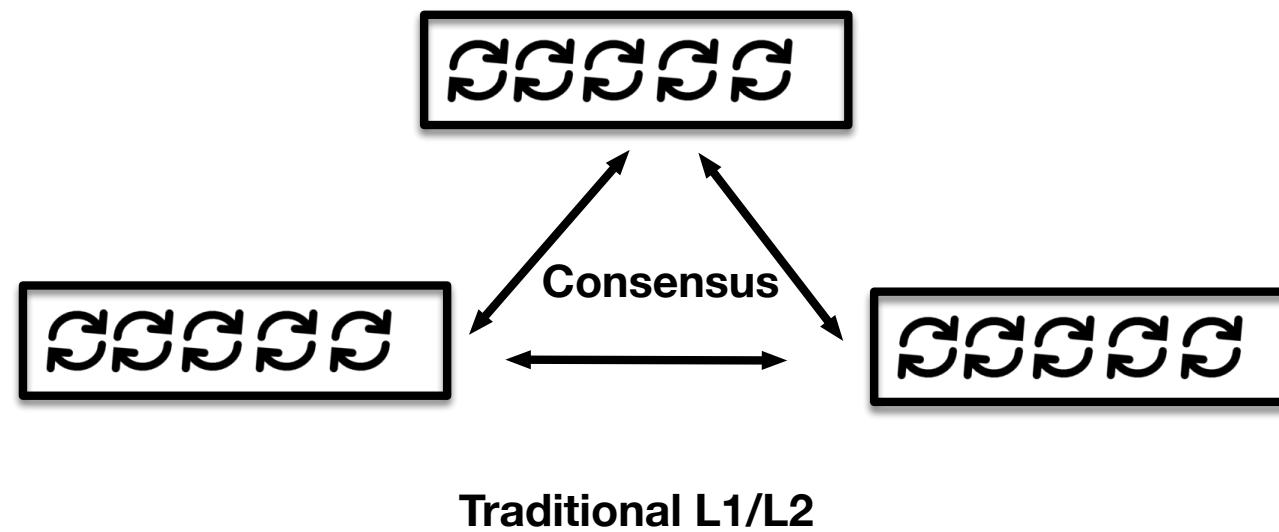
---

- ↗ Unlike traditional blockchains there is no shared *global* state. State is still shared but only amongst interested parties. This achieves censorship resistance, privacy, data availability and unparalleled efficiency. The lesson from the last 10 years is that global ordering and “everyone validating everything” cannot scale.
- ↗ State can be verified off-chain, i.e., like physical cash, state is *locally*verifiable. This is the key innovation and possible due to single-input UTXOs in the consensus layer.



# COMPARISON

- ↗ Unicity remotely resembles a (zero transaction) L1 and a set of dynamically spawned L2s, one per contract (agent). Agent instances perform local transaction ordering, obtain a unicity proof and synchronize with other agent instances. There is always an interested party who is responsible for validation and availability.



- ↗ Unlike L2s, agents are public templates (UAT1...) that define specific tasks (e.g. game NPCs). Similar to Kubernetes, agent instances are spawned on demand, executed and synchronized by interested parties.

# SO WHAT ?

## Native Execution

**Unlimited throughput  
as code executes in  
parallel in local  
environments.**

## Privacy by Default

**Untraceable transactions.  
State is only shared agent  
to agent.**

## 10X UX

**No software no wallets  
just JSON objects.**

## Bridgeless

**Agents move chain2chain  
without bridges**

## ZK Ready

**Use any client side  
ZK prover**

# Decentralized AI

---

Verifiable state transfer and an integrated value transfer system for AI Agents.

# Decentralized Finance

Centralized performance with decentralized security



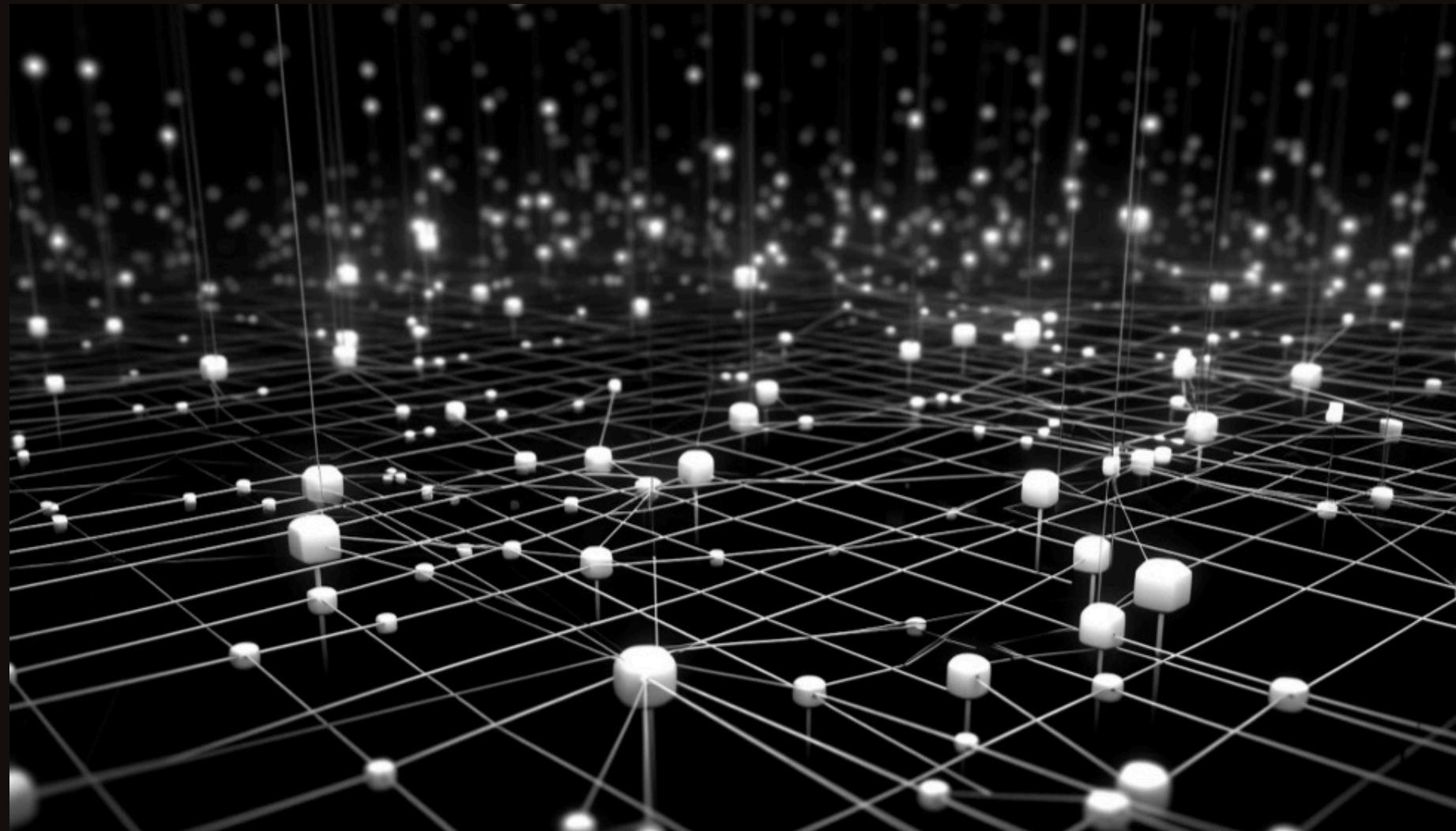
A decentralized exchange can be built up using autonomous agents each operating a component of the exchange (one per trading pair, margin management etc.)

Demo of UNIDEX – a centralized exchange with decentralized settlement.

[unidex.unityc.network](http://unidex.unityc.network)

# Agentic Infrastructure

Infrastructure as Code -> Infrastructure as Agents



If agents are self-contained and local then a (permissionless) distributed machine is not needed for security, freeing the application layer from the huge expense of decentralization.

Think of a decentralized version of Dfinity without the need for Dfinity infrastructure. Execution happens fully on client machines, with autonomous auto-discoverable run-time agent instances.

# MOMP Immersive simulations



Decentralization enables the scaling of massive online multi-player immersive simulations (client-server cannot go beyond a few hundred players in the same shared world).

Truly serverless gaming was the original motivation behind this work.



# TOKENOMICS

**Consensus Layer:**

**BTC tokenomics , 21M coins, 4-year halving. Zero pre-mine.**

**Agent Layer:**

**Subscription-based with fixed pricing within bands.**

**Fees are used to incentivize infrastructure operators.**



---

# RESOURCES

**WHITE PAPER:**

<http://www.unity-labs.com/>

**GITHUB:**

[www.github.com/unitynetwork/](https://www.github.com/unitynetwork/)

**BLOCK EXPLORER:**

[www.unity.network](https://www.unity.network)

---

**THANK  
YOU.**

**contact: unicity—devs@proton.me**