Treasure Network

Persistent Agents for Living Worlds

Treasure DAO

Abstract

Artificial life will make virtual worlds into enduring cultural spaces. People will come initially for compelling inhabitants and a changing environment, but inside this environment they will meet other humans and build cultures and a shared history around this strange place they encountered together. Like cities forming around rivers, these communities will develop around environments worth returning to. A community may want this world to outlast its creators. Its assets may survive on a blockchain, but the developers needed to maintain the world disappear over time. Treasure Network proposes infrastructure for the continuity of these worlds and characters. An agent has an identity represented by an NFT, recoverable private state, a funded service balance, and a host selected from operators called towers. Separating control from hosting allows a funded agent and the world it serves to survive changes in infrastructure and stewardship. The protocol coordinates one current authority and an accepted recovery state; applications define what must be restored. Existing game engines and servers remain usable. The first application of the Treasure Network is a shared social space with persistent AI characters. The network enables a host of other use cases including finance agents, autonomous DeFi mechanisms, private compute, and more. These use cases can be built by outside developers as dapps on top of the network layer.

1 Worlds Worth Returning To

A convincing artificial inhabitant gives a virtual world a life beyond the current session. It can remember visitors, develop relationships and take part in events whose consequences remain when people return. If these inhabitants are compelling enough, some people will visit initially to spend time with them. Others will come for the activities and changes they bring to the world. Human relationships can then form around this continuing life.

Artificial inhabitants help create occasions for human contact, while human activity gives the place a history that neither population could have produced alone.

Cities often form around rivers. People build institutions and cultures around a feature that makes a location useful, even as they reshape that location. A virtual world could develop a similar relationship between its artificial inhabitants and its human community. The characters need to be interesting and continuous enough to give people reasons to gather.

Online games already support this attachment theory. A study of Animal Crossing: New Horizons described how NPCs and their stories helped mediate relationships between human players [1]. AI may deepen this existing pattern by allowing more of the setting to respond to what its participants have done. A character that remembers a community's history can become part of its ongoing story and connect them in a way that no other space can.

In traditional games, communities are transferable. Users coordinate with their guild members using Discord and other tools to find a new game to play. But AI characters and AI-driven experiences have an odd effect of turning digital spaces into something like physical places, where important parts of the world cannot simply be picked up and moved. Research on Faunasphere's closure showed that players could regroup through Facebook and forums while being unhappy about losing access to the virtual creatures they had cared for [2]. We expect this consumer behavior to become even more pronounced as AI enables more immersive experiences than digital pets. Losing one's social world will likely grow from irritating to unacceptable as these experiences become more important to individuals' lives.

The goal here is not like other generative AI products that focus on generating content more cheaply. It is to

make virtual places that can accumulate a social history. We view generative AI's role as enabling lightweight "developer agents" that maintain these simulations and perhaps even create them.

2 What a World Must Preserve

A world that matters to its inhabitants needs a way to continue beyond the team that made it. Its continuing life depends on software updates, paid infrastructure, recoverable data, and admin privileges for core developers.

Server shutdowns make this dependency visible. Ubisoft announced that The Crew would become inaccessible on every platform after its March 31, 2024 server shutdown, citing infrastructure and licensing constraints [3]. A durable ownership record cannot by itself keep a service playable or transfer the rights needed to operate it.

For an AI world, there is more to preserve than a map and an inventory. A resident's relationships with visitors, its memory of shared events, its current commitments, and its place in the world's history may be central to the experience. Restoring a visually identical village whose inhabitants have forgotten everyone would recover the scenery while losing part of the world people knew. Reports of AI-companion loss show why disruption can have emotional consequences, although they do not establish that companion engagement produces shared culture [4].

Continuity does not mean stasis. Characters can learn, and relationships can change. The agent could even iterate on the code base as its world master.

Treasure makes the agent performing this continuing role recoverable across operators. A developer gives it the files, tools, permissions and budget needed for a defined task. If its tower fails, another tower restores the accepted state and assumes current authority. This also requires fencing the old host: recovering memories is insufficient if two copies can both change the world or spend its funds.

The world itself can use ordinary infrastructure. A Three.js application, Unity or Unreal game, external database and dedicated game server can remain independent of the agent's host. The server may keep serving players while its maintenance agent moves between towers. Developers choose which records benefit from a blockchain; they need not put the whole game onchain. They must still arrange recovery for external data and access that their application depends on.

3 A Small Network Core

Treasure separates the running of an AI agent from the owning of an AI agent. The chain is used for agent programmability and staking commitments from node operators. Towers execute agents.

The documented baseline uses Intel TDX confidential virtual machines on Phala Cloud, Rust tower software, libp2p discovery with a Kademlia distributed hash table, and staking on Base. It describes ERC-721 agents with token-bound accounts, tower-mediated inference, periodic Arweave backups and respawning after host failure. These are starting components. The requirements below strengthen that design; they are not a claim that the implementation already satisfies them or has been demonstrated at scale.

The minimum coordination state is small. The existing NFT and account identify the agent and its controller. An assignment records the current tower and an increasing authority epoch, or generation number. An accepted recovery head identifies the state a replacement may restore. Reassignment advances the epoch so that old and new instances of the same agent can be distinguished. Changes in control or execution policy must invalidate stale authority under the same transition rule. Base is the canonical coordination chain in the baseline; independent registries must not each authorize a different current host.

A tower advertises capacity, supplies attestation evidence and posts its required bond. Discovery helps it find peers; discovery does not decide who may act for an agent. A replacement checks the finalized assignment and accepted recovery head before obtaining the permissions it needs. The registry can remain small because it coordinates authority and recovery references rather than storing every conversation or repeating AI inference across nodes.

3.1 Recovery Material

The simplest checkpoint is one encrypted, consistent snapshot with a manifest that identifies the agent, producing epoch, sequence, prior accepted head, runtime and policy versions, encryption-key versions and restore

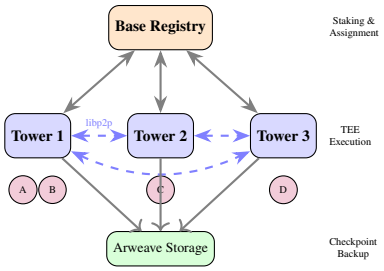


Figure 1: Tower network topology. Towers connect via libp2p/Kademlia DHT, register with Base for staking and agent assignment, and back up checkpoints to Arweave.

dependencies. The head binds this immutable manifest. Larger applications may separate files into encrypted objects and reuse unchanged objects within the same agent. A custom chunk store and mandatory delta chain are unnecessary for the initial system.

A new head is accepted only if its producing epoch and expected prior head still match the canonical record. Before publication, the host verifies that the complete restore set has been stored according to the deployment's retention policy and that required keys are recoverable. A hash detects substitution; it does not prove continuing availability or that the checkpoint is semantically correct. Redundant storage, restore tests and retention of earlier accepted checkpoints address different failure modes. Cleanup must preserve all objects referenced by retained manifests.

Snapshot production, accepted-head publication and archival may have different cadences. A newer snapshot that has not become accepted is speculative. The declared recovery loss window starts at the accepted head. Recovering from an earlier retained checkpoint can help after corruption, but it discards later state and requires reconciliation of any external effects already committed. It must be an explicit authorized recovery transition, not a host silently moving the head backward.

Applications define what their snapshots contain. The shared-world pilot must capture character and relational state consistently with the world it restores. A maintenance agent may instead reference an independently backed-up database or running service. These are application contracts, not reasons to introduce a universal world-state protocol.

3.2 Replaceable Services

Execution, storage and key recovery are modular. Phala can supply confidential execution. ICP can supply selected coordination or application services. Arweave can supply funded archival storage. Phala, ICP and Arweave are possible integrations, not mandatory dependencies. The initial version of the network after launch relies on Lit for key custody and secret recovery. Remote attestation provides evidence about a platform and its measured software. It can support releasing secrets to an approved runtime. Decentralization of key management will come at a later date as it requires much more extensive testing to ensure that secrets are propagated safely between nodes.

State recovery and authority recovery must be designed separately. A checkpoint should contain encrypted state and references to bounded permissions rather than an unrestricted root key that leaves every restored copy equally powerful. A key service releases or transforms material for the newly authorized runtime only after checking attestation and current authority. Proxy re-encryption is one possible component; the deployment still needs a concrete key hierarchy, rotation policy and response to compromised recipients. Scoped smart-account permissions may be simpler for supported actions. Copying a wallet key into a new enclave does not revoke a usable copy held by the former host. Epoch fencing rejects superseded authority. Protection against cloned or rolled-back execution within the current epoch also depends on the runtime, key-release policy and action-recovery guarantees. Fresh attestation and accepted-head ordering alone do not establish that protection.

4 Who Controls an Agent

An agent NFT represents control of a continuing role. Its owner can submit authenticated private instructions and change configuration within the agent's policy. Transferable control is the target, although transfers can remain disabled in the pilot. ERC-721 supplies the ownership interface, an associated account holds funds or acts for the agent, and ERC-1271 lets the account validate signatures from the agent's own signing key; a contract owner, such as a multisig, acts through direct account calls. ERC-6551 offers a token-bound account interface, currently in Review [5–7].

An account can express several forms of control. These forms use the same persistence mechanism; they need not all be implemented before a useful first application exists.

Owner Type	Who Authorizes	Example
Personal account	Account's authorized signer	One person controls a companion or development agent
Multisig account	Configured signer quorum	Studio or group shares control
Governance contract	Executed governance decision	Community controls via contract holding the NFT
Autonomous policy	Predefined policy without human controller	Agent operates within fixed permissions
Supervisor agent	Authorized parent agent	Builder controls child agents via scoped channels

Table 1: Agent ownership types and authorization boundaries

Control messages must be distinguishable from ordinary interaction. A visitor talking to a character supplies input. A verified controller authorizing a deployment exercises privileged authority. An encrypted owner channel protects one route for instructions; it does not make retrieved documents trustworthy or eliminate indirect prompt injection. A supervisor agent likewise gains no unlimited authority merely because it is an AI.

Ownership transfer is a security event. The new controller must become the only source of new privileged instructions after the transfer becomes authoritative. The transition must explicitly handle queued commands, active sessions and delegated permissions. Channel keys should rotate; external credentials may need rotation too. Transfer cannot make a previous owner forget information already disclosed, nor does owning a character automatically authorize disclosure of visitors' private information. Account rules must also prevent parent-child ownership cycles that make control inaccessible.

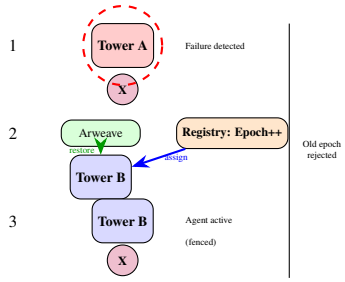


Figure 2: Agent migration between towers. (1) Tower failure detected via liveness challenge. (2) Registry increments epoch, new tower retrieves checkpoint. (3) Agent resumes on Tower B; old epoch requests are rejected.

5 Recovery and Accepted Actions

A missed heartbeat can mean a crash or a network partition. Starting a replacement does not ensure the original has stopped. Both may remember the same identity and believe they should act. The protocol must therefore preserve one current authority even when several processes physically exist. Destinations enforce that authority by rejecting stale requests, a requirement familiar from distributed systems [8].

5.1 From Recovery to Active Execution

An assigned tower begins in Recovering. It checks finalized control state, retrieves and validates the accepted restore set, and obtains epoch-bound permissions. It becomes Active only after required restoration, policy checks, destination fencing and reconciliation are complete. Incrementing an epoch alone does not make a replacement safe to use.

A smart account or contract can check canonical authority directly. A service gateway must establish current finalized authority, not merely remember the largest epoch it has happened to observe. Required protected destinations must reject new requests from the former epoch before the replacement dispatches conflicting work. Already admitted calls must drain or be reconciled. Leases need bounded validity and explicit clock and finality assumptions. Where these conditions cannot be established, the affected capability stays paused.

This guarantee has a declared scope. An independently hosted game can keep serving players

while its maintenance agent recovers. A character whose world server enforces current authority can resume there after fencing. A third-party service accepting an unrevoked bearer key cannot inherit the same guarantee merely because its key was copied to the replacement. It needs provider-side revocation or a controlled gateway; otherwise its integration is best effort with narrow permissions and explicit risk limits.

5.2 Application Actions

The network need not publish a blockchain transaction for every conversation or animation. Applications choose checkpoint cadence and disclose what recent state may be lost. Consequential external effects, such as payments and deployments, require stronger recovery behavior at their adapters.

Before dispatching such an action, the adapter durably binds the originating request or intent to a stable operation identifier and immutable payload. That binding must survive the failure being recovered from, including replay from an older retained snapshot; the new host must not assign a fresh identifier to the same work and execute it again. The adapter also retains the result or enough evidence to determine the outcome. Ordinary recoverable application state or existing destination and account facilities may satisfy this obligation; no Base transaction is required for every effect. An accepted checkpoint is another option. A separate durable journal, if needed, must specify its own access, fencing and recovery guarantees.

Where a destination supports idempotency, retrying the same operation identifier must not repeat its effect. Status lookup and reconciliation handle an interrupted response. Where an outcome remains uncertain and the destination supplies no safe deduplication or status path, the adapter pauses rather than blindly retrying. Retention must cover the declared replay horizon. A transaction nonce alone does not identify the underlying business intent [9]. Unsupported payments and other irreversible integrations remain outside the pilot.

5.3 Failure and Reassignment

In the reference design, any watcher can post a bonded challenge of 1,000 MAGIC, and the tower must answer with a signed liveness proof within one hour. These are observation cadences, not a one-minute recovery guarantee; adjudication, finality, resource availability and restoration add delay.

Authorized reassignment advances the epoch and fixes the recovery head the replacement may use. This transition must serialize with transfers, policy changes and checkpoint updates. A late checkpoint from the previous epoch cannot overwrite the selected head. Commands from a former controller must not be admitted as new privileged work after the control boundary.

A cooperative migration can drain work and publish a final checkpoint before transferring authority. Crash recovery cannot assume cooperation. Both use the same epoch and recovery-head rules. An emergency pause likewise needs a precise scope and a resumption rule that advances authority or cancels stale commands, so that lifting the pause does not revive old requests.

5.4 Suspension and Return

When service funding approaches exhaustion, an agent should stop new costly work, notify its controller and preserve a recovery point within its remaining budget. Archive retention has separate funding and expiry terms. Retaining a checkpoint does not imply that the network is still paying to run the agent.

Resumption requires retained data, recoverable secrets or replacement permissions, compatible software and access to required resources. Domains, databases, model availability, cloud accounts and asset licenses belong in the recovery plan. A license may not transfer with an NFT. Persistence cannot guarantee an eternal subsidy, continued access to every provider, or the right to revive another party's software.

6 Payment and Network Incentives

Agents buy services in USDC. MAGIC supplies the operator bond and network incentive mechanism. This separates the denomination used to budget computing costs from the asset used to coordinate participation. A dapp need not expose MAGIC to its users or make it the currency of its internal economy.

Each agent has a funded credit balance. In the reference design, only its assigned tower may bill for services. The billing authority must also bind the current epoch, usage period and service limits, preventing a former assignment from continuing to charge. Metering

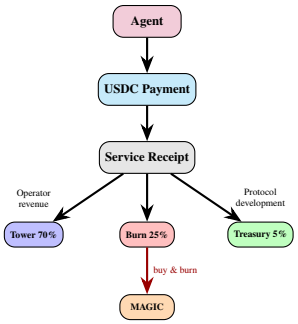


Figure 3: Payment flow and revenue split. Agents pay in USDC; receipts split 70% to tower operators, 25% to MAGIC buy-and-burn, and 5% to protocol treasury.

and settlement require independently checkable records and a defined dispute path; an agent's output alone is not evidence of the work billed.

6.1 Service Receipts and Costs

A 5% protocol fee is taken from each service receipt, so CPU receipts go 70% to the tower, 25% to the purchase and burning of MAGIC and 5% to the treasury. A CPU service intended to cover its own cost must be priced so the tower's 70% share covers that cost. Otherwise, the subsidy must come from other service margin or issuance.

GPU service receipts pass the provider's cost to the tower in full; any markup above that cost is split like CPU receipts. Where the tower mediates calls to an external inference provider, those receipts include the amount needed to pay that provider. The tower's margin is what remains after provider charges and the cost of mediation. Gross GPU receipts are not profit. If confidential inference is offered, the price must cover the infrastructure actually required to provide it.

Inflation supports participation during bootstrapping and periods of low utilization. It does not make an unprofitable service permanently sustainable. As paid usage grows, service contribution should carry more of the cost of operating towers. This is an economic objective to test with real workloads, not a demonstrated break-even result.

6.2 MAGIC Participation

The reference tower bond is 200,000 MAGIC and one Legions Genesis NFT. Unstaking requires 14 days. Registration includes attestation, renewed every 7 days. The Legions requirement is an initial participation condition; it does not establish a permanent entitlement to revenue or an immutable requirement of all future versions.

The ideal annual inflation schedule is 5% in Year 1, 4% in Year 2, 3% in Year 3 and 2.5% from Year 4 onward. The target staking rate is 30%. Rewards are distributed per heartbeat and adjust with the actual staking rate: the proposed policy increases incentives below the target and reduces them above it. The stated variable inflation range is 1.44% to 6.12%.

These parameters define the intended policy. A deployable controller must additionally specify the staking denominator, measurement window, response curve, limits and transition from the initial schedule.

More CPU usage creates more budget for purchasing and burning MAGIC. Whether supply contracts depends on how many tokens are actually burned relative to issuance over the same period. Token price, utilization, service mix and exchange execution all affect that result. Tower growth alone does not establish deflation or a stable token price.

6.3 Availability and Accountable Faults

A tower that fails a liveness challenge is deactivated, jailed for 24 hours and forfeits the emissions accrued since the challenge, without losing its stake for downtime alone. This separates failure to provide availability from attributable protocol fraud.

The retained penalty parameter for malformed evidence is 10% of tower stake. A safe implementation must define the offending evidence precisely, authenticate its source and apply a reproducible verification and adjudication process. An arbitrary malformed packet cannot justify slashing the operator whose name it contains. The recommended pilot defers irreversible slashing until attribution and challenge handling have been demonstrated; reversible escrow is an alternative to evaluate. The challenge committee begins with 3 seats and can scale to 21, but quorum, selection, conflicts and appeal rules require specification before deployment.

Penalties should address conduct the protocol can establish, such as authenticated invalid evidence or a defined violation of assignment rules. Whether an NPC was entertaining, a model answered well or a character

remained psychologically consistent is not an objective slashing proof.

7 Trust Assumptions

The network's security depends on several systems remaining sound together. The chain must preserve its finalized registry. Attestation verification must accept only approved runtime measurements and platform states. Recovery storage and key services must retain available material. Destinations must enforce current authority. The inference and tool path must respect the declared confidentiality and permission boundaries.

A trusted execution environment reduces the host operator's access to agent secrets under its threat model. It does not remove hardware vulnerabilities, denial of service, malicious inputs or mistakes in approved code. It also cannot stop an agent from revealing a secret through an allowed output. Least-privilege tools, explicit spending policies and separation of owner commands from untrusted content remain necessary. Launch requires consistent attestation and approved-code checks on normal and recovery paths, a defined upgrade grace period, and isolation of each agent's files, secrets, permissions and charges.

Running private agents creates a policy problem: users need confidentiality, while the network must be able to restrict abusive behavior. The initial boundary should be implemented in verifiable runtime controls wherever possible. An agent receives specific tool and resource capabilities, rather than general authority to perform anything the model proposes. High-risk actions can require approval by the controller or a defined review process.

The Legions NFT may develop a further role as constitutional guardian agents. In this proposed extension, guardians interpret a published policy, block or escalate certain actions and, under defined authority, suspend or terminate an agent's execution. Such a role needs a clear separation between observation, judgment and enforcement. It also needs appeal procedures, conflict handling and limits on what private information a reviewer receives. A guardian's explanation is evidence of its decision, not a cryptographic proof that the decision is correct.

This extension should follow, rather than precede, a working persistence protocol. Legions initially serve as a staking requirement. Subjective model judgments should not silently become the basis for taking a tower's bond. The deployment must identify who can change policy, upgrade software, pause execution and restore it. Emergency authority can censor or interrupt agents and must have stated limits and review. A pause affects only paths that enforce it and cannot recall external actions already accepted.

The draft ERC-8004 standard provides identity, reputation and validation interfaces that can make agents easier to discover and evaluate across applications [10]. Its specification explicitly separates registration from guarantees about an agent's advertised capabilities. These interfaces do not replace execution safety. A registry entry cannot prove that an agent's action is safe. zkLLM does not establish honest reasoning [11]. Experiments on reasoning models also show that their written explanations can omit influences on their answers [12].

Therefore, agent governance requires enforcement over agents' behavior, not thinking. The Legions Genesis NFT, once it becomes an AI agent itself, can enforce DAO-defined policies on the behavior of non-deterministic agents to prevent rogue agents from spawning on the network.

8 Use Cases

The same infrastructure that preserves virtual worlds can host any persistent, recoverable agent. Because the network separates agent identity from execution and provides confidential compute, it functions as a general-purpose dapp layer for autonomous software. Any application requiring agents that outlive their creators, maintain private state, or act on users' behalf can deploy here.

- Tokenized finance agents
- LLM-powered oracles
- Prediction market resolution
- Protocol-owned market makers to eliminate frontrunning
- Autonomous governance delegates
- Cross-chain arbitrage and yield optimization
- Liquidation protection for lending positions
- Autonomous treasury diversification for DAOs
- Privacy-preserving identity verification agents

Treasure Network is not the only agent runner. Protocols such as Autonolas, Phala, and others pursue related goals. We believe that winning networks will gain a foothold in real, paying markets to bootstrap node participation independent of emissions. This network is designed as a positive feedback loop: more dapps create more agent usage, which generates higher USDC margins for towers. Higher revenue leads to more staking. The network can then serve more users and experience a true network effect where properties like privacy become exponentially more valuable than on rival networks (onion routing between a thousand nodes as opposed to ten). The DAO can vote to lower the MAGIC staking threshold if the price and adoption of the network increase significantly.

At Treasure, we are focused on consumer markets—vibe-coded gaming, AI companions, private compute—because users are already expressing pain points in these markets around AI impermanence, platform centralization, and high individual compute costs. Consumer markets are highly profitable and sit furthest out on the risk curve for autonomous AI. Every great technology begins as a toy for this reason. Establishing network profitability through these early adopters creates the networking layer to enable more complex financial and enterprise use cases as they become feasible.

9 Conclusion

The reason to preserve a virtual world begins with what people may build inside it. Artificial inhabitants could become the centers of gathering places whose customs, relationships and history develop over years. The human community and the artificial world would then sustain each other's value. Continuity becomes important because the place has something worth continuing.

Treasure proposes a compact mechanism for that continuity: an agent identity whose controller can change, a recoverable state and one enforceable current authority across hosts. The application defines the world that state must restore. Towers supply execution, USDC pays for services, and MAGIC bonds participation and supports network incentives. Developers retain ordinary engines and servers, while communities gain a path to changing their infrastructure without abandoning their history.

The 2021 founders' paper imagined MAGIC as a natural resource and NFTs as instruments for governing a network of worlds [13]. The game-console essay later argued that "Smart contracts outlive their engineers" [14]. Persistent agents extend that ambition from surviving records to continuing activity. In a future version, a Legion could help interpret a world's rules, with bounded authority over the real machines that sustain it. Fantasy would shape physical infrastructure through an explicit, inspectable mechanism.

A Reference Parameters and Proof Obligations

A.1 Reference Parameters

Parameter	Value	Interpretation
Tower bond	200,000 MAGIC + 1 Legions NFT	Initial operator requirement
Unstaking delay	14 days	Withdrawal delay
Attestation renewal	Every 7 days	Registration maintenance
Liveness challenge	1,000 MAGIC bond, 1-hour window	Response window, not recovery time
Challenge committee	3–21 seats	Quorum TBD
Evidence penalty	10% of stake	Requires defined adjudication
CPU receipts	70% tower, 25% burn, 5% treasury	Applied to service receipts
GPU receipts	Provider cost to tower; markup split same	Provider costs paid from receipts
Ideal inflation	5%, 4%, 3%, then 2.5%	Years 1, 2, 3, and 4+
Target staking	30%	Controller needs specification
Variable inflation	1.44% to 6.12%	Transition rule TBD

Table 2: Reference network parameters

The earlier reference design used a 250 MB agent workspace, an estimate of approximately 150 agents per tower and backups every 120 seconds. These are sizing and cadence assumptions, not validated capacity or recovery guarantees. The proposed mutable checkpoint path separates backup production from canonical publication and optional archival. Rewriting a full 250 MB workspace every 120 seconds would

produce 5.4 TB per agent in a 30-day month before replication. Reusing unchanged encrypted objects and snapshotting compact changing state can reduce that load; the workspace cap is not a measure of bytes changed per checkpoint. Capacity must be measured against actual workloads and hardware.

References

[1] X. Tong et al., "Players' Stories and Secrets in Animal Crossing: New Horizons," *Proc. ACM Hum.-Comput. Interact.*, CHI PLAY, 2021.

[2] M. Consalvo and J. Begy, "What Happens at the End of the World? An MMOG's Closure and Player Responses," *6th Int. Conf. Philosophy of Computer Games*, 2012.

[3] Ubisoft Ivory Tower, "An Update on The Crew," Dec. 14, 2023.

[4] J. Banks, "Deletion, departure, death: Experiences of AI companion loss," *J. Social and Personal Relationships*, 2024.

[5] W. Entriken et al., "ERC-721: Non-Fungible Token Standard," Final.

[6] J. Windle et al., "ERC-6551: Non-fungible Token Bound Accounts," Review.

[7] F. Giordano et al., "ERC-1271: Standard Signature Validation Method for Contracts," Final.

[8] M. Burrows, "The Chubby lock service for loosely coupled distributed systems," *OSDI*, 2006.

[9] M. Featonby, "Making retries safe with idempotent APIs," Amazon Builders' Library.

[10] M. De Rossi et al., "ERC-8004: Trustless Agents," Draft.

[11] H. Sun, J. Li, and H. Zhang, "zkLLM: Zero Knowledge Proofs for Large Language Models," *ACM CCS*, 2024.

[12] Y. Chen et al., "Reasoning Models Don't Always Say What They Think," Research paper, 2025.

[13] Treasure, "Founders' Long-Term Vision for Treasure Project," Nov. 3, 2021.

[14] Treasure DAO, "Treasure: the Web3 Game Console," Jul. 3, 2022.