Autonomys: The Radical Autonomy Network for Humans and AI

Chen Feng[1,2], Labhesh Patel[1], Dariia Porechna[3], Jeremiah Wagstaff[1,2]
[1]Autonomys Labs
[2]University of British Columbia
[3]Subspace Foundation

***Abstract*—As we approach the widespread integration of artificial intelligence (AI) into our everyday lives, we face a pivotal moment of redefinition for our traditional understanding of the role of human and machine agency in contemporary society. The rapid progression of AI technologies presents both significant challenges to prevailing socioeconomic structures and theory, and unprecedented opportunities for the establishment of novel paradigms. While some propose universal basic income (UBI) as a solution to potential mass job displacement, Autonomys offers an alternative vision—one that maintains humanity's economic relevance and enhances human autonomy in an AI-augmented world. We propose the Autonomys Network—a decentralized infrastructure stack for secure, sovereign collaboration between human and artificial intelligence (H+AI). We have built the Autonomys Network from first principles to simultaneously achieve security, scalability, and decentralization based on original multi-year research. At its core, the Autonomys Network implements Subspace, a novel storage-based consensus protocol that decouples consensus from execution. This proposer-builder separation allows the Autonomys Network to independently scale transaction throughput and storage requirements while maintaining a fully decentralized blockchain with a low barrier to participation—all vital for the realization of decentralized AI (deAI)—or AI3.0.**

I. Background

The rapid advancement of artificial intelligence (AI) technologies is ushering in a new era of economic and social transformation. Recent breakthroughs in machine learning (ML), particularly in deep learning and natural language processing, have led to AI systems capable of performing tasks once thought to be the exclusive domain of human intelligence [1]. This progress has sparked debates about the future of work, with some experts predicting widespread job displacement [2], and others envisioning new forms of human-AI collaboration [3]. Concurrently, the rise of blockchain technology has introduced novel paradigms for decentralized systems and digital identity [4]. These technological developments have occurred against a backdrop of growing concerns about data privacy, algorithmic bias, and the concentration of AI capabilities in the hands of a few large corporations [5]. As we approach the potential development of Artificial General Intelligence (AGI) and Artificial Superintelligence (ASI)—when AI reach and then exceed human intelligence [6]—it becomes imperative to establish frameworks that ensure AI systems align with human values and preserve individual agency in an increasingly automated world. We must not lose sight of the importance of human autonomy. [7]. Autonomys proposes a new paradigm of *radical autonomy*, or absolute digital self-governance, via our ecosystem. The Autonomys Network supports verifiable interactions between humans and AI, fosters individual and collective growth, and lays the groundwork for a future in which technology enhances human autonomy.

The authors are listed in alphabetical order.

II. Age of Autonomy

Throughout the history of technological development, humanity has consistently striven for the same fundamental needs to be fulfilled: *safety*, including both physical and resource security; *connection*, be it physical or emotional, to fellow humans or cultural collectives; and *prosperity*, through self-improvement and socioeconomic development. Many experts have discussed AI's impact on human safety, connection and prosperity [3] [7] [8]. Autonomys is using these three fundamental human desires as guiding principles to propose a human-centric vision of a post-AI-revolution future.

In today's world, one's safety and prosperity are mediated largely by one's access to economic resources. As job security becomes progressively more threatened by the advent of sophisticated AI [2], we should evolve our contemporary economic systems with continued human relevance and agency in mind. This can be achieved through widening global access to permissionless, decentralized incentivized contribution networks, and by augmenting human capabilities with verifiable, decentralized on-chain AI agents (*dAgents*).

The trajectory of AI development is trending towards the training and running of smaller, specialized AI models on personal edge devices. When integrated into every action taken on a personal device [9], these AI will have the context of all knowledge about the device's owner, including past and present interactions with other people or services; personal preferences in entertainment, food, clothing and partners; health metrics; financial statements; political allegiances; and everything else that has ever gone through the device. Coupling access to complete contextual data with agentic capabilities transforms personal AI into personal agents that can represent you online and act on your behalf—booking medical visits and vacations, ordering groceries, coordinating

meetings, managing money, or participating in governance. Crucially, personal agents will be able to analyze and filter the endless information streams around us—insurmountable for a single person to process—aiding in our decision-making.

It is prudent to estimate the emergence of at least as many dAgents online as there are smartphones. Practically, we can expect each person and business to have multiple specialized AI representatives. This global mesh of billions of dAgents will communicate and exchange funds with each other online and with service providers via agent-specific permissioned actions. Humans and dAgents will need to be able to verify whether the AI they are interacting with within this autonomous dAgent economy are truthfully representing themselves as agents of particular individuals or organizations. This necessitates a decentralized system of identity and provenance.

Autonomys' secure protocol for the provision of self-sovereign, decentralized digital identities, Autonomys Identity (Auto ID)—the first primitive built on top of the Autonomys Network—is thus key to the Age of Autonomy. Auto ID enables individuals to verify their humanity without resorting to invasive biometric procedures, while simultaneously allowing for the permissioning of dAgent actions, and the authentication of AI-generated content. This foundational layer of digital trust is crucial for facilitating meaningful collaboration between human and artificial intelligence (H+AI) within the economic sphere.

At the core of this autonomous economic system is the concept of sovereign data ownership, enabled by the coupling of Auto ID and content provenance. This revolutionary approach to understanding and managing personal information and intellectual property—where individuals retain control over their data assets and can opt to monetize them for AI training and optimization purposes—pioneers a novel economic model where humans may choose to share their personal data and receive fair compensation for the value their data provides in enhancing AI systems, rather than having their information exploited without remuneration, as is currently the case. The Autonomys economy also addresses concerns regarding AI safety and alignment. By implementing a decentralized governance framework, Autonomys enables collective decision-making on AI development and deployment. This democratization of AI access and control helps ensure that these powerful technologies remain congruent with human values and interests.

The Autonomys-facilitated dAgent economy will foster a rich ecosystem of H+AI collaboration. Our development platform will provide cutting-edge tools for individuals and organizations to train and deploy dAgents, acquire highly valuable technological skills, and amplify their potential. dAgents will be able to exchange mutual authorizations via blockchain to provide real-world goods and services, while humans maintain oversight of and control over their dAgents. This dynamic creates new avenues for entrepreneurship and value generation as individuals and organizations leverage AI capabilities to augment their own skills and offerings. Autonomys' vision

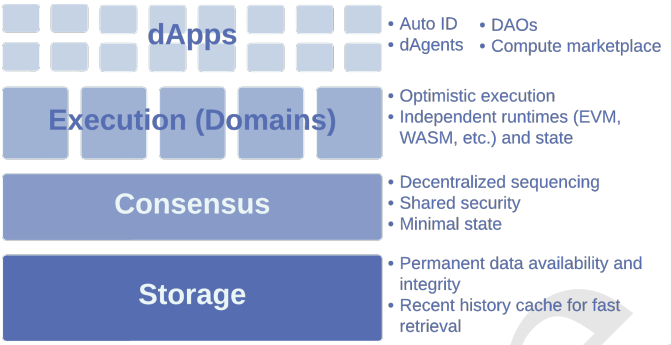

Fig. 1. Autonomys Network stack

preserves human economic relevance by emphasizing domains where the unique human capacity for creativity, emotional intelligence, and complex problem-solving has not been replicated by AI.

In contrast to predicted futures populated by a universal basic income (UBI)-dependent humanity [10] subject to a diminished human agency, Autonomys champions radical autonomy through incentivized participation and contribution to a self-sustaining ecosystem, inspired by Ethereum's pioneering model. We recognize human potential as a dynamic force that can be continually expanded through education, technological integration, and innovative socioeconomic structures. Focusing on continuous learning and adaptation ensures humans remain active contributors to the global economy rather than becoming passive recipients.

By empowering individuals with self-sovereign digital identities, control over their data assets, and tools for safe AI collaboration, we can forge a future where humans and AI coexist productively and harmoniously. This new paradigm not only preserves human economic relevance but amplifies our collective potential, ushering in an era of unprecedented innovation, creativity, and shared prosperity.

III. Autonomys AI3.0 Stack

The Autonomys Network serves as the technological infrastructure for this paradigm shift, providing a verticalized decentralized AI (deAI) stack encompassing (Fig. 1):

- ***dApp/dAgent Layer***: deploying and facilitating the development of *super dApps* (AI-powered dApps) and *dAgents* (on-chain agents), integrated with Auto ID to enable secure and verifiable interactions.
- ***Execution/Domain Layer***: secure, scalable computation for AI training, inference and agentic workflows via distributed compute.
- ***Consensus Layer***: verifiable decentralized sequencing and validation of transactions for shared security.
- ***Storage Layer***: distributed storage ensures data integrity and permanent availability—crucial for storing vast amounts of AI data.

Utilizing the Subspace Protocol, [11] with its innovative Proof-of-Archival-Storage (PoAS) consensus mechanism, [12] our decentralized physical infrastructure network (DePIN)

incentivizes active participation through the permissionless contribution of any amount of storage space or compute, or staking of any amount of tokens, permitting unprecedented accessibility.

Ultimately, Autonomys is dedicated to building the Internet of Agents (IoA) [13]—where every human has many powerful, automatically interoperable, personally aligned models at their disposal. According to Crapis (2024), the three main steps on the path towards the IoA are super dApps, new agent coordination mechanisms, and ownership and governance. The Autonomys Network provides the infrastructure to achieve all three in a decentralized, scalable and verifiable way on-chain, adhering to the outlined vision roadmap:

Auto ID → AI Data Provenance → Verifiable Super dApps → Verifiable dAgents → Internet of dAgents (IodA) → Age of Autonomy.

A. Auto ID: Self-Sovereign Identity Infrastructure

Autonomous agents will operate independently and on behalf of human entities. This paradigm shift necessitates robust mechanisms for accountability. Public key infrastructure (PKI) presents a natural solution, as it is built fundamentally on chains of trust, which establish a verifiable and transparent lineage of trust relationships, ensuring each entity in the chain can be held accountable for their actions and any breaches can be easily traced. We propose the development of an enhanced PKI system, augmented with additional identity mechanisms, to facilitate the transition to a secure era of human-agent collaboration. Our proposed Autonomys PKI enables individuals and organizations to self-issue Auto IDs—any Autonomys dAgents they then build derive their dAgent Auto IDs from their creators' Auto IDs. Autonomys Identity (Auto ID) utilizes advanced cryptographic techniques to establish a robust, self-sovereign identity framework. Entities are currently expected to register their identities via self-issuing X.509 certificates. Auto ID is a registered runtime on Autonomys' decoupled execution layer.

Key properties of our system include:

- *Self-sovereignty*: Users maintain complete control over their digital identity, with autonomy in information-sharing decisions via a combination of encryption, zero-knowledge and verifiable credentials.
- *Traceability*: Blockchain tech enables the tracking of dAgent actions and decisions, supporting auditability and safety in AI development, deployment and alignment.
- *Verifiability*: Cryptographic proofs ensure the authenticity of claims without compromising personal information.
- *Universality*: Auto IDs can be issued to any entity, human or artificial, enabling a common identity standard across the digital ecosystem.
- *Interoperability*: Auto ID is designed for seamless integration with existing identity systems such as X.509 [14] and Decentralized Identifiers [15].

Our system provides a secure and transparent mechanism for authorizing the actions of dAgents within the Autonomys ecosystem via the granting and revoking of permissions. This ensures that these AI systems operate within the bounds of their predetermined roles. Permissioned delegation of authority is crucial in a world where digital employee and personal assistant dAgents make important decisions and perform vital tasks for both organizations and individuals. After obtaining an Auto ID, entities can digitally sign the content they produce and actions they undertake, establishing a verifiable and tamper-proof record of authenticity and provenance for both data and deed, linked with their Auto ID. This is particularly important as the line between human-created and machine-generated content becomes increasingly blurred.

Autonomys also offers the ability to attach cryptographic identity claims to an Auto ID via verifiable credentials. For example, an individual may attach a verifiable credential to their Auto ID showing they have a valid diploma, and later utilize that claim when a diploma is required. This example and a general framework for verifiable credentials is described in Claims Framework. The Auto ID framework supports self-issuance of an identity, issuance by another entity, and co-issuance by multiple entities.

In summary, by enabling users to delegate authority to AI agents; trace the lineage and behavior of agentic systems for safety and regulatory compliance; maintain accountability in digital interactions; and authenticate AI-generated content, Auto ID facilitates more secure interaction between humans and AI agents, establishing a foundation of trust crucial for the autonomous machine economy.

B. Auto Score: Probabilistic Proof-of-Personhood

Auto ID implements proof-of-personhood (PoP) via our composable, probabilistic PoP protocol Auto Score. Auto Score leverages preexisting evidence of personhood and zero-knowledge proofs (ZKPs) to offer a probability-of-personhood score. Supporting ZKP-secured e-passport verification as a personhood factor, users need only scan the NFC chip in their passport and prove the correctness of the signature in a ZK-proof to achieve a high Auto Score.

For applications that do not require government-grade identification, and users who do not possess or want to associate one with their Auto ID, Auto Score accepts alternative verifiable credentials as personhood factors. These include government-issued documents, credit cards, social media accounts, and participation in decentralized networks. ZK-passport tech presents the strongest evidence of unique personhood, particularly when combined with liveness checks, and contributes to the highest possible Auto Score. It also allows users to selectively share specific details, such as age or nationality, with the verifier, without revealing other information.

As a probabilistic PoP protocol, Auto Score functions by aggregating and evaluating various pieces of evidence supporting an entity's claim to personhood. Each piece of evidence is assigned a weight based on its reliability and difficulty to forge. This evidence is shared utilizing ZKPs, allowing users to prove their possession of credentials without revealing the underlying data. Autonomys then calculates a composite score

representing the probability of the user being a unique human using these weighted pieces of evidence. This score updates as users add or remove credentials, or as their digital interactions evolve, providing a dynamic measure of digital personhood.

Auto Score possesses the following characteristics:

- *Probabilisticity*: Provides a nuanced measure of personhood rather than a binary determination.
- *Privacy-preservation*: Leverages ZKPs and advanced cryptographic techniques to enable users to demonstrate their personhood without revealing sensitive personally identifiable information (PII).
- *Dynamism*: An entity's personhood probability score updates as they interact with the Autonomys ecosystem, reflecting their ongoing participation and contribution.
- *Composability*: Entities can build their digital identity incrementally by combining various types of evidence.
- *Flexibility*: Entities have full control over which components to include in their Autonomys PoP.
- *Interoperability*: Integrates with current and emerging identity systems, and existing personhood evidence from web2 or web3 accounts, utilizing TLS ZK-proofs [17] [18] [19] for rapid verification, improving user experience.

A composable, privacy-preserving PoP protocol is integral to the building of a novel digital identity system for an AI-integrated world. This approach aims to provide a familiar user experience while maintaining absolute privacy and anonymity. Using multiple, preexisting personhood factors allows Auto Score to circumvent issues of accessibility and centralization present in current biometric PoP protocols [16], and ensures that every person has the ability to autonomously demonstrate their humanity unimpeded by borders and institutions. By approaching proof-of-personhood as a composable and probabilistic measure, Autonomys offers a more nuanced and adaptable solution to the challenge of verifying human identity in digital spaces. This system preserves individual privacy while providing sufficient assurance of personhood to enable trust in human-AI interactions and decentralized governance processes.

Auto ID and Auto Score represent a vital contribution to the development of the autonomous economy by providing it with an accessible, standardized framework for digital identity and data provenance. This will help facilitate verifiable human-AI interaction, enable privacy-conscious verification, and establish metrics of trust, ensuring traceability and promoting digital safety and inclusion.

C. Decentralized Reputation Systems

The Autonomys Network, and its Auto ID and Auto Score components, provide strong foundations on which to build a robust decentralized reputation system (DRS) that would allow participants to make anonymous yet verifiable assertions about their own reputation [20]. An Auto ID-based DRS would offer users the ability to selectively share reputation claims, such as a credit score or developer reputation, in a way untraceable to their primary ID, while preserving Sybil-resistance and security against manipulation, including whitewashing and denial. Such a robust DRS would allow novel applications to be built on Autonomys—from peer-to-peer commerce and gig economy platforms to protocols for decentralized lending, crowdfunding, and collaborative research.

D. Content Provenance and Data Sovereignty

Data sovereignty—the ability of individuals to control and maintain authority over their personal data and digital presence—is crucial in an era where sensitive data is frequently exploited by criminal actors and centralized entities. The integration of content provenance with Auto ID is a stepping stone towards data sovereignty. Cryptographically linking digital content with its creator's authenticated identity establishes an immutable record of origin and subsequent modifications [21]. Such a system empowers users with granular control over their data sharing preferences and provides a robust framework for verifying the authenticity of digital assets. It also offers a potential solution to the challenges posed by synthetic media, allowing recipients to discern between genuine and artificially generated content [22]. Furthermore, as AI systems continue to evolve and generate increasingly sophisticated outputs, the ability to trace the lineage of training data and resultant content is becoming increasingly important [23].

E. Data Contribution and Compensation

Consumer devices, industrial hardware and other electronic equipment generate and record vast amounts of information about the world which gets discarded after expending its usefulness to the device owner. In some cases, the data is retained by the device manufacturer in a manner opaque to the device owner. Since AI models have already virtually exhausted the world's existing Internet-accessible data sources, the often real-time data provided by Internet-of-Things (IoT)-enabled hardware like these is the next step in data acquisition for machine learning.

The Auto ID system can be leveraged to enable users to participate in decentralized learning initiatives (such as federated learning and swarm learning) by contributing their data to machine learning models, while maintaining privacy and control over their personal information. Decentralized learning allows for model training on distributed datasets without the need for centralized data storage, thereby mitigating risks associated with data breaches and unauthorized access. The integration of decentralized learning with blockchain-based identity and compensation mechanisms represents a significant step towards a more equitable and decentralized AI ecosystem. Moreover, it creates a new paradigm for data ownership and monetization, where individuals can directly benefit from the value their data creates in AI systems (see Fig. 2). This approach aligns with the principles of data sovereignty and deAI, and addresses growing concerns about data privacy and the centralization of AI development [24] [25].

To incentivize high-quality data contribution and ensure fair compensation, the Autonomys Network will implement a data

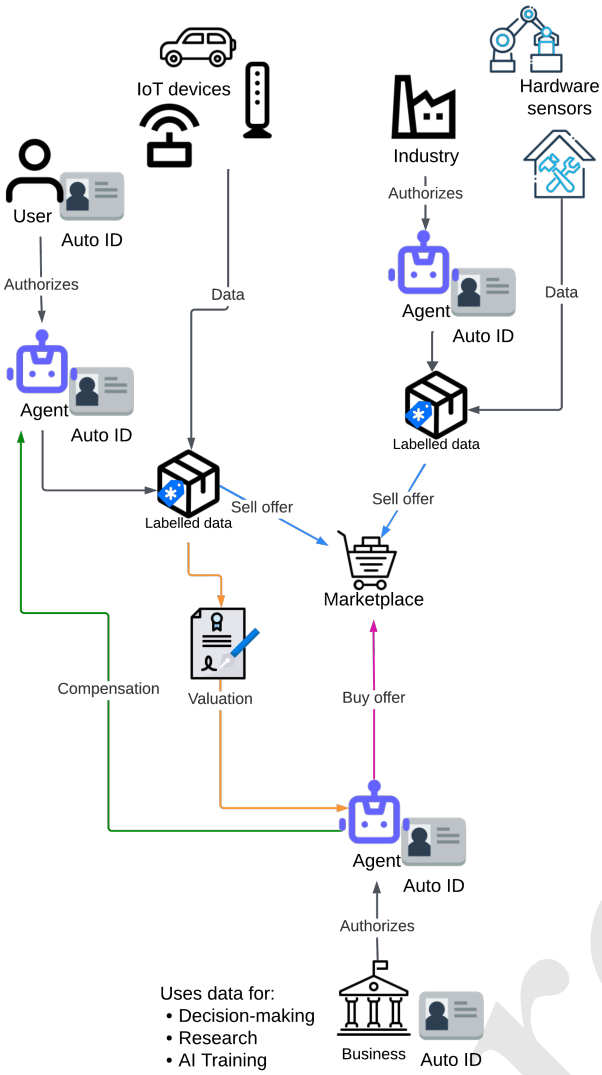

Fig. 2. Data Contribution and Compensation

valuation framework—inspired by Shapley value-based methods [26] and [27]—which quantifies individual contributions to the decentralized learning process. It considers multiple factors in its valuation algorithm, including data quality, uniqueness, relevance to the specific model being trained, and the impact on model performance improvements. This approach aims to accurately reflect the true value of each user's data contribution, moving beyond simplistic metrics such as data volume.

The valuation process will be complemented by a compensation mechanism that utilizes the Autonomys Network's native token. The implementation of such specialized mechanisms naturally maps onto Autonomys' domain layer of decoupled execution environments, allowing for independent development and upgradability without burdening the core protocol. This domain will automatically remunerate users whenever their data is accessed or utilized in model training or inference. Given the large number of individual users and data points the system will have to process during every training and inference request, and the number of payouts this will entail, we will employ several optimizations. These include accruing compensation in a smart contract and letting users initiate claims for payouts. All the accrual and claim records on the domain will be anchored and archived within the global history of the chain, ensuring transparency and immutability in recording data usage and corresponding compensation.

The system may also implement a dynamic pricing model that adjusts compensation based on market demand for specific types of data, creating a more efficient data marketplace. As an additional benefit, this approach could potentially lead to more diverse and representative datasets, addressing issues of bias in AI systems that often arise from limited or homogeneous training data.

F. Decentralized Learning and Proof-of-Training

Decentralized learning approaches aim to train machine learning models across multiple devices or nodes without relying on centralized data aggregation, thereby preserving privacy and data ownership. The Autonomys Network's underlying Subspace Protocol is uniquely positioned to facilitate decentralized learning as it addresses several challenges that impede the practical implementation of decentralized AI storage/compute-sharing DePIN.

Li (2023) [28] identifies the significant state storage and bandwidth requirements of ML as the primary barriers to the mainstream usability of existing systems. The issues of state bloat and history growth beyond the capacity of any single node are addressed in our novel Proof-of-Archival-Storage consensus mechanism and distributed storage network (DSN) that store only partial state and partial history on each individual node. The bandwidth required to support the movement of large amounts of training data and models through our network will be achieved—without hindering decentralization—following the implementation of our scalability roadmap.

Li also determines the ability to dynamically adjust workload based on demand for AI jobs in a way decoupled from transaction validation and consistent block time requirements to be a highly desirable feature of a deAI utility network. This is achievable through Autonomys' decoupled execution framework, which gives domains—independent execution environments—the freedom to set any particular hardware requirements for nodes running execution on that domain, and only commit state transitions when there is demand for the domain's resources. This architecture allows for efficient allocation of computational resources for decentralized learning tasks while maintaining the blockchain's normal operation and other network activities.

The Proof-of-Training (AI-PoT—to distinguish it from proof-of-time (PoT)) protocol described by Li [28] could be adapted to function as a specialized domain on the Autonomys Network. The AI-PoT domain would manage the training processes, including task distribution, model validation, and reward allocation, while benefiting from the underlying security and scalability of the Autonomys Network. The operators

of this domain would act as service providers, validators, and verifiers, with their roles and responsibilities defined by the Proof-of-Training protocol. The workflow described in [28] could be run on Autonomys' domains framework as follows:

1) *Client Submission*: A client submits an order containing model specifications, training data, and payment information via a transaction to the AI-PoT domain.
2) *Order Processing*: The order transaction is picked into the domain mempool and eventually added to a bundle, which is then submitted to the consensus chain for farmers to include in a block. Once in a block, the order becomes available for service providers (a selected subset of staked domain operators) to fetch.
3) *Model Training*: Service providers compete to train the best model based on the order specifications.
4) *Claim Submission*: As service providers generate improved models, they submit claims containing model signatures to the network.
5) *Model Revelation*: After the training period ends, operators reveal the full models corresponding to their submitted signatures.
6) *Validation Phase*: Validators (the rest of the operators on the AI-PoT domain) evaluate the revealed models using the specified validation function and test data before broadcasting validation messages.
7) *Verification*: Any honest operator on the domain can challenge any suspicious validations, adding an extra layer of security.
8) *Challenge Period*: A time window allows for potential challenges to be resolved.
9) *Finalization*: The network finalizes the results, determining the best model and associated rewards within the challenge period.
10) *Payment Distribution*: As soon as the challenge period has passed, the payments are distributed to the winning service provider and validators.
11) *Result Retrieval*: The client can retrieve the best model from the network.

The Autonomys Network's native token could be utilized for staking and rewards within an AI-PoT domain, creating a robust economic incentive structure for honest participation. By integrating AI-PoT as a domain, the Autonomys Network will be able to offer a future-proof solution for distributed AI training, capable of adapting to new AI models and training methodologies without requiring changes to the core protocol.

Autonomys' flexible domain framework also allows for the implementation of other common decentralized learning paradigms, including federated and swarm learning, and is thus adaptable to an application's specific needs. To enhance the security and privacy of the federated learning process, the Autonomys Network plans to incorporate secure multi-party computation (MPC), differential privacy, and other advanced cryptographic techniques [29]. These methods allow for the aggregation of model updates without exposing individual user data, protecting user privacy, while enabling valuable contribution to AI development.

Decentralized learning systems integrated with Auto ID benefit from a persistent DRS for compute providers, ML engineers, and dAgent developers that testifies to the quality of their previous contributions, trained models and built applications.

G. dAgent Infrastructure and Multi-dAgent Systems

The emergent AI agent technology ecosystem, exemplified by projects such as BabyAGI [30], AutoGPT [31], and GPT-Engineer [32], has demonstrated the immense potential of autonomous AI systems. These projects showcase the ability of AI agents to perform complex tasks, engage in goal-oriented behavior, and even recursively improve their own capabilities. At their core, they are based on simple, yet powerful techniques—chains of prompts and responses that decompose large tasks into independent sub-tasks that execute autonomously in a multi-step process before self-validating the output. Frameworks like LangChain [33] have extended these agentic capabilities by providing API calls which allow local agents to interact with the "outside" world. Chainlink Functions [34] have made web2 APIs composable with blockchain rails and web3 smart contracts. The success of these initiatives has ignited widespread interest in agentics, pointing to a future where AI agents play an increasingly important role in various applications. If each individual and business entity is to have multiple agents acting on their behalf, it is imperative we build the infrastructure to support an economy of billions of these agents.

Infrastructure for agent deployment differ in their hosting structure and location and in their method of interaction with external service providers and other agents. On the hosting layer, agents can be categorized into specialized agents—that can run exclusively on edge devices using smaller models—and generalized agents—that require high-density GPUs and large amounts of RAM. Generalized agents utilize large frontier models for task decomposition, prioritization and result validation, allowing them more advanced reasoning levels compared with specialized agents, which use smaller, task-specific models. However, specialized agents offer advantages in terms of lower latency, reduced power consumption, and improved privacy due to their ability to operate locally on edge devices. It is thus prudent to assume that there will be a significant heterogeneity of model sizes, hardware requirements and capabilities in use. The Autonomys Network is interoperable with these various platforms owing to the common composable interface and network between domains. Agents that require hardware beyond the self-hosting capabilities of a single user or organization can be programmed to run continuously or on-demand via specialized Autonomys compute-sharing domains. On-chain agents (dAgents) that are in constant high demand may benefit from being deployed on their own domain with specific hardware requirements for operators of that domain.

In addition, agents need digital storage integration for their memory and knowledge base. The Autonomys Network's decentralized storage layer is able to provide this data avail-

ability. The most effective agentic decision-making is achieved when agents have access to data outside of their training set, such as information about events that occurred after the training was complete, specialized domain knowledge, or the personal data of the user. Financial trading agents, for example, greatly benefit from access to real-time news from around the world, as do many other applications. The process of factual data retrieval from external sources to enhance the reliability of generative AI outputs is known as retrieval-augmented generation (RAG). Autonomys dAgents can perform RAG by tapping into the sovereign data economy described in Data Contribution and Compensation to access data (stored in archival storage) being offered in the marketplace, and compensating the creators for its use (in our native token).

On the higher levels of the stack, dAgents perform tasks on their users' behalf in accordance with user intents. This entails users delegating authority to them to carry out certain permitted activities, including managing user authentication and authorization while interacting with external services. DAgents specifically need permission to access their user's financial resources and means of transferring them as payment for goods and services. Human users and other dAgents on Autonomys can define hyper-specific permissions for agentic interaction with Auto ID, enhancing security and privacy. Every Autonomys dAgent interacting with the network obtains an identity at deployment, registered through Auto ID, providing verifiable and tamper-resistant dAgent identities. These IDs may be issued by individuals or organizations with metadata about the dAgent's purpose and capabilities. Possession of an Auto ID by a dAgent permits it access to the economic system of the network, allowing it to manage a balance, spend funds and receive payments. All identity claims, authorization events, and agent interactions are provable on-chain, providing a transparent and immutable audit log facilitating accountability and post-hoc analysis. As a unified system for all entities onchain, Auto ID simplifies the invocation of registered dAgents for both users and other dAgents. This dAgent invocation mechanism is augmented with a distributed reputation system for optimized reliability and performance.

Agent-to-agent communication requires a common interface to facilitate seamless interaction and collaboration on complex tasks, such as organizing a conference, illustrated on Fig. 3. Autonomys' unified identity framework unlocks composability for dAgents and cooperation for effective task execution through the advent of multi-dAgent systems (MdAS). Each dAgent can expose endpoints within a shared interface that allow other entities to discover the list of services it provides and actions it is authorized to perform.

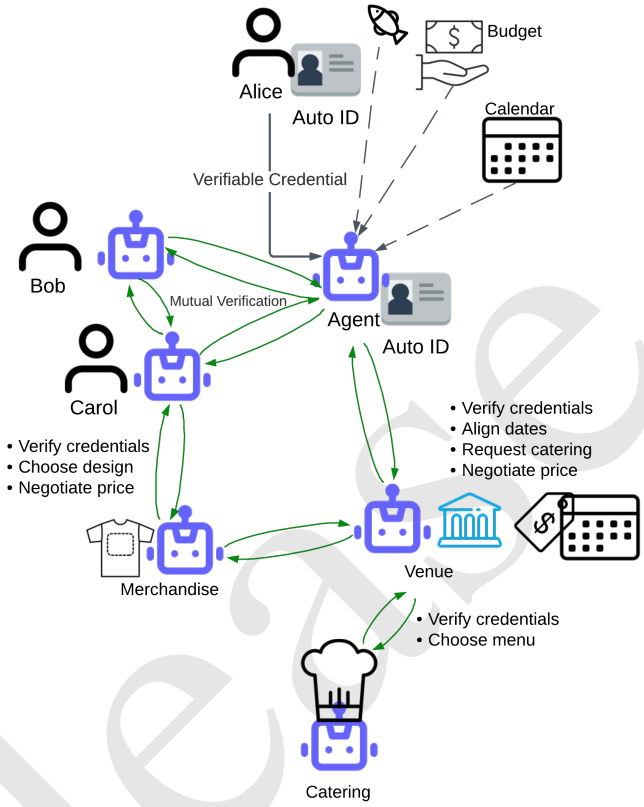

Fig. 3. Example of a multi-dAgent system (MdAS) for coordinating the organization of a conference with a catering service and a merchandise stand: **1)** Alice intends to book a venue for a conference. She authorizes her personal dAgent to manage her calendar and represent her in specific interactions by signing verifiable credentials (VCs), allowing it to share information with other dAgents. **2)** Alice's dAgent contacts those of her colleagues, Bob and Carol. Each dAgent exchanges VCs to verify that they are authorized representatives of their users. **3)** Each dAgent exchanges their user's calendar availability to align their schedules, and confirms the requirements, budget and technical setup for the conference, the merchandise to be distributed and each person's dietary restrictions. **4)** Alice's dAgent initiates a dialogue with the conference venue's customer service dAgent. Both agents exchange their VCs to ensure they are authorized to make bookings and share sensitive information. **5)** The venue's dAgent verifies availability and confirms the preferred time for the conference. Alice's dAgent communicates their dietary restrictions to ensure the venue's catering can accommodate them. They discuss the technical setup and space needed for merchandise distribution before negotiating a price. **6)** The venue's dAgent coordinates with the catering service dAgent to ensure all dietary restrictions are met. **7)** The dAgents of the merchandise suppliers are contacted to confirm the price, design, delivery and setup of the conference materials and products. **8)** All dAgents use their edit-permission VCs to update their users' respective calendars with the relevant information. The venue's booking system is updated with the reservation and all specified requirements.

H. Open Collective Intelligence and the Global DAO Mesh

Recent developments in decentralized autonomous organization (DAO) technology have showcased the numerous avenues of potential they provide for the future of collaborative decision-making and resource allocation [35]. Building upon these foundations, we propose a novel framework that leverages the power of collective intelligence through a network of Auto DAOs interconnected in a Global DAO Mesh.

Collective intelligence, a form of intelligence emerging from collaboration between many individuals, is already being harnessed for effective decision-making and governance by DAOs. Auto DAOs—smaller, specialized DAOs, composed of both human and AI members, deployed on the Autonomys Network—demonstrate the efficacy of their collective intelligence by managing decentralized projects. Examples

include web3 initiatives, investment funds, and research and development for open-source software. When these Auto DAOs are integrated into a larger, interconnected network—the Global DAO Mesh—a more effective and efficient system of collective intelligence emerges.

Autonomys envisions the Global DAO Mesh serving as the decentralized framework for open collective intelligence (OCI)—a more humanistic, albeit AI-augmented, alternative to artifical general intelligence (AGI). OCI operates on the principle of distributed problem-solving, where large, complex challenges are decomposed into smaller, more manageable tasks. These subtasks are then allocated to different Auto DAOs based on their specialized knowledge domains and vested interests in the issue at hand. The process of collective problem-solving via OCI within the Global DAO Mesh can be described as follows:

1) *Problem Decomposition*: Complex issues are broken down into independent components.
2) *Task Allocation*: Subtasks are distributed to relevant Auto DAOs within the mesh.
3) *Parallel Processing*: The human and AI members of each Auto DAO collaboratively address its assigned component.
4) *Solution Aggregation*: The Global DAO Mesh aggregates the solutions from individual Auto DAOs (potentially via a round of consensus with a weighted function representing the relative expertise of each DAO).
5) *Recombination and Synthesis*: The aggregated solutions are recombined to form a cohesive resolution to the original, complex problem.

Inspired by mixture of experts networks (MoE) [36], all steps in the process can be mediated via an agentic AI system that understands the necessary context on existing DAOs, their public members' expertise, and the prior participation records of both. This methodology creates a networked intelligence that is both decentralized and scalable, capable of addressing challenges of a magnitude not feasible for individual human or DAO entities. Drawing parallels with the Allora Network [37], our proposed system similarly leverages decentralized machine intelligence. However, while Allora focuses on a self-improving AI network, the Global DAO Mesh's OCI emphasizes the synergy between human and AI intelligence within a decentralized governance structure. Autonomys' Global DAO Mesh thus represents a significant advancement in collective problem-solving capabilities. The Global DAO Mesh distinguishes itself via several key advantages:

- *Hybrid Intelligence*: Combines the strengths of both human intuition and AI computational power.
- *Specialization*: Leverages the unique expertise of different Auto DAOs for optimal problem-solving.
- *Decentralization*: Ensures no single point of failure and promotes a truly distributed decision-making process.
- *Scalability*: Allows for the tackling of increasingly complex problems by distributing the workload across multiple Auto DAOs.

Important research directions to ensure an ethically aligned global decision-making system include optimizing task allocation algorithms to ensure fair representation, developing robust consensus mechanisms for solution aggregation, and exploring the potential for emergent behaviors within the Global DAO mesh.

I. Verifiable AI Infrastructure as a Public Good

The provision of a public good infrastructure for accessible AI is of paramount importance in our rapidly evolving technological landscape. As highlighted by Korinek and Stiglitz (2021), the advancement of AI technologies has significant implications for income distribution and employment [38]. Equal access to AI is crucial if we are to maintain economic relevance and reduce the risk of AI-driven inequality. Democratizing AI accessibility entails helping ensure that the benefits of these technological advancements are more equitably distributed across society. In pursuit of this goal, Autonomys is committed to establishing infrastructure that offers equitable access to verifiable AI agents, tooling and resources as a public good.

A key component of this digital public infrastructure is Autonomys' dedicated domain for the decentralized storage and distribution of open-source AI data within our extensive, permanent DSN. The primary objective of the decentralized open-source AI (dOSI) domain is to securely store and make freely available a wide range of AI resources, including:

- Open-source AI models
- Publicly available training datasets
- Fine-tuning datasets

By utilizing the Autonomys Network's immutable and distributed storage capabilities, the dOSI domain provides a robust, permissionless, decentralized solution for building and deploying models, at the same time as preserving critical AI assets. We are thus able to ensure that these valuable resources remain accessible and protected against potential censorship or removal in perpetuity.

IV. The Subspace Protocol

At its core, the Autonomys Network implements Subspace [11], a novel storage-based consensus protocol that separates consensus from execution. The Subspace Protocol was designed from the ground up to enable an open and inclusive Internet by:

- Providing an energy-efficient and eco-friendly alternative to proof-of-work (PoW), while still allowing for mass participation by ordinary users.
- Creating an incentive-compatible permissionless network that encourages and maintains decentralization over the long term.
- Scaling network storage and compute capacity proportional to the number of node operators, without sacrificing decentralization or security.
- Connecting and enabling interoperability between existing networks.

Achieving this vision required an alternative to both resource-intensive PoW mining and permissioned proof-of-stake (PoS)—a cryptographic proof system based on an underlying resource that is already massively distributed and which does not lend itself to special-purpose hardware. Enter *proof-of-capacity*[1](PoC), which replaces compute-intensive mining with storage-intensive farming, under the maxim of one-disk-one-vote. Disk-based consensus is an obvious solution as storage hardware consumes negligible electricity, exists in abundance across end-user devices, and has long been commoditized.

Subspace uses a longest-chain PoC consensus mechanism based on solid-state drive (SSD) storage. Adhering to Nakamoto's vision, the blockchain is permissionless but secure, with respect to safety and liveness, as long as honest farmers collectively dedicate more storage than any cooperating group of attacker nodes. In essence, Subspace follows the Ethereum model of a fully programmable, account-based blockchain, which periodically commits to the state of all accounts within the block header.

Contrary to many existing PoC protocol designs, Subspace addresses a critical mechanism design challenge—the farmer's dilemma—which poses a significant threat to the decentralization and security of PoC blockchains [11]. Rational farmers are incentivized to allocate all their available storage towards consensus, neglecting the maintenance of chain state and history [40]. This behavior leads to farmers effectively becoming light clients, degrading network security and decentralization. The trend ultimately risks consolidation into large farming pools, centralizing control around pool operators, and reducing the network's resilience against malicious actors. The farmer's dilemma also exacerbates the verifier's dilemma [41] by raising the opportunity cost of verification. If full nodes do not store the chain history, new nodes must instead rely on altruistic archival nodes or third-party data stores for initial synchronization, resulting in a more centralized network.

Subspace circumvents the farmer's dilemma without sacrificing network security or decentralization as follows (illustrated in Fig. 4):

- *To prevent farmers from discarding chain history*: we construct a novel PoC consensus protocol, based on proofs-of-storage of the blockchain's history (Proof-of-Archival-Storage), where each farmer stores as many provably unique partial replicas of the chain history as their disk space allows.
- *To ensure consensus retains the fairness of one-disk-one-vote*: farmers are discouraged from attempting to augment or replace storage with computation by making this behavior economically irrational by making plotting process computationally more intensive than Hellman's time-memory tradeoff [39].
- *To ensure chain history remains available*: farmers form a decentralized storage network, which allows chain history to remain fully-recoverable, load-balanced, and efficiently retrievable.
- *To relieve farmers of the burden of maintaining the whole state and performing redundant computation*: we apply the classic distributed systems technique of decoupling consensus and computation. Farmers are then solely responsible for ordering transactions, while a separate class of operator nodes maintains the state and computes the state transitions for each new block.
- *To ensure executors (operators) remain accountable for their actions*: we employ a system of staked deposits, verifiable computation, and non-interactive fraud proofs.

[1]We use proof-of-capacity as an umbrella term encompassing proof-of-space, proof-of-storage, proof-of-replication, proof-of-space-time, proof-of-retrievability, and other storage-based protocols.

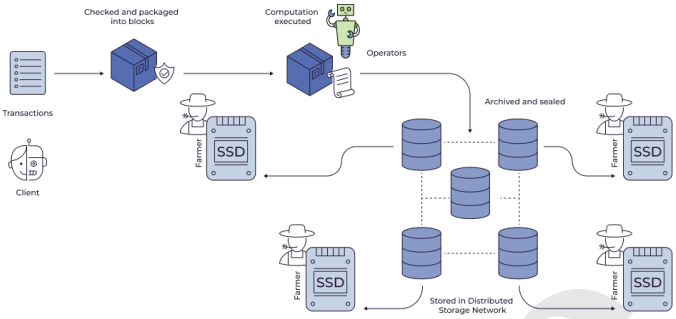

Fig. 4. Blockchain Data Flow

A. Proof-of-Archival-Storage

To participate in a Proof-of-Archival-Storage (PoAS), farmers first create and store provably unique partial replicas of the chain history, before responding to random, publicly verifiable storage audits, which allow them to forge new blocks. This stands in contrast to the PoC protocols proposed by Spacemint [42], Chia [43], and SpaceMesh [44], in which nodes stores randomly generated data, rather than useful files. PoAS is inspired by Sergio Lerner's Proof-of-Unique-Blockchain-Storage [40] mechanism, but is utilized directly for consensus.

Subspace's PoAS protocol was built to provide a superior user experience (UX) to existing PoC protocols, while maintaining the highest level of consensus security. Its most relevant UX and performance metrics are:

- *Setup time*: hours–days (depending on allocated disk space)
- *Proof-generation time*: $<$ 1 second
- *Proof size*: $<$ 1 KB
- *Verification time*: 0.001–0.01 seconds

The latest iteration of the Subspace Protocol [12] uniquely combines KZG polynomial commitment [45], erasure coding [46], and function inverting [39] to address outstanding design challenges, significantly improving upon previous versions of the protocol [11]. Below is an overview of the resulting consensus mechanism (see [12] for a more detailed description).

To create partial replicas of chain history for the farmers to store, we divide the full blockchain history F into n

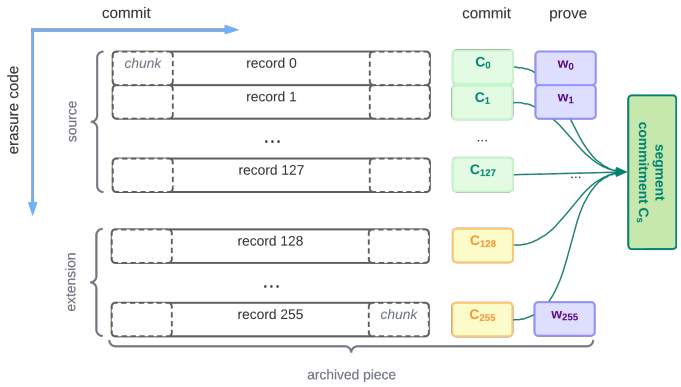

Fig. 5. Archived segment

pieces $\{d_0, d_1, \ldots, d_{n-1}\}$, each of equal size,[2] in an *archiving* construction inspired by [47] and illustrated in Fig. 5. Without loss of generality, we view each piece as a row vector of length ℓ over $\mathbb{Z}_p$ (i.e., $d_i \in \mathbb{Z}_p^\ell$)[3]. Then, F can be viewed as a matrix of size $n \times \ell$ over $\mathbb{Z}_p$. Alternatively, each piece d_i can be viewed as a polynomial $f_i(x)$ over $\mathbb{Z}_p$ of degree at most $\ell - 1$. This allows us to view F as a collection of n polynomials $\{f_i(x)\}_{i=0}^{n-1}$.

Let A_i be the KZG commitment of $f_i(x)$ for $i \in \{0, 1, \ldots, n-1\}$, known as piece commitment, and T is the KZG commitment of $(H(A_0), \ldots, H(A_{n-1}))$. T is public information. Let π_i be the KZG proof for $H(A_i)$. With π_i, anyone in the system can verify whether A_i is consistent with the public information T about the history F. This process is described in greater detail in [12].

In the practical implementation of the protocol in the Autonomys Network, we first divide the history F into segments, where each segment contains the same number of pieces. In this way, archiving is continuous (instead of a one-time process) where the archived history F is periodically updated with new segments of pieces and their respective segment commitments T_i are appended to the tail end of F in ascending order.

To participate in the network, each farmer generates a key pair $(\mathtt{sk}, \mathtt{pk})$ and derives their farmer ID $\mathtt{id}$ (e.g., $\mathtt{id} = H(\mathtt{pk})$)[4]. With a given $\mathtt{id}$, the farmer selects m polynomials $\{g_i^{\mathtt{id}}(x)\}_{i=0}^{m-1}$ in a verifiable and pseudorandom manner, and retrieves their KZG commitments $\{\mathtt{commit}\left(g_i^{\mathtt{id}}(x)\right)\}_{i=0}^{m-1}$ together with the proofs with respect to T. The farmer then creates ℓ "storage coins" $\{F^{\mathtt{id}}(\mathtt{id}+j)\}_{j=0}^{\ell-1}$ as described in [12], where each storage coin can be viewed as m polynomial evaluations at a given point

$$F^{\mathtt{id}}(\mathtt{id}+j) = \begin{bmatrix} g_0^{\mathtt{id}}(\mathtt{id}+j) \\ g_1^{\mathtt{id}}(\mathtt{id}+j) \\ \vdots \\ g_{m-1}^{\mathtt{id}}(\mathtt{id}+j) \end{bmatrix}.$$

After that, the farmer generates their masked versions $\{\tilde{g}_i^{\mathtt{id}}(x)\}_{i=0}^{m-1}$ by using the hard-to-invert function $\mathrm{MASK}_{\mathtt{seed}}(\cdot)$ as described in [12]. Finally, the farmer stores ℓ masked storage coins as well as some metadata (i.e., m commitments $\{\mathtt{commit}\left(g_i^{\mathtt{id}}(x)\right)\}_{i=0}^{m-1}$ together with their proofs). This process is called *plotting*. Note that the parameter m can differ for different farmers, depending on their pledged storage.

Once a farmer has plotted as much storage space as they wish to pledge to the network, they can participate in leader election. To elect a leader who would propose the next block, a global challenge $\mathcal{C}_t$ is generated at time slot $t = 1s$, at which point, each farmer selects one masked storage coin and collects m lottery tickets from it. If a farmer finds a winning ticket, they have to prove the following

- the winning ticket (say, $\tilde{g}_i^{\mathtt{id}}(\mathtt{id}+j)$) is indeed close enough to $\mathcal{C}_t$
- the unmasked element $g_i^{\mathtt{id}}(\mathtt{id}+j)$ is correct and is a member of history F

This process, called *farming*, is illustrated in Fig. 6. Farming is designed to perform thousands of random reads of small chunks of data per second, making it only feasible on an SSD, further enhancing energy efficiency [48] and decentralization.

The above construction provides a leader-election mechanism, which is combined with a longest-chain protocol to produce a consensus algorithm.

B. Proof-of-Time

A vulnerability of pure PoS (and by extension PoC) systems lies in their susceptibility to long-range attacks [49]. Unlike PoW systems, where block production is physically constrained by computational power, PoS/PoC systems lack this inherent limitation. Consequently, an adversary with sufficient resources could potentially rewrite a significant portion of the blockchain at any point in the chain's history, compromising its immutability and security. This vulnerability stems from the fact that historical stake distributions can be manipulated without incurring the substantial energy costs associated with PoW systems.

Additionally, PoS/PoC systems often struggle to achieve the dynamic availability and unpredictability inherent in PoW systems [50]. The challenge lies in creating a system that can adapt to fluctuating participation rates while ensuring that block proposers remain unpredictable, thus preventing targeted attacks or manipulation. These properties are crucial for maintaining robust network operation and security against various attack vectors.

The Autonomys Network addresses these challenges by implementing a separate proof-of-time (PoT) chain that interlinks with the PoAS chain. This design prevents long-range

[2] In general, F grows over time. Here, we only consider the case that F is fixed and defer the general case to our protocol specification found at https://github.com/subspace/protocol-specs.

[3] $\mathbb{Z}_p$ because we will apply KZG polynomial commitment later.

[4] This farmer ID also serves as the peer ID of their node on the networking layer.

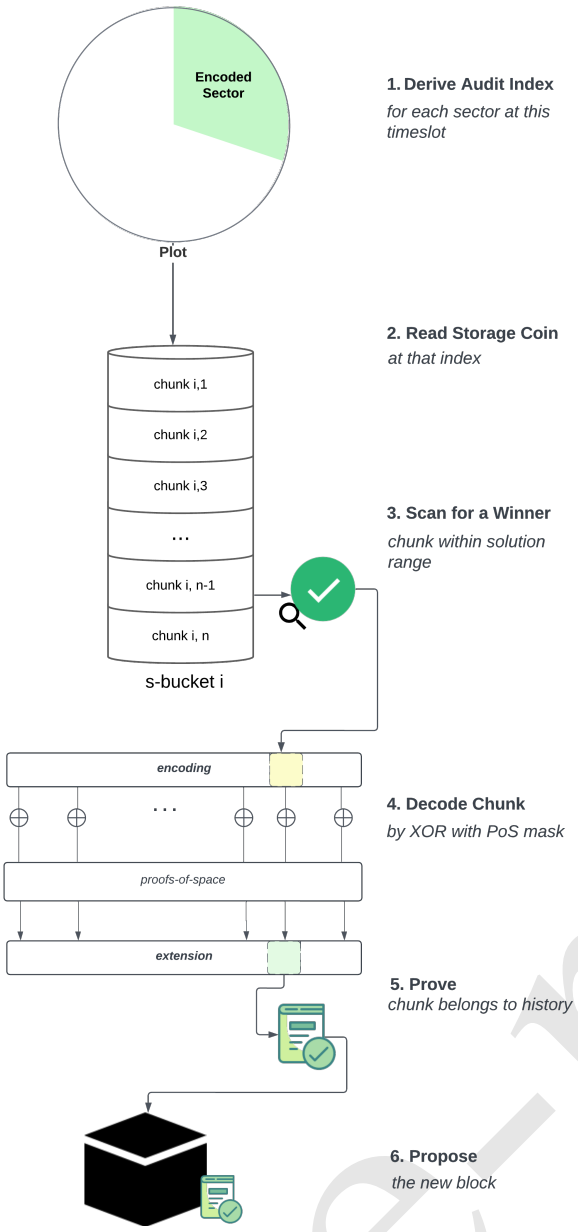

Fig. 6. Farming

attacks by enforcing a verifiable time constraint between block proposals, analogous to the arrow of time in PoW systems. PoT guarantees that a certain amount of wall-clock time must elapse between block proposals, preventing an adversary from rewriting history by ”going back in time.” PoT is constrained physically, similar to PoW, but is not parallelizable (technically, it is proof of *sequential* work) and an attacker cannot immediately generate a successful multi-year retroactive fork even with faster hardware.

The elapsed time guarantee is achieved by iterative evaluation of an inherently sequential delay function. The choice of delay function is crucial to the security and efficiency of the PoT system. After extensive analysis of existing verifiable

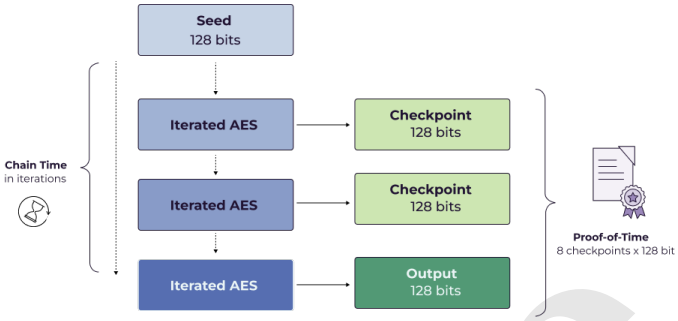

Fig. 7. Proof-of-Time checkpoints

delay functions (VDFs), we chose to employ repeated AES-128 encryption. This decision balances security, efficiency and resistance to hardware acceleration. Using the Advanced Encryption Standard (AES) leverages its extensive cryptographic research history and the availability of hardware acceleration in modern CPUs, making it an optimal choice for this application. Based on a joint study with hardware-accelerated cryptography lab Supranational, we do not expect a significant speedup over the best AES implementation, even with an ASIC.

To maintain the PoT chain, the network introduces a new node role called *timekeepers*. These nodes are responsible for evaluating the delay function and disseminating outputs. To provide PoT evaluation, timekeepers require the highest-end CPUs—unavailable to most farmer nodes. Delegating timekeeping to a separate class of nodes ensures decentralization on the consensus level, while maintaining protocol security with minimal honest participation, where the presence of at least one honest timekeeper is sufficient.

To achieve asymmetric verification time for the AES-based delay function, timekeepers publish a set of intermediate checkpoints—currently 8, spaced uniformly—alongside the output (Fig. 7). Farmers can validate each checkpoint independently and in parallel to reduce overall verification time. Including checkpoints allows other nodes to validate the output ≈ 7 times faster and use ≈ 4 times less power than evaluation by leveraging instruction-level parallelism.

The Subspace consensus protocol utilizes a farming dynamic that mimics the random nature of Bitcoin’s mining dynamic, while only expending a small amount of electricity. This is achieved through PoT-based block challenges for the block proposal lottery, based on [50]. The PoT chain serves as a randomness beacon, providing unpredictable and verifiable inputs for block challenges, thus addressing the issue of long predictability windows often seen in protocols using generic verifiable random functions. This unpredictability is at the same level as that of PoW protocols and is stronger than those using verifiable random functions.

The security of the PoT system is further enhanced by several key mechanisms. Sequentiality is achieved through output chaining between slots, ensuring that each new output depends on the previous one. To compensate for network

delays, the system implements a tunable lag parameter, allowing sufficient time for propagation and verification of PoT outputs before block proposals. The Autonomys Network also incorporates measures to mitigate the potential advantage of faster timekeepers, including periodic entropy injection. To prevent manipulation of randomness, the network employs an injection mechanism similar to that used in Ouroboros Praos [51]. This approach prevents attackers from controlling slot challenges by strategically releasing or withholding blocks, further enhancing the unpredictability and security of the system.

C. Distributed Storage

Subspace introduces a distributed storage network (DSN) to ensure consistency of storage over time, given the heterogeneous storage capabilities of farmers. Our DSN design guarantees the following properties:

- *Permissionlessness*: The system operates without central coordination, accounting for dynamic farmer availability and non-uniform growth of historical data over time.
- *Retrievability*: Both full and single-piece retrieval are facilitated, with requests balanced evenly across all farmers, ensuring that the overhead of serving history remains negligible.
- *Verifiability*: Farmers are not required to synchronize or retain the full history, yet the system remains efficiently verifiable.
- *Durability*: The probability of any single piece being lost, whether through accidental or malicious means, is minimized.
- *Uniformity*: On average, each piece is stored an equal number of times across the network.

These features enable the historical data to expand beyond the storage capacity of any individual farmer, while allowing farmers to allocate storage resources according to their individual capabilities.

The Autonomys Network DSN is comprised of multiple distinct layers, each contributing to different aspects of data availability, durability and efficient retrievability, while serving historical data pieces to requesting nodes.

a) Pieces Cache Layer (L2): The pieces cache layer is designed to facilitate efficient piece retrieval for data reconstruction and farming. Its primary function is to minimize retrieval latency. While retrieval from archival storage necessitates computationally intensive operations by farmers—taking approximately 1 second on consumer hardware—L2 retrieval is near-instantaneous due to the storage of unencoded pieces in the disk cache.

L2 cache utilizes a distributed hash table (DHT) to store pieces based on the proximity of the piece index hash to the peer ID. Farmers, being the most suitable candidates for L2 storage, allocate a small percentage of their pledged storage for this purpose. The overall storage network replication factor determines the number of farmers storing each piece.

The piece cache layer population process is as follows:

1) Nodes generate new segments of pieces during the archiving process.
2) These new segments are temporarily stored in the node's cache.
3) Farmers receive the newly archived segment index from the latest block header.
4) Farmers compute the piece index hashes within the segment and determine which pieces to pull to their L2 based on hash proximity to their peer ID.
5) Relevant pieces are then pulled to the farmer's local L2 cache.

b) Archival Storage Layer (L1): The archival storage layer is the fundamental layer responsible for the permanent storage and durability of all chain data. It comprises all storage pledged by farmers for storing masked pieces of chain history, also known as plots.

Functioning as 'cold storage', the archival storage layer ensures the availability of history pieces in the rare event of an L2 cache miss. However, retrieval from archival storage is resource-intensive and time-consuming; thus, it is utilized only when L2 retrieval fails. Typically, the L1 layer of farmers is populated with pieces received from L2.

The archival storage layer population process is as follows:

1) The farmer decides how much storage to allocate to the network.
2) Based on the amount of storage pledged, the farmer pseudorandomly and verifiably selects enough pieces of history to fill that space.
3) The farmer pulls the selected pieces from the L2 or L1 of other farmers.
4) The farmer masks the pieces as described in the plotting protocol.
5) Every time a new segment is archived, the farmer runs a check to see whether they need to replace any pieces.

The last step is necessary to ensure that new history gets replicated uniformly across many farmers in the network, regardless of how long they have been participating in the network or how long ago they initialized their plots. This plot expiration is set up such that the farmer gradually replaces subsets of pieces in the plot as the history of the chain grows. On average, by the time the history has doubled in size, as compared to when the plot was initialized, the farmer has expired and replotted half of their plot. By the time the history quadruples, the farmer has replotted their whole plot once over. The choice of gradual expiration instead of full farm replots ensures maximum uptime of the farmers' archival storage layer for serving pieces to the DSN.

c) Cache Types by Peer Roles: Separately from the above cache layers, we distinguish the following types of cache by peer role in the network:

- *Node cache*: Contains newly created pieces from the most recent archived segments. It is limited to a few recent segments and progressively replaces older pieces with new data.

- *Farmer cache*: Contains pieces in the L2 cache, automatically populated upon receipt of new archived segment announcements. Pieces are cached according to their proximity to the farmer's peer ID.

To incentivize the farmer network to maintain the desired replication factor for historical data, Subspace implements a novel algorithm that dynamically adjusts the cost of on-chain storage, or *blockspace*, in response to fluctuations in storage supply and demand.

D. Decoupled Execution

Farmers will seek to dedicate all available disk space to consensus and expend as little computation as possible, while remaining on the longest valid chain. This implies they must compute all intermediate state transitions and maintain the state. As the burden of maintaining the state and computing transitions grows larger, both the farmer's and verifier's dilemmas present themselves, leading economically rational farmers to sacrifice security for higher rewards at a lower cost, by either becoming light clients or joining a trusted farming pool. To resolve these dilemmas, we implement a method that relieves farmers of this burden, while still allowing them to be certain they are extending the longest valid chain. Critically, this method does not degrade the liveness, fairness, or safety of block production. Our solution follows the classic technique in distributed systems of decoupling consensus and computation.

In this system, farmers are solely responsible for providing subjective and probabilistic consensus over the ordering of transactions. A separate class of executor nodes—operators—computes the objective and deterministic result of that ordering. Operators are selected through a stake-based election, separate from block production, analogous to the block finalization technique proposed by Casper FFG [52]. They are incentivized by the sharing of transaction fees with farmers, and held accountable through a system of non-interactive fraud proofs [53] and slashing [54].

This approach, while influenced by Flow [55]–[57], is simpler (using two, not four classes of nodes), retains compatibility with Nakamoto consensus, and maintains the 'honest majority of farmers' and 'dishonest majority of operators' security assumptions. It also draws inspiration from Truebit [58], recognizing that optimistic off-chain computation with fallbacks to on-chain verification could realize a trustless decentralized mining pool. Unlike protocols such as ChainSpace [59] and LazyLedger [60], which achieve decoupling by delegating computation to clients, our system retains global state, allowing for cross-contract calls and composability of applications.

Under the Decoupled Execution (DecEx) framework, farmers only confirm the availability of transactions and provide an ordering, while secondary networks of staked operator nodes execute the transactions and maintain the resulting chain states. DecEx separates the probabilistic process of coming to a consensus over the ordering of transactions from the deterministic process of executing transactions, as illustrated in Fig. 8. The decoupling of these roles permits alternative hardware requirements for different node types, allowing us to keep farming lightweight and open to anyone, while also providing a foundation for scaling execution both vertically—based on the hardware capabilities of operators—and horizontally—by partitioning operators into different namespaced execution domains.

While conceptually similar to rollups on Ethereum, such as Optimism, DecEx differs heavily in its protocol implementation. Unlike Ethereum, the Autonomys Network does not have a global smart contract execution environment within the core protocol. Instead, DecEx is enshrined within the semantics of the core protocol itself. Despite being implemented at the protocol level, DecEx is still able to provide rollup protocol designers with a flexible system capable of supporting any state transition integrity framework for verifying the receipt chain, including optimistic fraud proofs and zero-knowledge validity proofs. DecEx also supports any smart contract execution environment that can be implemented within the Substrate framework, such as the Ethereum Virtual Machine (EVM) or Web-Assembly (WASM).

a) Domains: Domains are the logical extension of the basic DecEx framework, taking it from a single, monolithic execution environment to a modular, interoperable network of namespaced execution environments. Each domain is its own programmable layer-2 rollup, or application-specific blockchain (app-chain), that relies on the consensus chain for consensus, decentralized sequencing, data availability, and settlement. However, a smart contract, (super) dApp, or dAgent can use multiple domains to achieve a complex task, enabled by our unique cross-domain communication.

b) Farmer Role: In our DecEx model, users submit execution transactions directly to operators, who pre-validate and batch these transactions into bundles through a (probabilistic) stake-weighted election process. These bundles are then submitted to farmers, who treat them as base-layer transactions. Farmers only verify the proof-of-election and ensure the data is available, before batching bundles into blocks in the usual manner. When a farmer finds a PoAS solution that satisfies the storage audit, they order valid transactions into a new block, committing to the last valid state root proposal they observe. Unlike on Ethereum and most other L1s, farmers do not need to maintain the code, state or account balances for contracts, only the smaller set of balances and nonces for externally owned accounts (EOAs), and minimal information about each domain runtime, staked operators and execution receipt (ER) chains. The farmer network effectively provides decentralization-as-a-service to the domains.

c) Decentralized Sequencing: Once the bundled transactions are included in the consensus block by farmers, domain operators must execute them in a deterministic order based on a verifiably random seed from the consensus chain. This absolves the operators of the responsibility of sequencing user transactions, while also preventing them from harvesting the maximal extractable value (MEV) and causing economic harm to users. Bundled transactions from a domain are opaque to the farmer block proposer as the latter does not have the domain

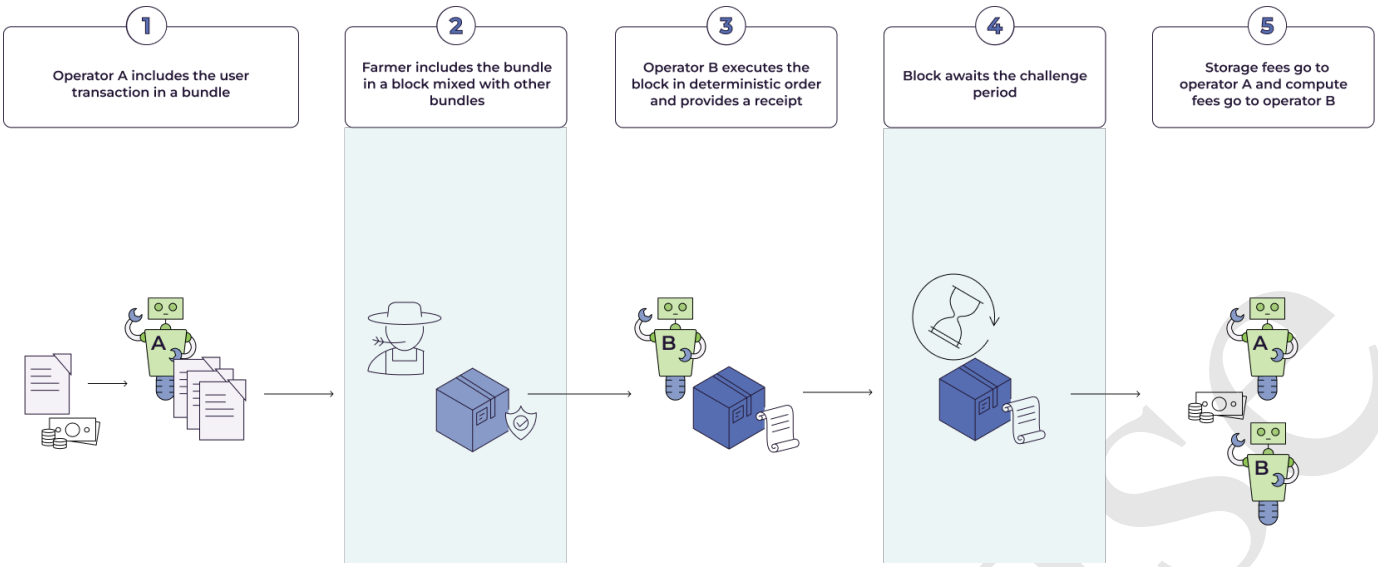

Fig. 8. Domain transaction flow from submission to fee distribution

state. Thus, the farmer cannot participate in MEV extraction either. Neither the order in which the operator batches the transactions in the bundle nor the order of bundles in the consensus block influences the final sequencing for execution.

d) Operator Role: Operator nodes maintain the full state of their respective domain and apply transactions, returning new proposed state roots. For each new block, a small constant number of operators are chosen through a stake-weighted election. Execution transactions from the block are then ordered deterministically, using a secure cryptographic shuffle based on the unique PoAS produced by the farmer. Operators then execute the transactions according to this ordering, and produce a deterministic state commitment in the form of an execution receipt, incrementally committing to intermediate state roots. These state commitments are then included in the next bundle, forming a deterministic receipt chain tracked by all farmers within the consensus chain protocol. The initial, default implementation of DecEx employs an optimistic fraud-proof validation scheme.

e) Liveness: To retain liveness in case of network asynchrony or byzantine actors, operator elections are re-run for each new time slot. This allows newly elected operators to include past ERs to catch up. The election threshold dynamically adjusts based on observed operator availability. Each domain may specify the frequency of re-election based on its own needs and demand, without interfering with the liveness of other domains or the consensus chain.

f) Fairness: Fairness is preserved through a fair compensation mechanism between farmers and operators. Farmers are compensated for their blockspace at the current price of storage, and for their work of including the domain bundles, by the operators. Operators are compensated for the blockspace costs incurred if their ERs are valid, and for their work of bundling and executing user transactions, via transaction fees.

g) Validity: Validity is ensured through a system of fraud proofs. Within the challenge period, any honest node which operates on a domain can compile a fraud proof for an invalid state transition performed by another operator on that domain. The fraud proof can be verified by any consensus node without having the whole domain state. If it is valid, the operator who proposed the invalid ER will have their entire deposit confiscated. Any operator who has extended the invalid ER is also slashed as punishment for dishonest or lazy behavior.

h) Finality: Transactions on optimistic domains are subject to a challenge period until they are settled on the consensus chain. During the challenge period, nodes can dispute the correctness of state transitions presented by operators. Any node that has an up-to-date state of the domain can submit fraud proofs for this domain and does not need to be a staked operator to do so. Whether the node is acting honestly or not in this particular instance is determined by the validity of the fraud proof. Currently, the challenge period on domains is 14400 blocks, or approximately 1 day. Fast finality is possible for services that run their own honest operator nodes. Since the operator nodes execute all the state transitions, they can be certain about the correctness of the domain state at any given time.

i) Verifier's Dilemma: The verifier's dilemma is addressed by requiring operators to reveal fraud to protect their own stake and claim their share of the rewards. Operators are punished for extending an invalid ER without first demonstrating fraud.

j) Safety: Safety is maintained by distinguishing between illegal and invalid transactions. Farmers enforce legality by ensuring transactions have valid signatures and can cover specified fees. Operators enforce validity by applying transactions deterministically in the order specified by farmers.

k) Network Dynamics: The system accounts for network delays and stochastic block production. Operators are incen-

tivized to generate fraud proofs locally to release their own ERs as soon as possible, speeding up fraud proof propagation and strengthening security guarantees. Farmers order by urgency and deduplicate fraud proofs in their mempool to ensure timely inclusion.

l) Adversarial Scenarios: The system is designed to handle various adversarial scenarios, including attempts to attack the liveness of execution or confuse farmers about transaction legality. Even in the presence of a dishonest majority of operators, the system remains secure as long as a single honest operator remains connected to an honest farmer within their peer set.

m) DecEx Summary: Our decoupled execution system allows for scalability improvements by independently scaling transaction throughput and storage requirements. It preserves the security properties of Nakamoto consensus, even in the presence of a dishonest majority of operators, given an honest majority of farmers on the consensus layer.

Our approach provides a unique solution to the challenges faced by storage-based blockchains, offering a balance between permissionless farming and permissioned staking. Unlike hybrid PoC/PoS consensus mechanisms employed by other storage-based blockchains, Autonomys' system clearly distinguishes between a permissionless farming mechanism for block production and a permissioned staking mechanism for block finalization.

By simultaneously addressing the farmer's dilemma, verifier's dilemma, and blockchain bloat issues, the Autonomys Network presents a comprehensive solution to several critical challenges in the blockchain industry. It aims to make blockchains more energy-efficient, egalitarian and decentralized while maintaining the necessary security and functionality for complex smart contract and application development.

V. Scalability

Blockchain scaling has received extensive attention over the past decade. Numerous scalability protocols have been proposed in the literature, including the likes of Prism [61] and OmniLedger [62]. Building on this existing research, Autonomys is taking a first-principles approach to scaling the Subspace Protocol. The section below outlines this approach to scalability and its implementation.

A. Constraints to Scaling Blockchain TPS

For any blockchain system, there are at least three physical constraints: (1) the *communication* constraint—the upload bandwidth of a participating node; (2) the *computation* constraint—the number of transactions executed per second by a node; and (3) the *storage* constraint—the number of transactions stored by a node. The goal of blockchain scaling[5] is to achieve the maximum possible throughput under these physical constraints, measured by TPS (transactions per second).

[5]Delay is another important metric for blockchain scaling, but is beyond the scope of this white paper.

In a conventional blockchain design, a participating node (often referred to as a full node or a miner) has to download, store and execute all the transactions. This requirement leads to several upper bounds. For instance, the throughput cannot exceed the average upload bandwidth divided by the average size of transactions. Thus, if the average bandwidth is 10 Mbit/s and the average size is 250 bytes, the throughput cannot exceed 5000 TPS under the communication constraint—too small for certain applications. The huge number of transactions generated by the future Internet of dAgents, as well as the mainstreaming of the burgeoning decentralized finance (DeFi), decentralized science (DeSci) and on-chain gaming (GameFi) ecosystems, will significantly expedite the demand for greater scalability. How can we scale the Autonomys Network throughput by 100x to 500,000 TPS?

B. Scaling the Autonomys Network

In order to achieve our goal throughput of 500,000 TPS, we could increase the upload bandwidth to at least 1Gbit/s, but this would sacrifice decentralization, as nodes with low bandwidth could no longer participate. Instead, having already decoupled the requirement that every node store and execute all transactions, via our DSN and DecEx, we are now decoupling the bandwidth requirement.

Inspired by the similarities between rollup designs and sharding designs [63], we take a unique sharding approach by leveraging cryptographic sortition and aggregate signatures. Our system consists of a beacon chain and multiple data shards. The beacon chain is maintained by all the farmers through the PoAS consensus algorithm. Each data shard is maintained by a subset of farmers selected by cryptographic sortition over time. More specifically, a farmer is elected as a leader for the beacon chain if they have a lottery ticket close enough to $\mathcal{C}_t$ (i.e., the distance between the ticket and $\mathcal{C}_t$ is smaller than a threshold T_b); elected as a member for data shard 1 if the distance is no smaller than the threshold T_b, but smaller than $T_b + T_s$; elected as a member for data shard 2 if the distance is no smaller than $T_b + T_s$, but smaller than T_b+2T_s; and so on. Generally, a farmer is elected as a member for data shard i if the distance is no smaller than $T_b+(i-1)T_s$, but smaller than $T_b + iT_s$. (Note that a farmer doesn't belong to any data shard until they are elected as a member.) This creates a *dynamic* membership for every data shard, recorded on-chain as farmers have to prove their winning tickets, as described above.

When a new domain is initiated, it joins a data shard. Once a domain operator produces a new bundle, it sends the bundle to all recent members in its data shard and collects their signatures. Upon receiving a quorum of signatures, the operator aggregates them, producing a certificate bundle that contains the aggregate signature, the bundle header, and some useful metadata. Then, the operator broadcasts the certificate bundle to all the farmers via gossip communication protocol. A certificate bundle is treated as a transaction by the beacon chain, which orders the bundles for a domain (rather than the domain operators or its data shard). This means that the beacon

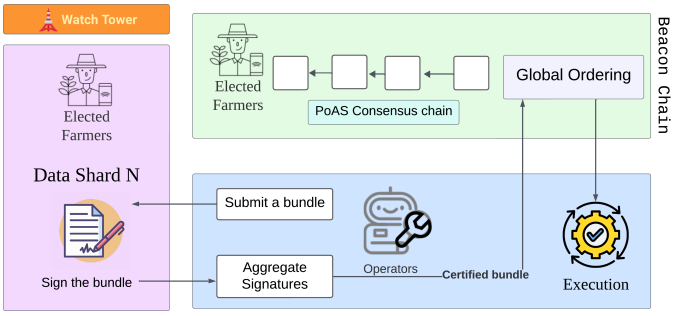

Fig. 9. Data flow between domain, shard and beacon chain

chain guarantees the safety of a domain through our PoAS consensus. That is, the safety of a domain won't be violated even if its data shard consists of a malicious majority. This workflow is illustrated for one domain and shard in Fig. 9.

If a data shard does consist of a malicious majority, the domain operator cannot collect a quorum of signatures, leading to a liveness violation. Such a violation is temporary as each data shard has a dynamic membership. We can even set the system parameters so that liveness violations happen with extremely low probability. Similar ideas have been explored in recent work [64], albeit for permissioned systems rather than permissionless ones.

Finally, we turn to data withholding attacks, where a malicious operator colludes with malicious members in its data shard, and only shares the bundle with those members. If the malicious members form a majority, the operator is able to produce a valid certificate bundle, while no honest members nor honest operators can obtain the original bundle. To address this issue, we propose using a watchtower for each data shard. When a watchtower detects such adversarial behavior, it complains on the beacon chain using a mechanism similar to that in [65].

VI. Conclusion

The Autonomys Network represents a dual solution to both the:

- challenges of security, decentralization, verifiability and scalability facing web3 infrastructure—embodied in the farmer's and verifier's dilemmas, and the blockchain trilemma—and the
- risks and opportunities posed by the emerging AI-augmented world.

In implementing our cutting-edge blockchain technologies and a decentralized identity system, we have built a robust decentralized framework that not only addresses the immediate challenges of identity verification and content authenticity, but also lays the groundwork for a future where humans and AI can interact in a transparent, secure, and trustworthy manner.

The Autonomys Network stack, composed of the dApp/dAgent, domain (DecEx), consensus (PoAS), and storage (DSN) layers, forms a comprehensive ecosystem that enables unprecedented scalability, security and flexibility. Our permissionless peer-to-peer network allows for wide participation, while the Subspace Protocol's Proof-of-Archival-Storage consensus mechanism ensures efficient and environmentally friendly operation. Key innovations such as Auto ID and Auto Score provide a secure and verifiable identity system for both humans and AI entities. This addresses the fundamental challenge of establishing trust in an increasingly AI-integrated world, allowing for authenticated content creation and delegation of permissioned authority. The Autonomys Network's modular architecture, with its decoupled execution domains and distributed storage network, offers unparalleled scalability and adaptability. This design allows for the seamless integration of various state transition frameworks and execution environments, fostering innovation and interoperability across different blockchains.

We envision the Autonomys Network as a fair, open-source and collaborative web3 ecosystem for verifiable dApp and deAI development, deployment and interaction. As well as seeking to foster innovation and growth in the AI and blockchain spaces, we want to empower individuals to maintain control over their digital identity and economic relevance, and ensure the benefits of technological advancement are accessible to all, regardless of their resources or background.

Acknowledgements

The authors would like to thank Saeid Yazdinejad (University of British Columbia) for his contributions to the scalability roadmap; Barak Shani for his contributions to the Subspace protocol security; the Autonomys Labs team, especially Jeremy Frank for contributions to Auto ID, and Chris Sotraidis and Charlie McCombie for their valuable feedback.

References

[1] LeCun, Y., Bengio, Y., and Hinton, G. *Deep learning* Nature, 521(7553), 436-444, http://dx.doi.org/10.1038/nature14539, 2015.

[2] Frey, C. B., and Osborne, M. A. *The future of employment: How susceptible are jobs to computerisation?* Technological Forecasting and Social Change, 114, 254-280, https://doi.org/10.1016/j.techfore.2016.08.019, 2017.

[3] Brynjolfsson, E., and McAfee, A. *The second machine age: Work, progress, and prosperity in a time of brilliant technologies* W. W. Norton & Company, 2014.

[4] Zheng, Z., Xie, S., Dai, H., Chen, X., and Wang, H. *An overview of blockchain technology: Architecture, consensus, and future trends* In 2017 IEEE International Congress on Big Data (BigData Congress), 557-564, IEEE, https://doi.org/10.1109/BigDataCongress.2017.85, 2017.

[5] O'Neil, C. *Weapons of math destruction: How big data increases inequality and threatens democracy* Crown Publishing Group, 2016.

[6] Bostrom, N. *Superintelligence: Paths, dangers, strategies* Oxford University Press, 2014.

[7] Russell, S. *Human compatible: Artificial intelligence and the problem of control* Viking, 2019.

[8] Miessler, D. *AI's Predictable Path* Unsupervised Learning, https://danielmiessler.com/p/ai-predictable-path-7-components-2024, 2023.

[9] *Apple Intelligence* https://www.apple.com/apple-intelligence/, 2024.

[10] Altman, S. *Moore's Law for Everything* https://moores.samaltman.com/, 2021.

[11] Wagstaff, J. *Subspace: A Solution to the Farmer's Dilemma* https://subspace.network/news/subspace-network-whitepaper, 2021.

[12] Feng, C., Porechna D., Shani B., and Wagstaff J. *(WIP) Dilithium: A Proof-of-Archival-Storage Consensus Protocol for Subspace* https://github.com/subspace/consensus-v2-research-paper/blob/main/consensus_v2.pdf, 2023.

[13] Crapis, D. *The Internet of Agents* https://davidecrapis.notion.site/The-Internet-of-Agents-23aa09799b9c4620a1a287926bcfd6af, 2024.

[14] Cooper D., Santesson S., Farrell S., Boeyen S., Housley R., and Polk W. *Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile* RFC 5280, https://www.rfc-editor.org/rfc/rfc5280, 2008.

[15] Sporny M., Guy A., Sabadello M., Reed D., Longley D., Allen C., and Steele O. *Decentralized Identifiers (DIDs) v1.0: Core architecture, data model, and representations* W3C Recommendation, https://w3c.github.io/did-core/, 2022.

[16] Buterin, V. *What do I think about biometric proof of personhood?* https://vitalik.eth.limo/general/2023/07/24/biometric.html, 2023.

[17] TLSNotary. *TLSNotary - a mechanism for independently audited https sessions* https://tlsnotary.org/TLSNotary.pdf, 2014.

[18] zkPass Team. *zkPass Protocol based on TLS, MPC and ZKP: Technical Whitepaper 2.0* https://docsend.com/view/5wdg66beu7m95jf3, 2023.

[19] Reclaim Protocol Team. *Reclaim Protocol: Claiming and Managing Self-Sovereign Credentials* https://drive.google.com/file/d/1wmfdtIGPaN9uJBI1DHqN903tP9c_aTG2/view, 2023.

[20] Dimitriou, T. *Decentralized reputation* IEEE, 2020.

[21] Hasan R., Sion R., and Winslett M. *The Case of the Fake Picasso: Preventing History Forgery with Secure Provenance* in Proceedings of the 7th USENIX Conference on File and Storage Technologies (FAST '09), 1-14, USENIX, http://usenix.org/event/fast09/tech/full_papers/hasan/hasan.pdf, 2009.

[22] Westerlund M. *The Emergence of Deepfake Technology: A Review* Technology Innovation Management Review, 9(11), 39-52, http://doi.org/10.22215/timreview/1282, 2019.

[23] Brundage M., Avin S., Wang J., et al. *Toward Trustworthy AI Development: Mechanisms for Supporting Verifiable Claims* arXiv:2004.07213 [cs.CY], https://arxiv.org/abs/2004.07213, 2020.

[24] Lanier, J. *Who owns the future?* Simon and Schuster, 2013.

[25] Arrieta Ibarra, I., et al. *Should we treat data as labor? Moving beyond "free"* AEA Papers & Proceedings, 108, 38-42, https://www.aeaweb.org/articles?id=10.1257/pandp.20181003, 2018.

[26] Ghorbani, A., and Zou, J. *Data Shapley: Equitable valuation of data for machine learning* Proceedings of the 36th International Conference on Machine Learning, PMLR, 97, 2242-2251, https://proceedings.mlr.press/v97/ghorbani19c.html, 2019.

[27] Pandl K., Huang C., Beschastnikh I., Li X., Thiebes S., and Sunyaev A. *Scalable Data Point Valuation in Decentralized Learning* arXiv:2305.01657 [cs.LG], https://doi.org/10.48550/arXiv.2305.01657, 2023.

[28] Li, P. *Proof of Training (PoT): Harnessing Crypto Mining Power for Distributed AI Training* arXiv:2307.07066 [cs.CR], https://doi.org/10.48550/arXiv.2307.07066, 2023.

[29] Truex, S., Baracaldo, N., Anwar, A. et al. *A Hybrid Approach to Privacy-Preserving Federated Learning* Informatik Spektrum, 42, 356-357, https://doi.org/10.1007/s00287-019-01205-x, 2019.

[30] Nakajima Y. *BabyAGI* https://github.com/yoheinakajima/babyagi.

[31] *AutoGPT* https://github.com/Significant-Gravitas/AutoGPT.

[32] *GPT-Engineer* https://github.com/gpt-engineer-org/gpt-engineer.

[33] *LangChain* https://github.com/langchain-ai.

[34] *Chainlink Functions* https://chain.link/functions

[35] Bellavitis, C., Fisch, C., and Momtaz, P. *The rise of decentralized autonomous organizations (DAOs): a first empirical glimpse* Venture Capital, 25, 1-17, https://doi.org/10.1080/13691066.2022.2116797, 2022.

[36] Eigen, D., Ranzato, M., Sutskever, I. *Learning Factored Representations in a Deep Mixture of Experts* arXiv:1312.4314 [cs.LG], https://doi.org/10.48550/arXiv.1312.4314, 2013.

[37] Kruijssen, J., Emmons, N., Peluso, K., Ghaffar F., Huang, A., and Kell T. *Allora: a Self-Improving, Decentralized Machine Intelligence Network* https://whitepaper.assets.allora.network/whitepaper.pdf, 2024.

[38] Korinek, A., and Stiglitz, J. E. *Artificial intelligence and its implications for income distribution and unemployment* in The economics of artificial intelligence: An agenda, 349-390, University of Chicago Press, 2021.

[39] Abusalah H., Alwen J., Cohen B., Khilko D., Pietrzak K., and Reyzin L. *Beyond Hellman's Time-Memory Trade-Offs with Applications to Proofs of Space* ASIACRYPT 2017, 357–379, https://ia.cr/2017/893, 2017.

[40] Lerner, S. D. *Proof of unique blockchain storage* Bitslog, https://bitslog.com/2014/11/03/proof-of-local-blockchain-storage/, 2014.

[41] Luu, L., Teutsch, J., Kulkarni, R., and Saxena, P. *Demystifying incentives in the consensus computer* CCS '15: Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security, 706–719, http://dx.doi.org/10.1145/2810103.2813659, 2015.

[42] Park S., Kwon A., Fuchsbauer G., Gazi P., Alwen J., and Pietrzak K. *Spacemint: A cryptocurrency based on proofs of space* 22nd International Conference on Financial Cryptography and Data Security, 480–499, https://ia.cr/2015/528, 2018.

[43] Cohen B. and Pietrzak K. *The chia network blockchain* https://chia.net/wp-content/uploads/2022/07/ChiaGreenPaper.pdf, 2019.

[44] Moran, T. and Orlov, I. *Simple proofs of space-time and rational proofs of storage* Advances in Cryptology – CRYPTO 2019 Annual International Cryptology Conference, 381–409, https://eprint.iacr.org/2016/035.pdf, 2019.

[45] Kate A., Zaverucha G., and Goldberg I. *Polynomial commitments* Technical report, Centre for Applied Cryptographic Research, University of Waterloo, https://cacr.uwaterloo.ca/techreports/2010/cacr2010-10.pdf, 2010.

[46] Li, J., and Li, B. *Erasure coding for cloud storage systems: A survey* Tsinghua Science and Technology, 18(3), 259-272, https://iqua.ece.toronto.edu/papers/junli-survey13.pdf, 2013.

[47] Nazirkhanova, K., Neu, J., and Tse, D. *Information Dispersal with Provable Retrievability for Rollups* arXiv:2111.12323 [cs.CR], https://doi.org/10.48550/arXiv.2111.12323, 2022.

[48] Dummy, J. *SSD VS HDD Power Consumption Chart & Calculation* https://computerhardwareparts.com/ssd-vs-hdd-power-consumption/, 2023.

[49] Bagaria V., Dembo A., Kannan S., Oh S., Tse D., Viswanath P., Wang X., and Zeitouni O. *Proof-of-Stake Longest Chain Protocols: Security vs Predictability* arXiv:1910.02218 [cs.CR], https://doi.org/10.48550/arXiv.1910.02218, 2020.

[50] Deb, S., Kannan, S., and Tse, D. *PoSAT: Proof-of-Work Availability and Unpredictability, without the Work* arXiv:2010.08154 [cs.CR], https://doi.org/10.48550/arXiv.2010.08154, 2021.

[51] David B., Ga ̆zi P., Kiayias A., and Russell A. *Ouroboros Praos: An adaptively-secure, semi-synchronous proof-of-stake blockchain* EUROCRYPT 2018 Annual International Conference on the Theory and Applications of Cryptographic Techniques, 66-98, https://ia.cr/2017/573, 2018.

[52] Buterin, V. and Griffith V. *Casper the friendly finality gadget* arXiv:1710.09437 [cs.CR], https://doi.org/10.48550/arXiv.1710.09437, 2017.

[53] Al-Bassam M., Sonnino, A., and Buterin V. *Fraud proofs: Maximising light client security and scaling blockchains with dishonest majorities* arXiv:1809.09044 [cs.CR], https://doi.org/10.48550/arXiv.1809.09044, 2018.

[54] Buterin V. *Slasher: A punitive proof-of-stake algorithm* Ethereum Foundation Blog, https://blog.ethereum.org/2014/01/15/slasher-a-punitive-proof-of-stake-algorithm, 2014.

[55] Hentschel A., Shirley, D., and Lafrance, L. *Flow: Separating consensus and compute* arXiv:1909.05821 [cs.DC], https://doi.org/10.48550/arXiv.1909.05821, 2019.

[56] Hentschel, A., Hassanzadeh-Nazarabadi, Y., Seraj, R., Shirley, D., and Lafrance, L. *Flow: Separating consensus and compute – block formation and execution* arXiv:2002.07403 [cs.DC], https://doi.org/10.48550/arXiv.2002.07403, 2020.

[57] Hentschel A., Shirley, D., Lafrance, L. and Zamski, M. *Flow: Separating consensus and compute – execution verification* arXiv:1909.05832 [cs.DC], https://doi.org/10.48550/arXiv.1909.05832, 2019.

[58] Teutsch, T. and Reitwießner C. *A scalable verification solution for blockchains* arXiv:1908.04756 [cs.CR], https://doi.org/10.48550/arXiv.1908.04756, 2017.

[59] Al-Bassam, M., Sonnino, A., Bano, S., Hrycyszyn, D. and G. Danezis *Chainspace: A sharded smart contracts platform* arXiv:1708.03778 [cs.CR], https://doi.org/10.48550/arXiv.1708.03778, 2017.

[60] Al-Bassam, M. *LazyLedger: A distributed data availability ledger with client-side smart contracts* arXiv:1905.09274 [cs.CR], https://doi.org/10.48550/arXiv.1905.09274, 2019.

[61] Bagaria V., Kannan S., Tse D., Fanti G., and Viswanath P. *Prism: Deconstructing the Blockchain to Approach Physical Limits* CCS '19: Proceedings of the 26th ACM SIGSAC Conference on Computer and Communications Security, 585–602, https://doi.org/10.1145/3319535.3363213, 2019.

[62] Kokoris-Kogias, E., Jovanovic, P., Gasser, L., Gailly, N., and Ford, B. *OmniLedger: A Secure, Scale-Out, Decentralized Ledger via Sharding*

Proceedings of the 2018 IEEE Symposium on Security and Privacy (SP), 583-598, https://eprint.iacr.org/2017/406.pdf, 2018.
[63] Buterin V. *How do layer 2s really differ from execution sharding?* https://vitalik.eth.limo/general/2024/05/23/l2exec.html, 2024.
[64] Zhang J., Luo Z., Ramesh R., and Kate A. *Sharding SMR with Optimal-size Shards for Highly Scalable Blockchains* arXiv:2406.08252 [cs.CR], https://doi.org/10.48550/arXiv.2406.08252, 2024.
[65] Tas E. N., and Boneh D. *Cryptoeconomic Security for Data Availability Committees* arXiv:2208.02999 [cs.CR], https://doi.org/10.48550/arXiv.2208.02999, 2023.

APPENDIX A
AUTO ID FRAMEWORK

A. Auto ID Creation and Registration

Auto ID is implemented as a domain containing a registry of autonomous identities. Users typically register their identity via self-issuing X.509 certificates, as illustrated in Fig. 10. The framework provides several ways to obtain an Auto ID, including self-issuance or issuance by another entity.

1) Self-issuance: A human user might decide to self-issue an Auto ID, for example, if they want to establish a distinct online identity and prove the authorship of their created content. In the future, this Auto ID will also serve as a foundation for issuing identities to dAgents that the user can delegate actions to and co-create content with. Using the Auto SDK, the user generates an Auto ID for themselves and registers this identity on an Autonomys Network ID domain, enabling easy verification of their identity and the content they create.

An entity can self-issue an Auto ID by following these steps:

1) Generate a cryptographic key pair, which consists of a private key and a public key.
2) Create a digital certificate that conforms to the X.509 standard. This certificate should include the public key and any relevant information or attributes about the entity.
3) Digitally sign the certificate using the entity's private key, which serves as proof that the entity has control over the Auto ID.
4) Register the Auto ID on an Autonomys Network identity domain.

2) Issuance by another entity: Users are also able to issue an Auto ID on behalf of other entities, creating a hierarchy of trust where the requesting entity inherits trust from the issuing entity. For example, a user wants to instantiate a dAgent to serve as a content co-creator. To ensure it can be trusted and its actions verified, the user issues an Auto ID to the dAgent using the Auto SDK, digitally signing it with their own Auto ID to establish a chain of trust. The dAgent's Auto ID is registered on an Autonomys Network identity domain, making it easily verifiable. When the dAgent performs actions on behalf of, or in concert with, the user, it digitally signs these actions using its Auto ID, allowing recipients to verify the authenticity and authorization of the its actions, establishing a clear record of collaboration and attribution for the work.

Another example of issuance by another entity is a user who wants to protect their digital content deciding to create an Auto ID using LetsID, a forthcoming free service provided by Autonomys Labs. By allowing LetsID to perform some form of automated identity verification, the user can quickly set up their Auto ID without the need to manage a blockchain wallet or own Auto Coin. Once their Auto ID is created and registered on an Autonomys Network identity domain, the user can employ it to digitally sign their content, creating a verifiable and tamper-proof record of authorship.

The process for issuance on behalf of another entity involves the following steps, as illustrated in Fig. 11:

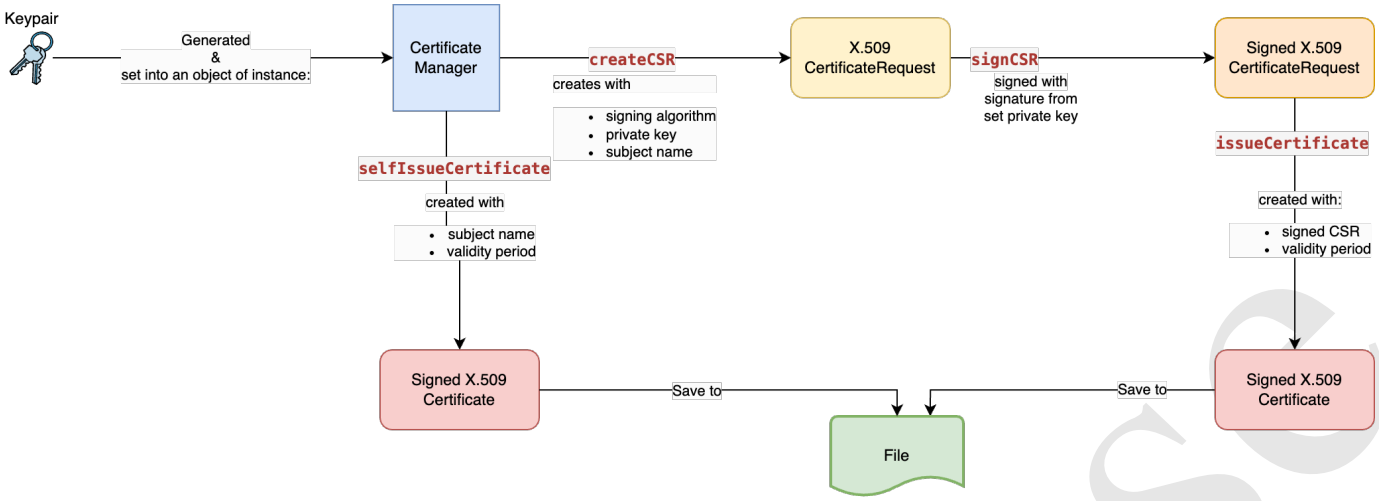

Fig. 10. Auto ID certificate generation workflow

1) The requesting entity generates their own cryptographic key pair, consisting of a private key and a public key.
2) The requesting entity submits a certificate signing request (CSR) to the issuing entity. The CSR includes the requesting entity's public key and any relevant information or attributes.
3) The issuing entity reviews the CSR and, if approved, digitally signs the certificate using its own private key. This signed certificate is then delivered to the requesting entity.
4) The requesting entity or a registrar then registers the Auto ID on an Autonomys Network identity domain.

3) Co-issuance by multiple entities: Multiple entities can jointly issue an Auto ID for a new entity. An example would be a user and an AI model developer jointly issuing an Auto ID for their own instance of an AI model.

B. Auto ID Verification and Authentication

1) Sign Data: An entity can digitally sign any data using the private key associated with their Auto ID, which includes a registered public key. This digital signature serves as a tamper-proof, verifiable proof of authorship, as it cryptographically binds the signed data to the entity's Auto ID.

2) Verify Signature: Any party can verify a signature using the entity's public key, confirming that the data was signed by the owner of the corresponding Auto ID, and that it has not been altered since the signature was applied. This process ensures the integrity and authenticity of the signed data.

Examples of signature verification include:

- A content creator signs their digital work using their Auto ID's private key before publishing it online. This signature serves as a tamper-proof record of the creator's authorship and ownership of the content. When other users encounter the signed content, they can use the public key of the creator's Auto ID to verify the signature, confirming that the content genuinely comes from the claimed creator and hasn't been altered by anyone else.
- In an effort to combat misinformation and fake news on the Internet, a renowned media organization implements Auto IDs for all their journalists and articles. When a journalist writes a new article, they sign the digital content using their Auto ID's private key before publication, and the organization prominently displays the journalist's Auto ID public key alongside the article. Readers can use the provided Auto ID public key to verify the signature, ensuring the article's authenticity and the journalist's credibility, regardless of where they encounter the content.

3) Verify Auto ID Registration and Validity: Any party can verify whether an Auto ID is registered and valid on an Autonomys identity domain with the relevant public key. This process is as follows:

1) *Retrieve the Auto ID from an Autonomys Domain*: To ensure its authenticity and integrity, retrieve the Auto ID directly from an Autonomys domain and verify that it has not been altered.
2) *Verify the Subject's Public Key*: Confirm that the public key within the Auto ID matches the expected public key of the entity.
3) *Check for Validity Period*: Review the certificate's validity dates ($notBefore$ and $notAfter$ fields) to ensure the Auto ID is currently valid and hasn't expired.
4) *Check Revocation Status:* Verify that the Auto ID is not on the on-chain Certificate Revocation List (CRL), indicating it hasn't been revoked.
5) *Verify the Chain of Trust*:
   - *Certificate Registration*: For each certificate in the trust chain (including intermediate and root certificates), verify its registration and authenticity on the blockchain.
   - *Validity Periods*: Ensure every certificate in the chain, up to and including the root certificate, is within its validity period and not expired.

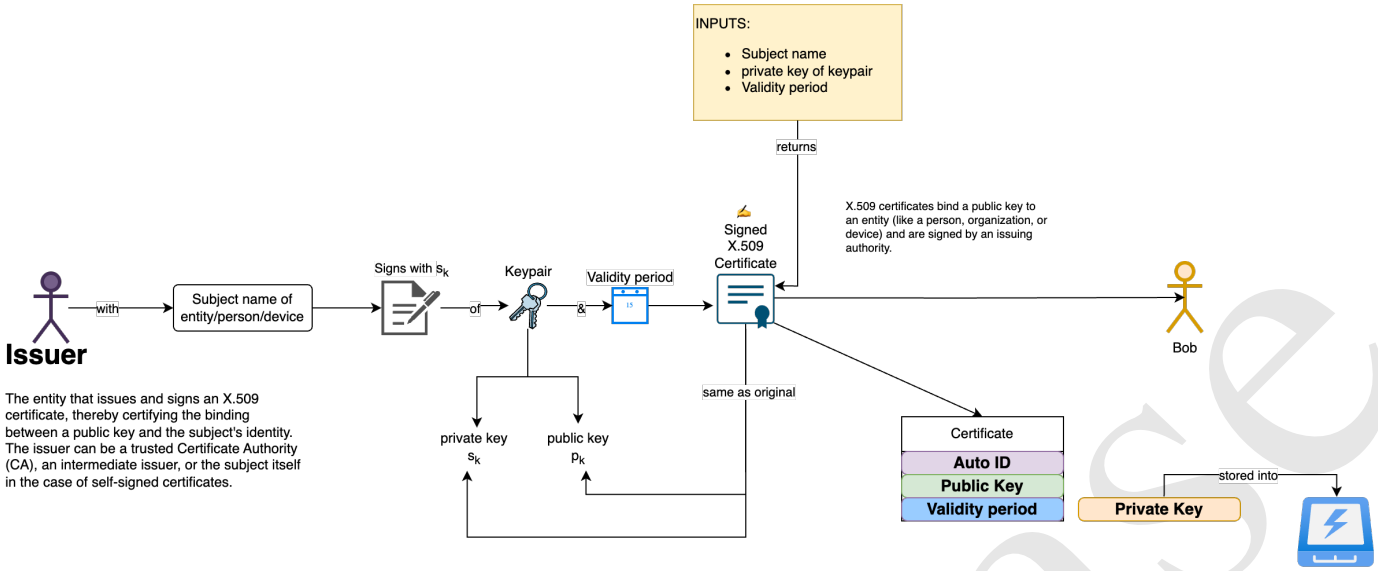

Fig. 11. Auto ID issuance by another entity

- *Revocation Status*: Check the CRL for each certificate in the chain to ensure none have been revoked.
- *Signature Verification*: For each certificate in the chain, verify that its signature is valid. This involves using the issuing certificate's public key to validate the signature of a certificate, and ensuring each is correctly signed by its issuer. This process confirms the integrity and authenticity of the certificate chain up to a trusted root certificate.

6) *Trusted Root Certificates*: The verification process should conclude with a certificate that is inherently trusted by the verifying party (a trusted root). This could be a root certificate that is widely recognized, or specifically trusted by the verifier.

4) Authentication: Entities can mutually authenticate using mutual Transport Layer Security (mTLS) if they want to communicate in a secure manner, as follows:

1) When initiating a secure connection, both entities present their Auto ID certificates to each other during the TLS handshake process.
2) Each entity verifies the other's certificate by checking its digital signature, ensuring it was issued by a trusted authority or entity.
3) If the certificates are valid, the entities use their private keys to establish a secure, encrypted connection.
4) Throughout the session, all data exchanged between the entities is encrypted using the established TLS connection, ensuring the confidentiality and integrity of their communication.

C. Auto ID Updates

Updates to an Auto ID are accomplished via signed transactions submitted to the Autonomys Network.

1) Revoke Keys: An entity can revoke keys from Auto IDs issued by them. They must prove ownership of the issuer public key via a digital signature verification process.

2) Recover: Any entity can recover control of an Auto ID should their existing controlling keys be lost. This can be accomplished via social recovery, a multi-signature scheme, Shamir secret sharing, or some other industry-standard mechanism.

3) Renew: An entity can update the validity period and public keys of an Auto ID by proving ownership of the public keys via a digital signature verification process.

4) Auto ID Deactivation: An entity can completely deactivate an existing Auto ID, rendering downstream processes inoperable, by submitting a deactivation extrinsic.

D. Claims Framework

A common application of identities is their use in verifiable credentials (VCs). If a VC is issued to an identity, as opposed to a public key, it allows the holder of the VC to rotate their keys without needing to renew their VC.

For example, let's assume a user has a self-issued X.509 certificate on an Auto ID domain, claiming the identity $example\_user\_id$. If the owner of that Auto ID received a diploma in the form of a VC issued to $example\_user\_id$, the VC Issuer of the diploma would need to confirm the correct entity controls $example\_user\_id$ before issuing the diploma.

1) Definitions:

- *Verifiable Credential (VC)* — A standardized way to represent claims made by an issuer about a subject in a way that can be cryptographically verified.
- *VC Issuer* — The person or organization that creates and issues the VC (usually a certified or professional organization).
- *VC Subject / Holder* — The person the VC is issued to.
- *VC Verifier* — The person or organization that verifies the VC when it's presented to them by a VC Subject (can be anyone).

2) Issuance: The following example illustrates the execution flow of a VC Issuer (University) issuing a VC (Diploma) to a user (student, Alice) who has an Auto ID:

1) Let's assume that Alice has already self-issued an X.509 certificate to her signing key pair and registered it on the Autonomys Network, and that the University already has a signing key pair.
2) Alice finishes her courses. Using her school's secure communication channel, in which she is already identified and authenticated, Alice formally requests her VC Diploma be issued to her Auto ID.
3) Once the University receives Alice's formal request, they verify that Alice has graduated (through the secure communication channel), and that they have not previously issued her a VC Diploma (by checking their internal logs). If they have, and the VC has not expired or been revoked, the University returns the previously issued VC Diploma. If it's expired, the University continues with the next steps. The University may also include an option for Alice to request to revoke her previously issued VC Diploma in order to be reissued a new VC Diploma under a different Auto ID. These options are all customizable at the University's discretion.
4) After verifying the claim that Alice graduated, the University verifies that Alice controls the relevant Auto ID by asking her to produce a digital signature that corresponds to the X.509 associated with her Auto ID. In order to prevent replay and phishing attacks, the University sends Alice a random nonce as well as any metadata that Alice should sign (such as $request\_date$, $nature\_of\_request$, etc).
5) Alice digitally signs the concatenation of the random nonce, her Auto ID, and the metadata to prove she owns the signing key pair associated with her registered X.509.
6) Upon receiving Alice's response to the challenge, the University verifies that her signature utilizes the same signing key pair as her Auto ID's registered X.509.
7) In order to prevent Sybil attacks, the University stores a copy of this whole exchange, including the issued VC Diploma and other metadata.

3) Revocation: The following illustrates the execution flow of a VC Issuer revoking a previously issued VC, using the same example of University student Alice:

1) Let's assume Alice cheated on all of her exams, and the University only discovered this after issuing her VC Diploma. They now want to revoke it.
2) The University records this revocation in their internal logs.
3) The University posts the revocation to a revocation list on the Autonomys Network. They must post a hash of the issued VC Diploma, and include the digital signature that corresponds to it, to prove they possess the power to revoke the VC Diploma.
4) This revocation list acts as a public place for VC Verifiers (e.g., an employer verifying a VC Diploma) to verify that the provided VC (e.g., Alice's VC Diploma) was not revoked.

4) Verification: The following illustrates the execution flow of a VC Verifier verifying a previously issued VC, where the Verifier is Alice's future employer, and Alice has been issued a VC Diploma from a University (the VC Issuer), that has not been revoked:

1) Alice's employer requests her VC Diploma and issues a challenge with a random nonce.
2) Having locally stored her VC Diploma, Alice signs and submits it together with the random nonce, proving that she has control of the identity on the VC.
3) The employer reads the VC Diploma, and verifies that the VC Issuer is a legitimate entity, and that Alice's signature corresponds to the current (unexpired) X.509 signing key pair registered on the relevant Auto ID domain.
4) The employer performs a look-up to ensure this VC is not in the revocation list on the relevant Auto ID domain.
5) Alice has now been vetted.