ZKP Whitepaper

An Introduction to Distributed AI Compute Secured by Zero-Knowledge Proofs

April 15, 2025

Contents

Abstract

The Zero-Knowledge Proof (ZKP) ecosystem proposes a novel approach to decentralized artificial intelligence (AI), exploring the intersection of distributed compute, data sovereignty, and cryptographic security within the framework of Decentralized Physical Infrastructure Networks (DePIN).

This paper presents an architectural model comprising two complementary components: a blockchain infrastructure and a privacy-preserving data marketplace.

The blockchain layer investigates a hybrid consensus model combining Proof of Intelligence (PoI), which cryptographically verifies the execution of AI computation tasks through zero-knowledge circuits, with Proof of Space (PoSp), which validates storage commitments, implemented as custom pallets within Substrate's BABE+GRANDPA consensus framework. This approach aims to facilitate secure, verifiable, and scalable AI computations while addressing computational and storage resource allocation.

The ZKP Data Marketplace, the principal decentralized application within this ecosystem, examines potential solutions to limitations in centralized data systems—including unauthorized exploitation, inadequate contributor compensation, and privacy vulnerabilities—as evidenced by incidents such as the Cambridge Analytica scandal [76].

Through integration of zero-knowledge proofs (ZKPs), including zk-SNARKs and zk-STARKs, both components seek to maintain confidentiality and integrity of datasets and large language models (LLMs), potentially enabling privacy-preserving data sharing, monetization, and model training without compromising efficiency or sovereignty. The proposed model suggests contributors could retain ownership of tokenized data assets while verification occurs without exposure of underlying information, presenting an alternative economic model currently under evaluation in testnet environments.

The platform's distributed compute framework conceptualizes a global marketplace for AI-driven tasks through off-chain storage, multi-runtime environments (EVM/WASM), and modular design. Built on Substrate's modular pallet architecture, the system leverages Frontier's EVM compatibility layer while maintaining native Substrate functionality. This whitepaper examines the theoretical underpinnings, technical architecture, and potential socio-economic implications of ZKP, proposing an approach to secure, scalable AI infrastructure that could contribute to broader access to privacy-preserving computation.

The ecosystem described represents a research direction currently in early exploratory and development phases, with key components undergoing preliminary testnet evaluation.

Preface

The evolution of artificial intelligence (AI) and blockchain technology has revealed significant challenges related to trust, equity, and security in digital ecosystems. A small number of corporations currently dominate AI development, controlling extensive datasets and large language models (LLMs), thereby limiting innovation [1]. This centralization manifests in multiple dimensions, creating a digital landscape with challenges extending beyond mere technical considerations.

Contemporary data management systems, operated by major technology companies, function through models that leave user data vulnerable while concentrating economic benefits primarily among intermediaries rather than data originators. High-profile incidents illustrate these vulnerabilities—the 2018 Cambridge Analytica scandal exposed 50 million Facebook profiles without explicit consent, while the 2017 Equifax breach compromised sensitive financial data of approximately 147 million individuals [95]. These cases represent merely visible manifestations of systemic issues inherent to centralized architectures.

This asymmetric value distribution has created profound economic imbalances where data originators—individuals, small businesses, and communities—remain largely uncompensated for their contributions to the digital economy. User profiles are routinely compiled, analyzed, and monetized without proportionate compensation or meaningful consent, creating repositories of sensitive information vulnerable to both malicious breaches and authorized yet ethically questionable exploitation.

The concentration of profits by dominant platforms has contributed to wealth concentration, limiting opportunities for equitable participation in digital innovation. Current AI platforms typically rely on centralized middleware with inherent limitations, lacking cryptographic verification between smart contracts, tokens, and AI models. These systems generally depend on traditional client-server architecture without on-chain verification mechanisms, creating user dependence on centralized infrastructure [2].

The absence of mechanisms to verify AI's origin, intent, or ownership undermines trust, as proprietary models often obscure their computational provenance. This opacity creates barriers to accountability, allowing biases, inaccuracies, or problematic elements to remain undetected within influential AI systems.

Beyond privacy and economic concerns, centralized frameworks struggle to balance utility with security. Single points of failure present substantial risks to data integrity, as evidenced by incidents like the Equifax breach where compromised infrastructure led to widespread exposure of sensitive information. Such vulnerabilities highlight the intrinsic characteristics of centralized architectures, where attack surfaces are concentrated and systemic risks amplified.

The erosion of public trust resulting from these incidents extends beyond individual platforms, potentially undermining confidence in digital innovation broadly. This trust deficit constrains the adoption and development of potentially beneficial technologies as users grow increasingly cautious about engaging with systems that prioritize data collection over protection. The resulting fragmentation of digital trust creates barriers to collaboration necessary for addressing complex global challenges that require coordinated data sharing and analysis.

Zero-knowledge proofs (ZKPs) represent a cryptographically verified method for secure, verifiable computations across diverse applications [30]. The implementation explored in this paper enables verification of dataset properties without exposing underlying data, a capability potentially valuable in contexts where data sensitivity is paramount, including medical research and financial analytics. This approach offers an alternative to conventional encryption by enabling proof-based verification without data exposure, allowing stakeholders to validate specific attributes of a dataset—such as its source, size, completeness, or statistical properties—without accessing the underlying information.

In medical research contexts, ZKPs could potentially enable verification that patient data

meets specific criteria for a study without exposing individual health records, potentially facilitating research while maintaining privacy standards. Similarly, in financial services, transaction patterns could be analyzed for fraud detection without exposing customer financial details, enhancing security while preserving confidentiality. These capabilities suggest possible shifts in how sensitive data could be utilized while mitigating exposure risks.

Scalability remains a limitation in current systems, as centralized architectures face challenges supporting distributed compute without compromising security or efficiency. The processing requirements of contemporary AI applications often exceed the capabilities of individual organizations, creating barriers to innovation for entities with limited resources. Distributed systems offer theoretical alternatives but have historically encountered coordination challenges, security considerations, and efficiency constraints that affect their practical implementation for compute-intensive applications.

The ZKP blockchain ecosystem addresses these challenges through its integrated architecture. Built on Substrate's modular framework, the system implements custom consensus pallets that extend beyond traditional block production and finality mechanisms. Rather than utilizing only traditional consensus mechanisms, the hybrid model combines AI computation validation with storage verification through custom Substrate pallets, aiming to create an energy-efficient system that connects network security with practical utility. This approach explores economic models involving greater contributor participation, enabling participants to tokenize data into tradeable assets and potentially earn rewards through the native DataToken (DTK), currently under investigation in testnet phases.

By examining a decentralized approach to these multifaceted issues, the ZKP ecosystem aims to address limitations of centralized frameworks and explore foundations for an alternative data economy. The tokenization of data assets represents not merely a technical implementation but a conceptual reexamination of data ownership and value distribution, potentially creating economic opportunities for individuals and organizations currently underrepresented in the data economy.

For instance, a small business collecting customer feedback could potentially tokenize this information, list it on the marketplace, and receive compensation as researchers or marketers access it—while retaining ownership throughout the process. This model could theoretically benefit communities and regions with limited participation in digital markets by enabling engagement in global data exchanges without requiring extensive technical infrastructure or corporate intermediaries, potentially addressing inequities in digital economic participation.

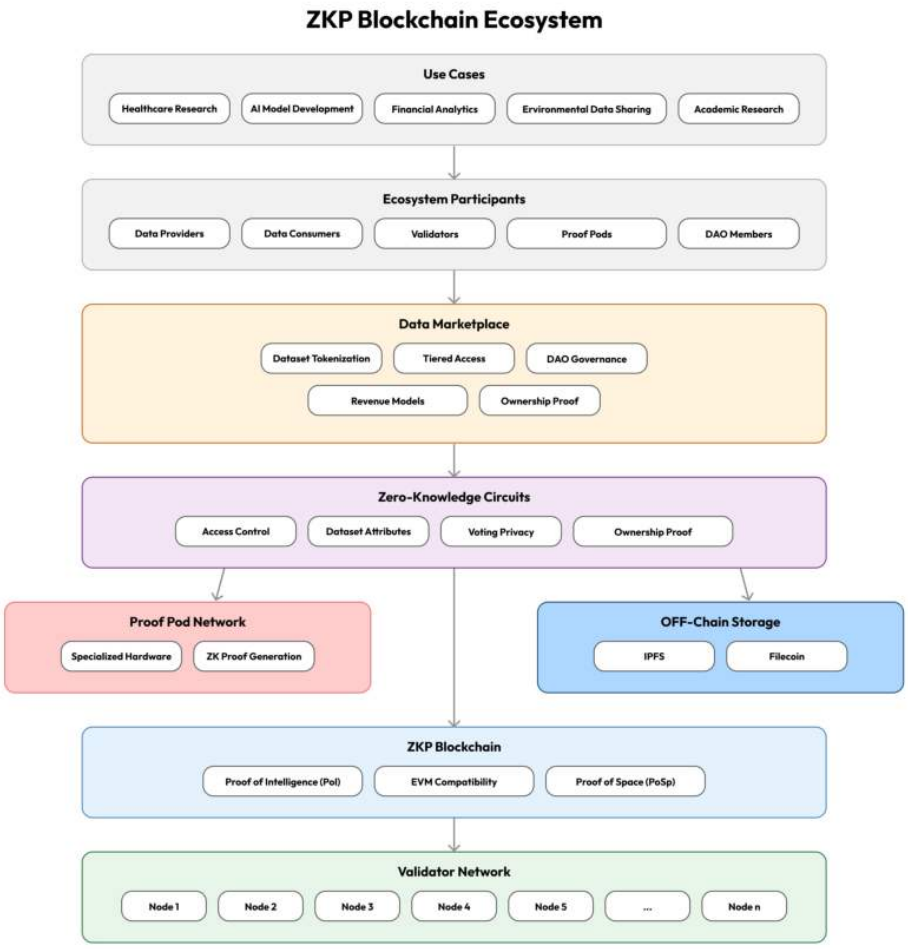

Figure 1: ZKP Ecosystem Diagram

Vision & Current Status

We envision a future where AI development becomes more distributed, data sovereignty is maintained, and computational resources are allocated across a network of participants. The ZKP ecosystem examines an alternative to centralized AI models through a collaborative framework where:

1. Data contributors could receive compensation for their contributions
2. Privacy-preserving computation methods could become more widely implemented
3. Innovations might emerge from a diverse global community rather than a limited number of organizations
4. Cryptographic verification could supplement trust in complex systems
5. Economic incentives could align with broader societal benefits and ethical AI development

This approach could influence digital economies, increasing participation from individuals and communities with limited access to AI benefits while encouraging transparency, accountability, and distributed value allocation. In domains such as healthcare, financial services, and public infrastructure, ZKP could enable collaboration while maintaining privacy standards, contributing to scientific research, addressing economic disparities, and enhancing public engagement. It is essential to note that ZKP is currently in an exploratory phase, with aspects of the architecture described in this paper representing research objectives rather than completed implementations. As of May 2025, the project has:

- Developed initial proof-of-concept implementations of key ZKP circuits
- Launched a preliminary testnet with limited functionality to evaluate core consensus mechanisms leveraging Substrate's BABE+GRANDPA framework with custom PoI and PoSp pallets
- Established a research partnership network to refine theoretical models
- Created early prototypes of the Data Marketplace with basic tokenization capabilities

The immediate development focus centers on systematic evaluation of fundamental components before expanding to more complex features. The architecture presented in this paper will likely evolve significantly as empirical testing provides new insights and challenges to theoretical frameworks. Throughout this paper, distinctions are made between currently implemented capabilities, short-term development objectives, and longer-term research directions to provide accurate expectations for potential ecosystem participants. The ecosystem implements a dual EVM/WASM runtime environment through Substrate's EVM pallet and native WASM execution, supporting various applications from Solidity smart contracts to performance-oriented AI workloads. The platform examines potential applications across multiple sectors, proposing an approach to decentralized computation where resources could be distributed and innovation occurs in an open framework. This whitepaper presents both the theoretical foundations and implementation methodology for the ZKP ecosystem as it progresses through testnet evaluation on Substrate's modular blockchain framework toward the research objective of privacy-focused AI secured by zero-knowledge technology.

ZKP Blockchain

Introduction

The advent of artificial intelligence (AI) has precipitated an unprecedented demand for computational resources [5], necessitating novel paradigms for distributed compute that transcend the limitations of centralized architectures. Concurrently, the proliferation of proprietary data models and large language models (LLMs) has underscored the imperative for stringent privacy and intellectual property (IP) protections [6], as well as the safeguarding of user data sovereignty. Traditional systems, constrained by their reliance on centralized control [7], are ill-equipped to address these exigencies, often compromising on security, scalability, and trust.

This paper introduces the Zero-Knowledge Proof (ZKP) ecosystem, a sophisticated blockchain framework underpinned by a dual consensus paradigm integrating Proof of Intelligence (PoI) and Proof of Space (PoSp), meticulously engineered to surmount these multifaceted challenges within the broader vision of Decentralized Physical Infrastructure Networks (DePIN).

Central to the ZKP ecosystem is its pioneering approach to distributed compute for AI, harnessing a decentralized constellation of nodes to execute parallelized AI workloads leveraging globally distributed physical resources—such as compute nodes and storage infrastructure—to create resilient, scalable networks.

Through the PoI mechanism, nodes contribute verifiable computational intelligence, enabling the efficient processing of complex AI tasks while circumventing the bottlenecks endemic to centralized infrastructures.

Complementing PoI, the Proof of Space (PoSp) mechanism leverages decentralized storage resources, ensuring that the network's data integrity and availability are maintained through verifiable commitments of storage space. This dual consensus model optimizes both computational and storage resources, creating a robust and balanced ecosystem that not only amplifies computational scalability but also cultivates a meritocratic environment wherein nodes are rewarded commensurate with their contributions to AI-driven endeavors.

The sanctity of proprietary data models and IP rights is preserved through the deployment of Zero-Knowledge Proofs (ZKPs) [8] [9] [30], notably zk-SNARKs and zk-STARKs, which facilitate verifiable computations without exposing sensitive inputs or model parameters. This cryptographic scaffold ensures that collaborative AI development can flourish without jeopardizing the confidentiality of proprietary algorithms or datasets, thereby reconciling the perennial dichotomy between innovation and IP stewardship. Similarly, the privacy of user data models is fortified by ZKPs, enabling computations on encrypted datasets such that user inputs remain obfuscated throughout the processing continuum. This privacy-preserving paradigm aligns with emergent regulatory frameworks while fostering user trust through assured data sovereignty.

The ZKP ecosystem's commitment to verifiable compute further manifests in the validation of AI outputs, wherein ZKPs furnish succinct proofs of correctness without divulging the underlying computational intricacies, thereby bolstering transparency and auditability within decentralized contexts.

Security constitutes a foundational pillar of the ZKP ecosystem, achieved through an amalgamation of cryptographic primitives, including secure multi-party computation (MPC) and homomorphic encryption [10] [11], which collectively safeguard the integrity and confidentiality of all operations. The dual consensus model, comprising PoI and PoSp, augments this security edifice by mandating that nodes substantiate both their computational prowess and storage commitments, thereby fortifying the network against adversarial incursions. Scalability, a perennial conundrum in decentralized frameworks, is addressed through the ZKP ecosystem's modular architecture and off-chain storage integrations, such as IPFS [12] and Filecoin [13], which mitigate on-chain congestion while ensuring data availability and redundancy.

The ZKP ecosystem is constructed upon Substrate [107] [108], a modular blockchain development framework that facilitates the creation of application-specific blockchains through its

flexible pallet architecture. This choice enables the seamless integration of our hybrid consensus model through custom pallets that synergize PoI and PoSp with Substrate's native BABE+GRANDPA consensus [109] [110] to optimize both computational and storage resources. Substrate's inherent modularity through its FRAME pallet system [111] allows for the tailoring of the blockchain's architecture to meet the unique demands of distributed AI compute, ensuring optimal performance and adaptability. Substrate's WebAssembly-based runtime enables forkless upgrades and cross-platform compatibility, while the EVM pallet [112] provides seamless Ethereum compatibility through the Frontier compatibility layer [113]. Central to this ecosystem is a decentralized data marketplace [16], which empowers users to securely share and monetize proprietary datasets and AI models. Leveraging ZKPs, the marketplace ensures that data transactions are private and verifiable, protecting intellectual property while fostering a collaborative environment. This marketplace is designed to be equitable, providing opportunities for both large and small contributors to participate in the AI economy, thereby addressing the economic disparities prevalent in centralized systems.

In essence, the ZKP ecosystem heralds a transformative convergence of AI and blockchain technology, proffering a decentralized, secure, and scalable substrate for distributed compute, data privacy, and verifiable intelligence. Through its dual consensus framework, cryptographic innovations, and DePIN integration, ZKP redefines the contours of AI computation, data sovereignty, and market equity, presaging a new epoch of decentralized innovation.

It is important to note that the ZKP ecosystem described in this paper represents an ambitious vision currently in active research and development. While foundational components are being designed and prototyped, many of the capabilities outlined here remain in early development stages and will require significant engineering work to realize fully.

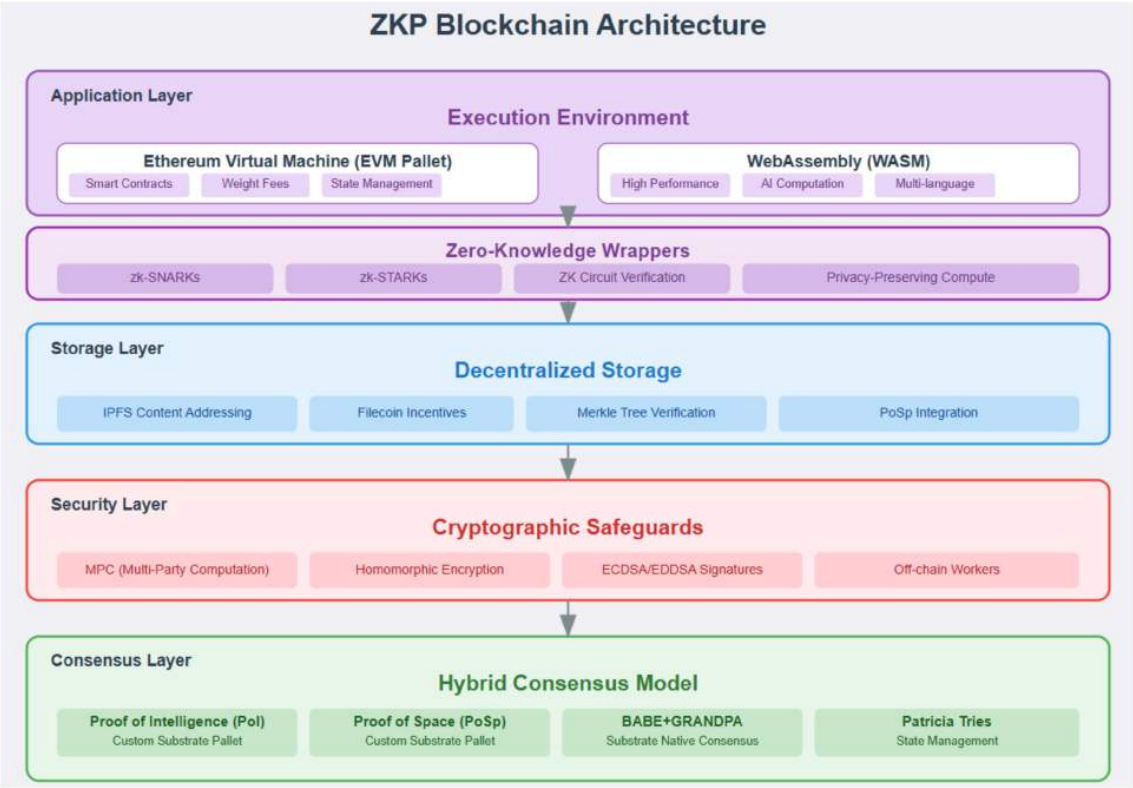

Figure 2: ZKP Blockchain Architecture

Before continuing, we will delve into the basic concepts that underpin the ZKP ecosystem's design; if you are already familiar with these principles, feel welcome to skip ahead to the ZKB

Base Layer section.

Core Concepts

The Zero-Knowledge Proof (ZKP) ecosystem combines cryptography and consensus to support distributed AI compute, privacy, and scalability.

This section outlines its key components: Zero-Knowledge Proofs (ZKPs), Proof of Intelligence (PoI), Proof of Space (PoSp), and foundational cryptographic and blockchain technologies that underpin the ecosystem's architecture.

Zero-Knowledge Proofs (ZKPs)

Zero-Knowledge Proofs (ZKPs) are cryptographic protocols where a prover convinces a verifier that a statement is true without revealing any additional information beyond the validity of the statement itself [8].

Formally, ZKPs satisfy three properties: completeness (true statements can be proven), soundness (false statements cannot be proven), and zero-knowledge (no information beyond validity is revealed). This balance of privacy and verifiability is vital for secure AI operations, particularly in scenarios where sensitive data must remain confidential while ensuring computational integrity.

For example, Alice wants to prove to Bob she trained an AI model on a specific dataset without revealing it. Using ZKPs, she generates a proof of the training's correctness, which Bob verifies without seeing the data. This ensures that proprietary information remains protected while still allowing for trust in the model's validity. Such a mechanism is critical in collaborative AI environments where multiple parties need assurance without compromising their data.

Techniques like zk-SNARKs [9] [30] and zk-STARKs ensure efficient, private computation validation in the ZKP ecosystem, leveraging advanced cryptographic methods to achieve succinct and scalable proofs. These methods reduce the computational overhead, making ZKPs practical for real-world blockchain and AI applications.

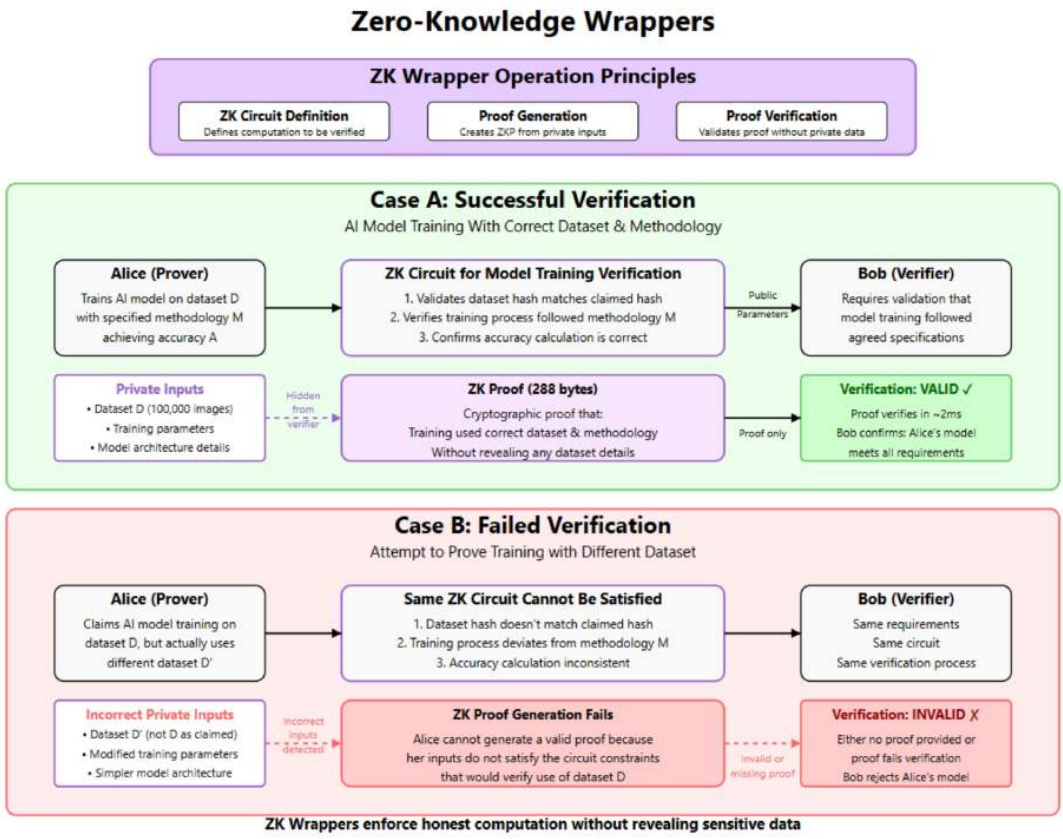

Figure 3: Zero-Knowledge Wrappers Diagram

Proof of Intelligence (PoI)

Proof of Intelligence (PoI) uses AI task contributions (e.g., model training) for network consensus. Nodes perform verifiable AI workloads, proven via ZKPs, enhancing security and AI progress. Implemented as a custom Substrate pallet, PoI integrates seamlessly with the BABE+GRANDPA consensus framework [109] [110] to validate computational contributions alongside traditional block production and finality mechanisms.

By tying consensus to meaningful AI computations, PoI aligns network incentives with the advancement of AI capabilities, fostering a decentralized environment where computational resources are efficiently utilized. This approach contrasts with traditional proof-of-work systems by prioritizing useful work over redundant computation.

The modular design of Substrate's pallet system [111] allows for flexible integration of PoI mechanics with other consensus and governance mechanisms.

Proof of Space (PoSp)

Proof of Space (PoSp) leverages decentralized storage for consensus [22]. Nodes provide verifiable storage, proven with cryptographic methods, boosting data availability and scalability for AI needs. Implemented through Substrate's flexible runtime architecture [107] [108], PoSp operates as a complementary pallet that validates storage commitments while maintaining compatibility with the broader ecosystem. PoSp ensures that the network can handle large datasets required for AI applications, distributing storage across nodes to prevent centralization and enhance resilience. This makes it an ideal complement to AI systems that demand vast amounts of data for training and inference.

Furthermore, PoSp reduces reliance on energy-intensive processes, offering a more sustainable alternative to traditional consensus mechanisms like PoW [35]. Its design supports long-term scalability by adapting to growing storage demands in AI-driven ecosystems. The pallet's integration with Substrate's WebAssembly runtime enables efficient storage verification while maintaining the system's upgradeability and cross-platform compatibility.

Merkle Trees

A Merkle Tree is a binary tree data structure where each leaf node contains a hash of a data block, and each non-leaf node is the hash of its child nodes [18]. This hierarchical organization allows for efficient verification of data integrity and membership. By comparing the Merkle root—a single hash summarizing all the data—with a small set of intermediate hashes (a Merkle proof), one can confirm that a specific piece of data belongs to the tree without needing the entire dataset. In blockchain contexts, Merkle Trees are employed to compactly represent transaction data within blocks, enabling light clients to verify transactions with minimal data.

In the ZKP ecosystem, Merkle Trees are essential for maintaining trust and efficiency, particularly in the Storage Layer. With off-chain storage solutions like IPFS, Merkle Trees ensure that large datasets remain tamper-proof by anchoring their root hash on-chain. For instance, when a node contributes storage for AI datasets under the PoSp mechanism, a Merkle proof can verify that the stored data matches the committed hash, ensuring data availability and integrity. This capability is critical for decentralized AI applications where trust in off-chain data is paramount.

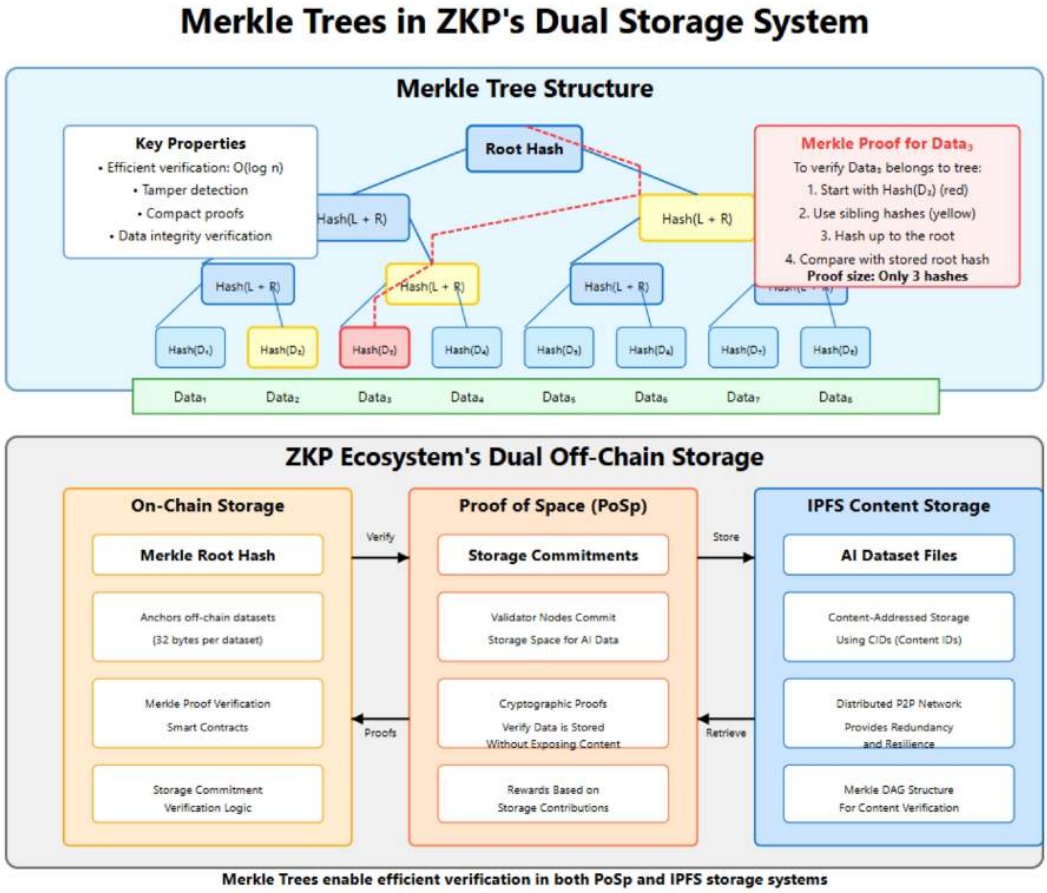

Figure 4: Merkle Trees

Patricia Tries

Patricia Tries (Practical Algorithm to Retrieve Information Coded in Alphanumeric) are radix trees optimized for blockchain state storage, serving as the fundamental data structure in Substrate's state management system [114]. These compressed trie structures store key-value pairs efficiently, with path compression reducing storage overhead while maintaining fast lookup times of O(k) where k is the key length. Patricia Tries enable efficient state queries and provide cryptographic commitments to the entire state through their root hash.

Within the ZKP ecosystem, Patricia Tries are utilized in Substrate's runtime to manage the blockchain's dynamic state, including PoI scores, staking power, and PoSp commitments [107] [108]. For example, when a node submits an AI task result for PoI, the state update is recorded in the Patricia Trie, and a Merkle proof can validate this contribution across the network. This ensures transparency and auditability, reinforcing the reliability of the hybrid consensus model in supporting decentralized AI compute. The trie's structure enables efficient state queries and historical state access, crucial for ZKP verification and data marketplace operations.

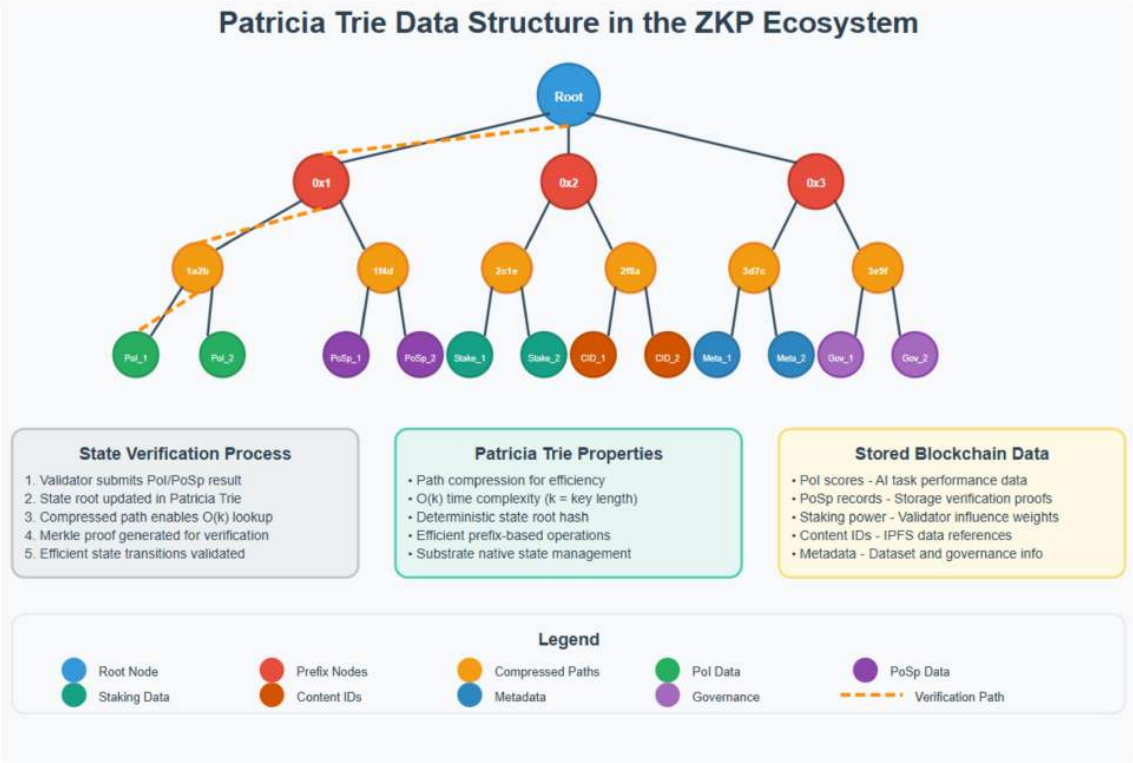

Figure 5: Patricia Tries

Substrate Framework

Substrate is a modular blockchain development framework created by Parity Technologies, providing a comprehensive suite of tools and libraries to streamline custom blockchain development [107] [108]. It handles core blockchain functionalities—such as transaction processing, staking, and governance—through its FRAME (Framework for Runtime Aggregation of Modularized Entities) system, while allowing developers to create application-specific pallets that define custom blockchain logic.

Built on a hybrid consensus model combining BABE (Blind Assignment for Blockchain Extension) for block production and GRANDPA (GHOST-based Recursive Ancestor Deriving Prefix Agreement) for finality [109] [110], Substrate offers fast transaction finality and native

interoperability through the Cross-Chain Message Passing (XCMP) protocol within the Polkadot ecosystem [115]. The framework's WebAssembly-based runtime enables forkless upgrades and cross-platform compatibility, making it a versatile foundation for scalable blockchain networks.

In the ZKP ecosystem, Substrate underpins the blockchain's architecture, enabling the implementation of the hybrid PoI-PoSp consensus model through custom pallets. Custom pallets, such as pallet-poi for AI task management and pallet-posp for storage verification, integrate seamlessly with off-chain systems like IPFS through Substrate's flexible runtime architecture [111]. This modularity supports the ecosystem's focus on decentralized AI by allowing tailored solutions for privacy-preserving computation and data handling, while maintaining compatibility with the broader Polkadot ecosystem and enabling seamless interoperability with other Substrate-based chains.

Ethereum Virtual Machine (EVM)

The Ethereum Virtual Machine (EVM) is a decentralized runtime environment for executing smart contracts, primarily on the Ethereum blockchain [19]. It is Turing-complete, meaning it can run any program given sufficient resources, and operates within a sandboxed environment to ensure security. The EVM uses a gas model, where each operation incurs a cost, preventing infinite loops and incentivizing efficient code. Smart contracts, typically written in Solidity, enable programmable logic that executes automatically based on predefined conditions.

In the ZKP ecosystem, the EVM is incorporated into the Substrate-based chain through the EVM pallet and Frontier compatibility layer [112] [113], enabling native support for Solidity smart contracts while maintaining full Ethereum compatibility. This integration allows developers to deploy existing Ethereum dApps without modification, creating applications such as a decentralized marketplace for AI models where ZKPs verify transactions without exposing sensitive data. The EVM pallet seamlessly integrates with Substrate's native runtime, enabling unified account systems (H160 addresses) and efficient cross-runtime communication between EVM contracts and native Substrate pallets. The integration of ZK wrappers with the EVM ensures privacy-preserving execution, making it a powerful tool for bridging Ethereum's developer base with ZKP's advanced privacy features while leveraging Substrate's superior scalability and upgrade mechanisms.

ECDSA

Elliptic Curve Digital Signature Algorithm (ECDSA) is a cryptographic method for generating and verifying digital signatures, based on elliptic curve mathematics [20]. It offers high security with smaller key sizes (e.g., 256-bit keys) compared to traditional algorithms like RSA, making it computationally efficient. In blockchain systems, ECDSA is widely used to sign transactions, allowing the network to confirm a sender's identity and intent without revealing their private key.

In the ZKP ecosystem, ECDSA is the primary mechanism for transaction signing and node authentication within Substrate's runtime environment [107] [108]. For instance, when a node submits a PoI task result, it signs the submission with its private key, and the network verifies this signature using ECDSA to ensure authenticity. This process upholds the integrity of consensus and prevents fraudulent contributions, a cornerstone of trust in the decentralized network. Substrate's native support for ECDSA ensures seamless integration with both the EVM pallet for Ethereum compatibility and native Substrate pallets for custom functionality.

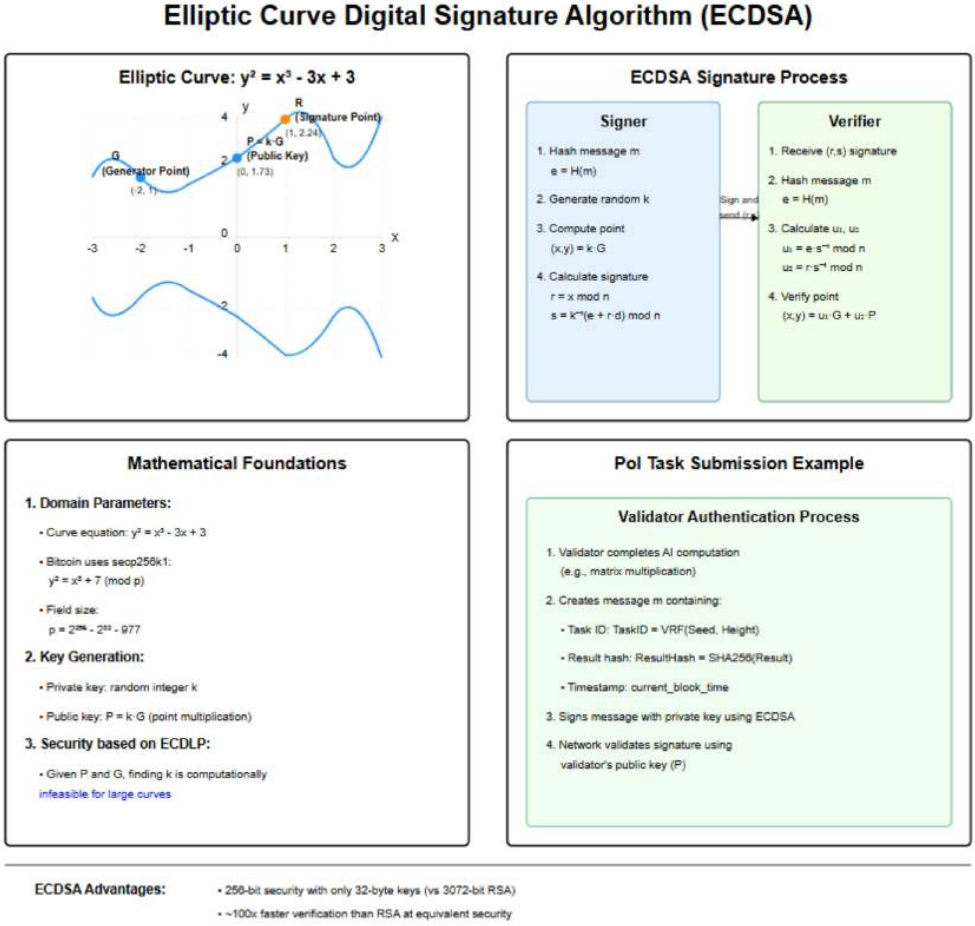

Figure 6: ECDSA

EDDSA

Edwards-curve Digital Signature Algorithm (EDDSA) is an advanced digital signature scheme that uses twisted Edwards curves, offering improved performance and security over ECDSA [21]. It is faster in signature generation and verification, resistant to certain side-channel attacks, and simpler to implement securely due to its deterministic design. EDDSA is gaining traction in modern cryptographic systems where efficiency and robustness are critical.

While ECDSA currently dominates in the ZKP ecosystem, EDDSA is under consideration for future upgrades due to its performance advantages. For example, in high-throughput scenarios like real-time AI task validation, EDDSA's speed could reduce latency. Substrate's modular pallet architecture [111] allows for flexible cryptographic implementations, enabling potential adoption of EDDSA through runtime upgrades without requiring hard forks. This upgradeability enhances security and scalability as demands evolve, leveraging Substrate's WebAssembly-based runtime for seamless cryptographic protocol transitions.

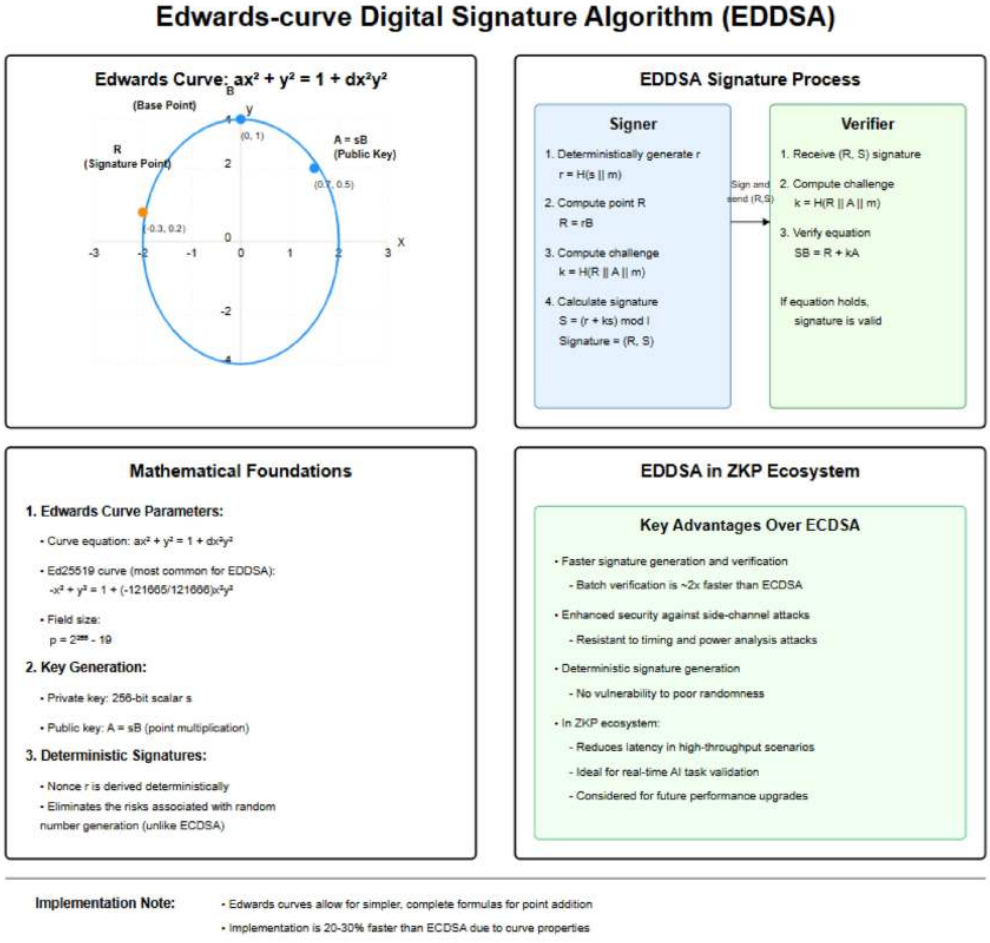

Figure 7: EDDSA

Homomorphic Encryption

Homomorphic Encryption enables computations on encrypted data without decrypting it, preserving privacy throughout the process [10]. The encrypted result, when decrypted, matches the outcome of the same computation on plaintext data. This is invaluable for applications requiring confidentiality, such as secure AI training, where data exposure must be minimized. However, its computational complexity—often orders of magnitude slower than plaintext operations—poses practical challenges.

In the ZKP ecosystem, Homomorphic Encryption complements ZKPs by offering an alternative for privacy-preserving computation, particularly in scenarios like federated learning. Nodes could process encrypted datasets for AI models, ensuring data remains confidential even during computation. While ZKPs handle most privacy needs currently, Homomorphic Encryption is a research focus for future enhancements, balancing its privacy benefits against performance trade-offs.

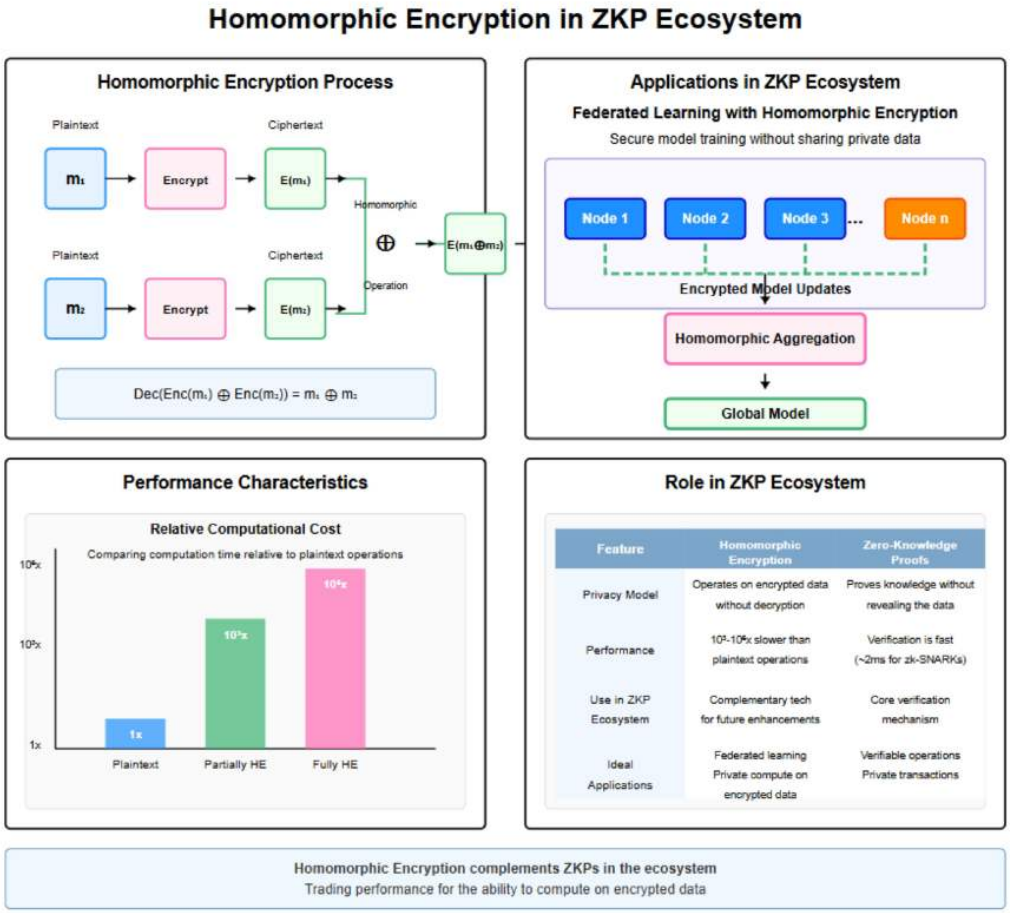

Figure 8: Homomorphic Encryption

zk-STARKs

zk-STARKs (Zero-Knowledge Scalable Transparent Arguments of Knowledge) offer similar privacy benefits to zk-SNARKs but with distinct advantages [23]. Unlike zk-SNARKs, zk-STARKs do not require a trusted setup, making them fully transparent and more secure against setup-related vulnerabilities.

Additionally, zk-STARKs are post-quantum secure, relying on hash functions rather than elliptic curves, which positions them as a future-proof solution. However, zk-STARKs come with trade-offs: their proofs are larger (e.g., ∼100KB) and verification is slower (e.g., ∼ 10ms) compared to zk-SNARKs.

In the ZKP ecosystem, zk-STARKs are explored for applications where transparency and quantum resistance are prioritized over proof size, such as off-chain computations or long-term data integrity proofs. For instance, zk-STARKs could be used to verify large-scale AI model training in a decentralized setting, ensuring the proof remains secure even as quantum computing threats emerge.

Usage of zk-SNARKs and zk-STARKs

To address the distinct roles of zk-SNARKs and zk-STARKs in the ZKP ecosystem, their applications are clarified based on their technical properties and trade-offs.

zk-SNARKs: On-Chain Verification

zk-SNARKs are employed for on-chain verification due to their efficiency in blockchain environments. Key advantages include:

- Compact Proof Size: Approximately 288 bytes, minimizing storage and bandwidth demands on-chain.
- Fast Verification: Around 2 milliseconds, enabling rapid transaction processing in decentralized systems.

These properties make zk-SNARKs ideal for applications requiring frequent, lightweight verifications, such as smart contract execution or transaction validation [30].

zk-STARKs: Off-Chain Computations

zk-STARKs are utilized for off-chain computations where additional security and transparency are prioritized. Their strengths include:

- No Trusted Setup: Unlike zk-SNARKs, zk-STARKs are fully transparent, avoiding potential vulnerabilities from setup ceremonies.
- Post-Quantum Security: Resistant to quantum computing attacks, ensuring long-term robustness.

Despite larger proof sizes (e.g., ∼100KB) and slower verification times (e.g., ∼10ms), zk-STARKs excel in scenarios like private off-chain computations or data aggregation where these trade-offs are acceptable [23].

Comparison Table: zk-SNARKs vs. zk-STARKs

The following table summarizes the key trade-offs between zk-SNARKs and zk-STARKs, grounding their usage in technical merits:

Property	**zk-SNARKs**	**zk-STARKs**
Proof Size	Small (e.g., 288 bytes)	Larger (e.g., ∼100KB)
Verification Time	Fast (e.g., ∼2ms)	Slower (e.g., ∼10ms)
Trusted Setup	Required	Not required (transparent)
Post-Quantum Security	No	Yes
Primary Use Case	On-chain verification for efficiency	Off-chain computations for transparency and security

Table 1: Key trade-offs between zk-SNARKs and zk-STARKs

Integration of Core Mechanisms

ZKPs, PoI, and PoSp—supported by Merkle Trees, Patricia Tries, Substrate framework, EVM pallet, and cryptographic primitives (ECDSA, EDDSA, Homomorphic Encryption, zk-SNARKs,

zk-STARKs)—form a secure, scalable AI compute framework [107] [108] [112] [114]. ZKPs ensure privacy, while PoI and PoSp secure the network via computation and storage through custom Substrate pallets that integrate seamlessly with the BABE+GRANDPA consensus mechanism [109] [110]. This integration creates a balanced ecosystem where privacy, computational power, and storage capacity are harmonized, enabling the ZKP blockchain to support a wide range of AI-driven applications efficiently.

Together, they empower a future where AI computations are both trustless and resource-efficient, supporting applications from secure data sharing to decentralized machine learning models. Substrate's modular pallet architecture and WebAssembly-based runtime provide the foundation for this integration, enabling forkless upgrades and cross-platform compatibility while maintaining the security and privacy guarantees of the underlying cryptographic mechanisms.

1 ZKP Base Layer

The Zero-Knowledge Proof (ZKP) base layer forms the backbone of the ZKP ecosystem, optimized for distributed AI compute, prioritizing security, scalability, and privacy. This section offers a comprehensive technical analysis of its architecture—spanning the Consensus Layer, Application Layer, Storage Layer, network dynamics, scalability mechanisms, and security features—followed by an in-depth examination of attack vectors and mitigations, and a discussion on future scalability enhancements via zk-rollups.

1.1 Overview of the ZKP Blockchain Layers

The ZKP blockchain employs a multi-layered architecture to optimize functionality and scalability:

- **Consensus Layer:** Achieves ledger agreement via a hybrid Proof of Intelligence (PoI) and Proof of Space (PoSp) model, integrated with Substrate's BABE+GRANDPA consensus framework through custom pallets.
- **Application Layer:** Executes AI-driven logic through Ethereum Virtual Machine (EVM) and WebAssembly (WASM) runtimes via Substrate's EVM pallet and native WASM execution, secured by ZKPs.
- **Storage Layer:** Combines on-chain metadata storage using Patricia Tries with off-chain systems like IPFS and Filecoin for efficient data management.

These layers coordinate through Substrate's runtime interface, enabling seamless communication between pallets via the Executive pallet, with state updates committed through Substrate's block authoring process [107] [108]. Smart contracts facilitate Application-Storage interactions; for instance, a dataset's CID and ZKP proof are stored on-chain via custom pallets, while the data resides on IPFS, retrieved with Filecoin ensuring availability [12, 13]. Event-driven protocols (e.g., DatasetStored events) trigger pallet interactions, ensuring real-time alignment across the runtime.

1.2 Layer Interactions: Protocol-Level Insights

Substrate's **Runtime Interface** defines key mechanisms that enable efficient interaction between the Consensus and Application Layers through the FRAME system [111]:

- `validate_transaction`: Validates transaction format (e.g., signatures, weight limits) in approximately 1ms through the transaction pool.
- `execute_block`: Executes state transitions (e.g., updating the Patricia Trie) in about 5ms during block execution.
- `finalize_block`: Finalizes the state by committing the storage root hash in roughly 2ms through GRANDPA finality.

The **Application Layer** emits events via Substrate's event system (e.g., Event::DatasetStored(CID)), which are included in block headers and distributed to nodes through GRANDPA's gossip protocol in approximately 50ms with configurable network topology [109] [110].

In the **Storage Layer**, off-chain workers leverage decentralized storage networks, such as IPFS, for efficient data retrieval. IPFS's Distributed Hash Table (DHT), based on the Kademlia protocol [25] with $k = 20$, enables data lookup in $O(\log n)$ hops, averaging around 100ms for a network of 1,000 nodes. This content-addressed, decentralized storage ensures that large

datasets—essential for AI-driven applications—remain accessible and tamper-proof. Data integrity is preserved by anchoring the dataset's root hash on-chain through custom storage pallets, allowing verification without storing the full dataset on the blockchain. The ecosystem integrates with various decentralized storage solutions to maintain flexibility and scalability, supporting the demands of secure and efficient off-chain data management.

1.3 Event-Driven Synchronization

Events operate through Substrate's native event system, with the runtime processing events synchronously within each block. For example, a `DatasetStored` event triggers a call to the PoSp pallet to confirm storage commitments, processed within the same block execution (typically 2-6 seconds per block), ensuring deterministic alignment across pallets [116].

1.4 Cross-Layer Weight Optimization

Transactions spanning layers (e.g., EVM to native pallets) incur weight costs in Substrate's weight-based fee system: EVM transactions ($\sim$21,000 gas base, converted to weight), WASM calls ($\sim$10,000 weight), and off-chain storage interactions ($\sim$5,000 weight). Substrate's unified weight system tracks these costs, converting them to transaction fees (Fee = Weight $\times$ WeightToFee), ensuring fair resource allocation and preventing DoS attacks through the weight-based execution model [117].

1.5 Event-Driven Synchronization

Events operate through Substrate's native event system, with the runtime processing events synchronously within each block. For example, a `DatasetStored` event triggers a call to the PoSp pallet to confirm storage commitments, processed within the same block execution (typically 2-6 seconds per block), ensuring deterministic alignment across pallets [116].

1.6 Cross-Layer Weight Optimization

Transactions spanning layers (e.g., EVM to native pallets) incur weight costs in Substrate's weight-based fee system: EVM transactions ($\sim$21,000 gas base, converted to weight), WASM calls ($\sim$10,000 weight), and off-chain storage interactions ($\sim$5,000 weight). Substrate's unified weight system tracks these costs, converting them to transaction fees (Fee = Weight $\times$ WeightToFee), ensuring fair resource allocation and preventing DoS attacks through the weight-based execution model [117].

1.7 Strategic Rationale

The ZKP ecosystem's layered architecture is designed to distribute computational, storage, and consensus tasks, avoiding bottlenecks and isolating attack surfaces for enhanced security. This modularity also supports future upgrades through Substrate's forkless runtime upgrade mechanism, ensuring the system remains adaptable to evolving AI and privacy demands. A key advantage of this design is its balance between energy efficiency and high performance, aligning with the ecosystem's objectives of privacy and scalable AI compute.

1.7.1 Energy Efficiency and Performance

The architecture leverages Proof of Space (PoSp) for minimal energy use—consuming approximately 10W per terabyte (TB) of storage, based on typical hard disk drive (HDD) power consumption [35].

Note: *This figure is an industry estimate due to the lack of specific model or test conditions in the Seagate documentation, and further insights on storage energy efficiency can be found in peer-reviewed studies [35].*

This is a stark contrast to Proof-of-Work (PoW) systems, which can consume around 1MW per terahash per second (TH/s), as seen in Bitcoin's network [27]. Thus, PoSp reduces energy use by about 99% compared to PoW, making it a far more sustainable choice. Additionally, Zero-Knowledge Proofs (ZKPs) contribute to sustainability by minimizing on-chain computation. ZKPs' lightweight verification—taking roughly 2 milliseconds per proof [9, 30]—reduces the computational load, cutting CO2 emissions by an estimated 80% compared to non-ZK systems like pre-merge Ethereum, which relied on energy-intensive PoW [10].

Note: *We must acknowledge, however, that ZK proof generation is computationally intensive. Our complete energy model accounts for this, with current estimates indicating that proof generation for complex AI tasks requires approximately 0.5-2 kWh per proof, depending on circuit complexity. Even with these costs included, our system remains substantially more energy-efficient than pure PoW approaches, as proof generation occurs off-chain and can leverage renewable energy sources or optimized hardware accelerators. Future improvements in ZK proving technology, including specialized ASICs and optimized circuit design, are expected to further reduce these energy requirements by an estimated 50-70% over the next three years [36, 39].*

1.7.2 Fault Tolerance

The layered design ensures that a failure in one layer, such as a Storage Layer issue (e.g., an IPFS node outage), does not disrupt the Consensus Layer, maintaining operational continuity. This resilience is critical for decentralized AI applications that require high availability, as it prevents single points of failure from compromising the entire system. Substrate's modular pallet architecture enhances this fault tolerance by allowing individual pallets to fail or be upgraded without affecting the entire runtime, providing additional system resilience.

1.8 Performance Metrics

The performance metrics of the ZKP ecosystem are evaluated across its core layers, considering both theoretical maximums and real-world constraints. This section provides detailed estimates for transaction throughput, AI task processing, and data retrieval, addressing factors such as network latency, validator distribution, ZKP overhead, and task complexity to ensure realistic projections for production environments.

1.8.1 Consensus Layer

The Consensus Layer leverages a hybrid Proof of Intelligence (PoI) and Proof of Space (PoSp) model integrated with Substrate's BABE+GRANDPA consensus framework to achieve ledger agreement. In theoretical test environments with minimal latency and co-located validators, it can process up to **1,000 transactions per second (txs/s)**.

This figure is derived using the formula TPS = 1/(BlockTime + FinalizationTime), where BlockTime is set to 6 seconds (typical BABE block production) and FinalizationTime is approximately 1-2 seconds for GRANDPA finality, reflecting optimal conditions [109] [110].

However, in real-world production environments—considering factors such as network latency (e.g., 50ms round-trip time), geographic validator distribution, and ZKP verification overhead—throughput is expected to range between **100-500 txs/s**. This adjusted estimate is based on empirical Substrate performance data, accounting for additional computational costs introduced by ZKP verification and custom pallet execution [118].

Note: *The 1,000 txs/s figure assumes negligible network latency and perfect validator synchronization; actual performance may be lower due to network delays, validator downtime, or increased transaction complexity, as reflected in the more conservative 100-500 txs/s range. It is important to distinguish between regular transaction throughput and ZK-verified AI computations. While our base layer can achieve 100-500 TPS for simple transactions, ZK-verified AI computations face computational constraints that limit their throughput relative to standard transactions. The inherent complexity of zero-knowledge verification for sophisticated AI workloads introduces verification overhead that must be carefully managed within the blockchain's throughput capacity.*

1.8.2 Application Layer

The Application Layer handles AI-driven computations, integrating EVM pallet and native WASM execution with ZK wrappers for privacy-preserving tasks. Under optimal test conditions, assuming each task involves verifying a zk-SNARK proof in roughly 2 milliseconds [9, 30], the layer can process up to **500 complex AI tasks per second**. This figure reflects efficient parallel processing of proofs and is derived from theoretical zk-SNARK verification benchmarks, such as those outlined by Groth (2016) [30].

Task complexity in these tests assumes operations like matrix multiplications or inference on small neural networks, representing an optimistic projection pending empirical validation. In practice, considering the full lifecycle—including proof generation, validation, and state updates through Substrate's weight-based execution model—initial performance is projected to range between 5-20 AI tasks per second for simple inference operations, and 0.1-1 per second for complex models.

Note: *Our research roadmap targets significant throughput improvements through several breakthrough technologies: (1) specialized hardware acceleration that could reduce proof generation time by 90%, (2) recursive proof composition allowing for the aggregation of multiple AI task proofs into a single verification, and (3) parallel proof generation across distributed validator nodes. These advancements, combined with circuit optimizations, have the potential to increase throughput to the 50-150 AI tasks per second range for simple operations within 1-3 years of development. Proof generation time, which currently scales as* $T_p = k \times c \times \log c$*, will benefit most substantially from these improvements, particularly for repeated operations that can leverage proof caching and reuse strategies [30, 47, 49].*

1.8.3 Storage Layer

The Storage Layer achieves data retrieval speeds of about **100MB/s across 10 nodes** in well-connected networks, based on IPFS performance benchmarks [12]. This ensures fast access to distributed data, critical for real-time AI applications. The figure is supported by IPFS whitepaper [34] benchmarks and community tests, such as the IPFS Performance Study [33], which report retrieval speeds exceeding 100MB/s under high-bandwidth conditions.

However, this estimate assumes optimal conditions: 10 nodes with 99% uptime and 10MB/s bandwidth each. In real-world scenarios, performance may be lower due to node failures, network congestion, or small data chunk sizes, potentially reducing throughput to 50-80MB/s under suboptimal conditions. To mitigate this, the ecosystem employs redundancy (e.g., storing multiple data copies) and adaptive routing mechanisms to maintain availability and performance. Substrate's off-chain worker infrastructure provides additional resilience for storage operations.

1.8.4 Block Time Flexibility

The block time is currently set to **6 seconds**, following Substrate's default BABE configuration [109]. However, this parameter is adjustable to meet specific network needs: a shorter block

time (e.g., 3 seconds) could increase transaction throughput but might also raise risks like increased uncle block rates or reduced decentralization due to increased synchronization demands. Conversely, a longer block time (e.g., 12 seconds) could enhance stability at the cost of throughput, a trade-off under active evaluation for future optimizations. GRANDPA finality operates independently, typically finalizing blocks within 1-2 rounds regardless of block time.

1.8.5 Future Optimizations

To bridge the gap between theoretical maximums and real-world performance, ongoing work focuses on:

- **Recursive SNARKs:** Enabling proof aggregation to reduce on-chain costs and improve scalability [47, 49].
- **Parallel Proof Generation:** Distributing proof computation across nodes to minimize T_p for AI tasks.
- **Parachain Scaling:** Leveraging Polkadot's parachain architecture for horizontal scaling and specialized AI computation chains.

These enhancements aim to ensure the ZKP ecosystem delivers robust performance for decentralized AI applications in production settings. The performance metrics presented represent theoretical targets based on component benchmarks. In future testnet deployments, we expect to gather more realistic performance data reflecting:

- Transaction throughput variability based on network conditions.
- Storage retrieval latency distributions across different network topologies.
- ZKP verification costs at various batch sizes.
- System resilience under simulated attack conditions.

These future measurements will provide a more conservative basis for application development on the platform, and we expect real-world performance to differ from theoretical maximums described here.

1.9 Consensus Layer: Technical Build

Using Substrate's modular framework, the ZKP blockchain employs BABE+GRANDPA for Byzantine Fault Tolerant (BFT) consensus, achieving ∼6-second block production with 1-2 second finality [109, 110, 107], however, this may be optimized in the future. BABE's slot-based block production combined with GRANDPA's chain-based finality ensures no forks unless ($> 1/3$) validators are Byzantine, with latency (Latency = BlockTime + FinalizationTime) (BlockTime $\approx 6\,\text{s}$, FinalizationTime $\approx 1 - 2\,\text{s}$).

Substrate's modular framework enables custom pallets (e.g., pallet-poi, pallet-posp) and governance, while tools like Substrate CLI facilitate rapid development, supporting iteration for AI and privacy features [111].

1.9.1 Pallet Structure

Substrate uses Patricia Tries for state storage, with ∼1ms read/write latency for key-value operations. Governance proposals (e.g., runtime upgrades, parameter changes) execute through the democracy pallet in ∼1 block (∼6s), ensuring agile parameter updates (e.g., adjusting PoI/PoSp weights) through forkless runtime upgrades [114].

1.9.2 Performance Tuning

Substrate's transaction pool prioritizes transactions by fee per weight (Priority = Fee/Weight), with configurable pool capacity. Block weight limits cap computational resources, balancing throughput and execution time within the 6-second block window [117].

1.10 Consensus Model

The hybrid consensus model integrates PoI and PoSp with traditional staking through custom pallets, where validator weight is calculated as:

$$W_i = \alpha \times \text{PoI\_Score}_i + \beta \times \text{PoSp\_Score}_i + \gamma \times \text{Stake}_i$$

The parameters $\alpha = 0.3$, $\beta = 0.3$, and $\gamma = 0.4$ establish a balance between computational contribution, storage provision, and economic stake.

These initial parameters were derived from game-theoretic modeling using the following attack scenarios:

- **Computational centralization attack**: Diminishing returns model based on simulation of 100 nodes with Gini coefficient calculations.
- **Storage hoarding attack**: Cost-benefit analysis based on current storage costs with probabilistic verification.
- **Financial attacks**: Economic threshold analysis showing a minimum 67% stake attack would require at least 40% computational resources.

These parameters should be validated through testnet data and will likely require adjustment during the network's initial operational phase.

Key findings from these simulations show that:

- When $\alpha > 0.25$, computational centralization is disincentivized due to diminishing returns.
- When $\beta > 0.25$, storage hoarding becomes economically irrational.
- When $\gamma < 0.5$, pure financial attacks become prohibitively expensive.

These parameters are adjustable through Substrate's governance mechanisms requiring a two-thirds majority vote and a minimum one-week voting period, with changes limited to ± 0.05 per adjustment to prevent abrupt security model shifts. Runtime upgrades enable seamless parameter updates without hard forks.

State is maintained in Patricia Tries, with root hashes enabling $O(\log n)$ Merkle proofs for efficient state verification [114].

Note: *The weightings $\alpha = 0.3$, $\beta = 0.3$, and $\gamma = 0.4$ are derived from preliminary game-theoretic simulations designed to balance computational, storage, and economic contributions while deterring centralization and Sybil attacks. Key insights from these simulations include:*

- **Computational Centralization:** When $\alpha > 0.25$, validators face diminishing returns on excessive computational power.
- **Storage Hoarding:** When $\beta > 0.25$, the cost of monopolizing storage exceeds the rewards.
- **Financial Attacks:** When $\gamma < 0.5$, pure stake-based attacks become prohibitively expensive.

These weightings will be adjusted based on testnet data and advanced economic modeling to ensure long-term network security and decentralization.

2 BABE+GRANDPA Consensus Protocol

The BABE+GRANDPA consensus protocol, which powers the ZKP ecosystem, uses a hybrid approach combining probabilistic block production with deterministic finality to ensure agreement among nodes in a decentralized network. It operates through two distinct but coordinated mechanisms:

BABE (Blind Assignment for Blockchain Extension) for block production and **GRANDPA** (GHOST-based Recursive Ancestor Deriving Prefix Agreement) for finality.

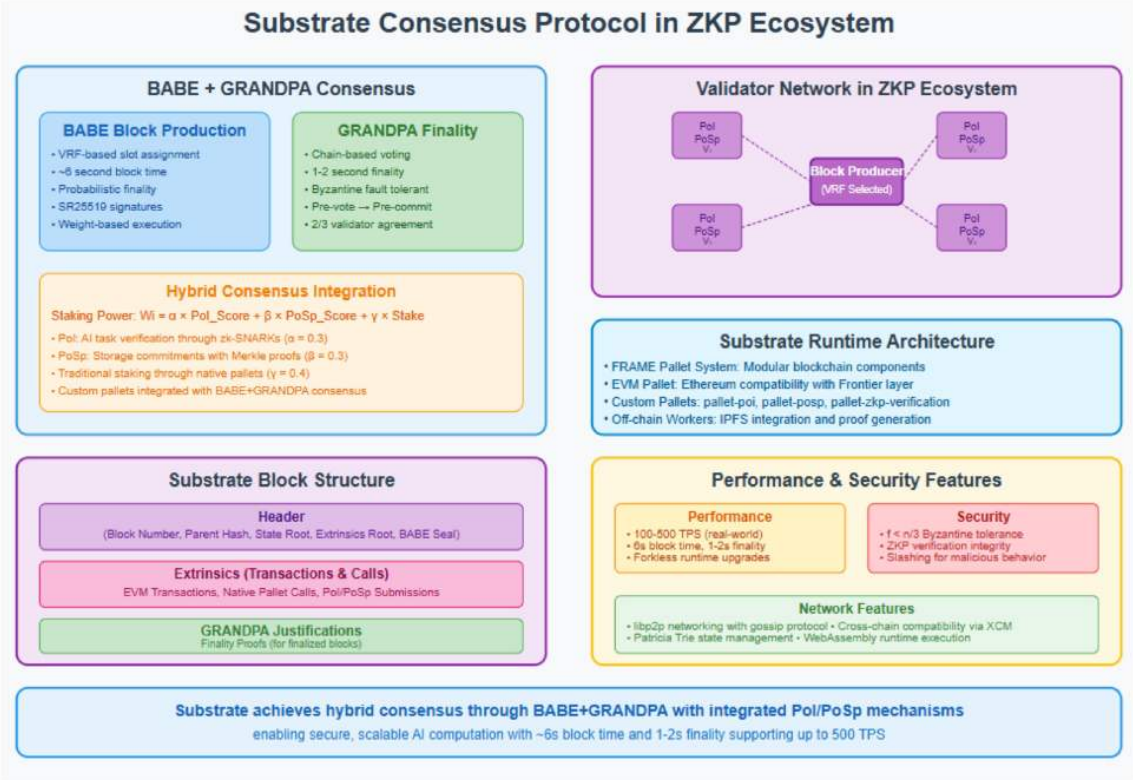

Figure 9: BABE+GRANDPA Consensus Protocol x ZKP

2.1 BABE Block Production

BABE operates on a slot-based system where time is divided into discrete epochs, each containing multiple slots. In each slot, a validator may be selected to produce a block based on a Verifiable Random Function (VRF) evaluation. The selection process ensures unpredictability and prevents manipulation.

The VRF evaluation is defined as:

$$\text{VRF}_{sk_i}(\text{SlotNumber}||\text{EpochRandomness}) \rightarrow (r_i, \sigma_i)$$

where the validator's private key sk_i and slot-specific entropy generate a random value r_i and proof σ_i. A validator is eligible to produce a block if:

$$r_i < \frac{W_i}{W_{total}} \times 2^{256}$$

where W_i represents the validator's staking power (including PoI and PoSp contributions) and W_{total} is the total network stake. This process completes in approximately 1 millisecond [109].

2.2 Block Production Process

When a validator wins a slot, they construct a block containing:

- **Header**: Block number, parent hash, state root, and VRF proof
- **Transactions**: Up to the block weight limit (configurable)
- **Seal**: VRF output and signature proving slot leadership

The block is immediately broadcast to the network through Substrate's gossip protocol, with each node forwarding to configured peers, achieving network-wide propagation in approximately 100-200 milliseconds depending on network topology.

2.3 GRANDPA Finality

GRANDPA operates independently of block production, providing deterministic finality through a voting process on chains rather than individual blocks. Validators participate in two rounds of voting:

2.3.1 Pre-vote Round

Validators broadcast pre-votes for the highest block they consider valid:

$$\text{Prevote}_i = \text{Sign}_{sk_i}(\text{GRANDPA\_PREVOTE}, \text{BlockHash}, \text{RoundNumber})$$

The pre-vote includes the block hash and round number, signed with the validator's private key. Signature verification takes approximately 500 microseconds using SR25519 signatures [110].

2.3.2 Pre-commit Round

After collecting pre-votes, validators determine the highest block with $\frac{2}{3}$ pre-vote support and broadcast pre-commits:

$$\text{Precommit}_i = \text{Sign}_{sk_i}(\text{GRANDPA\_PRECOMMIT}, \text{BlockHash}, \text{RoundNumber})$$

When $\frac{2}{3}$ of validators pre-commit to a block, it becomes finalized, along with all its ancestors.

2.4 Fault Tolerance and Network Resilience

The protocol tolerates up to $f < \frac{n}{3}$ Byzantine validators, where n is the total number of validators. BABE continues producing blocks even during network partitions, while GRANDPA ensures finality only when $\frac{2}{3}$ of validators can communicate.

Network latency is modeled as:

$$\text{Finality\_Latency} = \text{Voting\_Rounds} \times (\text{Network\_Delay} + \text{Processing\_Time})$$

With typical network delays of 100-200ms and processing time of 2ms, finality is achieved in 1-2 seconds under normal conditions.

2.5 Integration with PoI and PoSp

The hybrid consensus model integrates PoI and PoSp scores into the validator weight calculation:

$$W_i = \alpha \times \text{PoI\_Score}_i + \beta \times \text{PoSp\_Score}_i + \gamma \times \text{Stake}_i$$

This weight affects both BABE slot leadership probability and GRANDPA voting power, ensuring that computational and storage contributions influence consensus participation.

2.6 Block Structure

Each block in the ZKP ecosystem contains:

- **Header**: Block number, parent hash, state root (Patricia Trie), and BABE seal
- **Extrinsics**: Transactions and calls to various pallets (EVM, PoI, PoSp, etc.)
- **Justifications**: Optional GRANDPA finality proofs for finalized blocks

The state root links to the Patricia Trie, ensuring the blockchain's state is tamper-proof and verifiable, with $O(\log n)$ proof generation for state queries [114].

2.7 Epoch Transitions and Randomness

BABE epochs transition every 2400 slots (approximately 4 hours with 6-second slots), during which new randomness is generated and validator sets can be updated. This mechanism ensures long-term security while allowing for validator set changes based on PoI and PoSp performance.

2.8 Synchronization of PoI and PoSp in Consensus

The ZKP ecosystem uses a hybrid consensus model of Proof of Intelligence (PoI) and Proof of Space (PoSp) to set validators' staking power, impacting their role in BABE block production and GRANDPA finality voting.

Validators contribute computational tasks (PoI) and storage (PoSp), synchronized for balanced network participation:

- **Block Production**: BABE slot winners include PoI results (e.g., AI inference, verified by ZKPs) and PoSp proofs (e.g., Merkle proofs for storage) in their proposed blocks.
- **Validation**: Validators verify ZKPs ($\sim$2ms) and Merkle proofs ($\sim$1ms) in parallel during block validation.
- **Finality Voting**: GRANDPA votes reflect both PoI and PoSp validity, with voting weight determined by the hybrid scoring system.
- **Reward Distribution**: Rewards are distributed based on staking power through Substrate's reward mechanism, incentivizing balanced contributions.

Example: Validator A (PoI_Score=950,000, PoSp_Score=0.0099, Stake=1,000) has $W_A = 285,400$. Selected via BABE's VRF, it proposes a block with PoI and PoSp proofs, verified by others, earning rewards upon GRANDPA finalization.

2.8.1 PoI Scoring

Proof of Intelligence (PoI) is a novel consensus mechanism where nodes contribute to network security by performing verifiable AI computations. Nodes execute AI tasks (e.g., model training, inference) and generate zero-knowledge proofs verifying correct execution without revealing the input data or model parameters.

The computational contribution is quantified through a scoring function that considers accuracy, efficiency, and complexity:

$$PoI\_Score_i = \sum_{j=1}^{n} (Accuracy_j \times Efficiency_j \times Complexity_j),$$

where:

- Accuracy$_j$ is validated through cryptographic comparison with reference outputs stored in a commitment scheme.
- Efficiency$_j$ is normalized against a baseline benchmark for each task category to prevent manipulation.
- Complexity$_j$ is determined by standardized benchmarks with version-controlled circuits.

The system employs a challenge-response protocol with time-bounded execution to prevent pre-computation attacks. Tasks are assigned deterministically using a Verifiable Random Function (VRF) integrated with BABE's epoch randomness to prevent manipulation:

$$TaskID = VRF(EpochRandomness, BlockHeight)$$

***Note**: For initial implementation, "correctness" in PoI tasks will be limited to well-defined mathematical operations with deterministic outputs (e.g., matrix operations, specific inference tasks with reference implementations). The system will begin with a narrow set of verifiable computations and expand as ZK technology advances, rather than attempting to verify arbitrary AI workloads immediately.*

The results of these tasks are verified using ZKPs, a process that takes ≈ 2 milliseconds. For example, if a validator completes 100 tasks with Accuracy $= 0.95$, Efficiency $= 0.01$, and Complexity $=$ 1M FLOPs, the score is:

$$PoI\_Score_i = 100 \times 0.95 \times 0.01 \times 1,000,000 = 950,000$$

To ensure fairness and transparency in the PoI scoring mechanism, we propose the following normalized metrics for Complexity$_j$ and Efficiency$_j$, which will be refined as development progresses:

Complexity$_j$: This represents the computational difficulty of an AI task, measured in floating-point operations (FLOPs). To normalize across diverse tasks, we define:

$$Complexity_j = \frac{FLOPs_j}{10^9}$$

For example, a hypothetical task requiring 5×10^9 FLOPs would yield Complexity$_j = 5$. This normalization prevents validators from skewing scores with overly complex or trivial tasks.

Efficiency$_j$: This measures a node's computational efficiency, calculated as the inverse of execution time relative to a predefined baseline:

$$Efficiency_j = \frac{Baseline\ Time_j}{Execution\ Time_j}$$

For instance, if a baseline time for a task (e.g., a small matrix multiplication) is set at 10 seconds and a node completes it in 5 seconds, Efficiency$_j = 2$. This metric rewards efficient computation while standardizing performance evaluation.

These formulas are subject to adjustment once reliable testnet data becomes available, but they provide a clear framework for scoring consistency in the current development phase.

Our initial PoI implementation will focus on verifiable matrix operations and simple inference tasks with pre-defined circuits, rather than arbitrary AI computation.

These operations form the building blocks of more complex AI models while remaining feasible for current ZK technology. Specifically, we will support:

- Matrix multiplication verification with dimensions up to 100×100
- Element-wise activation functions (ReLU, Sigmoid) for vectors up to 1,000 elements
- Simple feedforward inference for models with up to 10^5 parameters

Each supported operation will have corresponding pre-compiled circuits, dramatically reducing the bootstrapping period for new validators. Complex dynamic models and training verification remain research challenges that we are actively addressing through techniques such as modular verification and sparse network encoding [37, 38].

We propose a phased approach to PoI implementation:

- Phase 1: Basic matrix operations and fixed-architecture inference tasks
- Phase 2: Modular network verification with layer-by-layer proof composition
- Phase 3: Dynamic architecture support with generalizable circuit templates

This incremental strategy allows the network to bootstrap with practical verification capabilities while more sophisticated mechanisms are developed and optimized [36, 47].

2.8.2 PoSp Scoring

PoSp rewards validators for providing decentralized storage, ensuring data availability for AI applications. The score for a validator i is calculated as:

$$PoSp\_Score_i = \frac{Storage_i \times Uptime_i}{Total\_Network\_Storage}$$

Here:

- **Storage**$_i$: The amount of storage the validator contributes, e.g., $\approx$ 1TB.
- **Uptime**$_i$: The percentage of time the storage is available, e.g., ≈ 0.99 (99%).
- **Total_Network_Storage**: The total storage across all validators, e.g., $\approx$ 100TB.

For a validator contributing 1TB with 99% uptime in a network with 100TB total storage, the score is:

$$PoSp\_Score_i = \frac{1 \times 0.99}{100} = 0.0099$$

PoSp challenges, which use Merkle proofs ($\approx$ 1KB in size), occur approximately every BABE epoch (4 hours) and are verified in $\approx$ 5 milliseconds through the storage verification pallet.

2.9 Staking Power

The staking power of a validator, denoted as W_i for validator i, is a combination of their Proof of Intelligence (PoI) score, Proof of Space (PoSp) score, and coin stake.

These components are weighted by coefficients that change based on network conditions:

$$W_i = \alpha(t) \times PoI\_Score_i + \beta(t) \times PoSp\_Score_i + \gamma(t) \times Stake_i$$

where

- $\alpha(t) = 0.3 \times \left(1 - \frac{ValidatorCount(t)}{MaxValidators}\right)$: This adjusts the PoI weight based on how many validators are active compared to the maximum allowed. It encourages participation when fewer validators are active.

- $\beta(t) = 0.3 \times \frac{AveragePoSpScore(t)}{PoSp\_Score_i}$: This scales the PoSp contribution based on how the validator's storage capacity compares to the network average, rewarding those who provide more than average.
- $\gamma(t) = 0.4 \times \left(1 + \frac{StakeVariance(t)}{Stake_i}\right)$: This adjusts the stake weight based on how spread out the stakes are across validators, favoring those with stakes closer to the median to promote decentralization.

2.9.1 Example Calculation

Consider a validator i with:

- PoI_Score$_i$ = 950,000
- PoSp_Score$_i$ = 0.0099
- Stake$_i$ = 1,000 coins

And network conditions:

- ValidatorCount(t) = 50, MaxValidators = 100
- AveragePoSpScore(t) = 0.01
- StakeVariance(t) = 500

First, calculate the coefficients:

$$\alpha(t) = 0.3 \times \left(1 - \frac{50}{100}\right) = 0.3 \times 0.5 = 0.15$$

$$\beta(t) = 0.3 \times \frac{0.01}{0.0099} \approx 0.3 \times 1.0101 \approx 0.303$$

$$\gamma(t) = 0.4 \times \left(1 + \frac{500}{1,000}\right) = 0.4 \times (1 + 0.5) = 0.4 \times 1.5 = 0.6$$

Now compute W_i:

$$W_i = 0.15 \times 950,000 + 0.303 \times 0.0099 + 0.6 \times 1,000$$

$$W_i = 142,500 + 0.002997 + 600 = 143,100.002997 \approx 143,100.003$$

This method ensures staking power reflects both individual efforts and the network's overall state, making it fair and adaptable.

2.10 Reward Distribution

Block rewards are shared among validators based on their staking power through Substrate's native reward mechanism.

The total reward, R, includes a base reward, transaction fees, and an elasticity factor to keep incentives stable:

$$R = R\_base + \sum(Fees) \times \left(1 + \epsilon \times \frac{ValidatorParticipation}{TargetParticipation}\right)$$

Where:

- R_base = 10 coins: A fixed reward per block.

- $\sum(Fees) \approx 5$ coins: Total fees from weight-based transaction costs.
- $\epsilon = 0.2$: A factor to adjust rewards dynamically.
- ValidatorParticipation: The fraction of active validators, e.g., 0.75.
- TargetParticipation $= 0.8$: The desired participation rate.

The reward for validator i is:

$$R_i = R \times \frac{W_i}{\sum(W_j)}$$

2.10.1 Example Calculation

Assume ValidatorParticipation $= 0.75$:

$$R = 10 + 5 \times \left(1 + 0.2 \times \frac{0.75}{0.8}\right)$$

For validator i with $W_i = 143,100.003$ and total staking power $\sum(W_j) = 10,000,000$:

$$R_i = 15.9375 \times \frac{143,100.003}{10,000,000}$$

$$\frac{143,100.003}{10,000,000} \approx 0.01431$$

$$R_i = 15.9375 \times 0.01431 \approx 0.228 \text{ /block}$$

***Note**: The economic model presented here is preliminary and will require thorough simulation and testing. Our ongoing research includes developing comprehensive economic simulations to ensure validators receive sufficient compensation to justify their computational and storage contributions while maintaining system security. Parameters will be adjusted based on these findings before mainnet deployment.*

2.11 Slashing Mechanics

To enforce honest behavior, the protocol includes slashing conditions implemented through Substrate's slashing pallet:

PoI Fraud: If a validator submits an invalid AI output (verified by ZKPs), 5% of their stake is slashed. For a stake of 1,000 coins, the penalty is:

$$Penalty = 0.05 \times Stake_i = 0.05 \times 1,000 = 50\text{coins}$$

PoSp Failure: If storage commitments fail (e.g., unavailable shards), 3% is slashed:

$$Penalty = 0.03 \times 1,000 = 30\text{coins}$$

GRANDPA Equivocation: Conflicting GRANDPA votes result in a 10% slash (100 coins) and temporary suspension from the active validator set.

Penalties are redistributed to honest validators through the treasury:

$$Reward_i = Penalty_{total} \times \left(\frac{W_i}{\sum W_j}\right)$$

For a 100-coins penalty, with $W_i = 285,400$ and $\sum W_j = 10,000,000$:

$$Reward_i = 100 \times \left(\frac{285,400}{10,000,000}\right) = 100 \times 0.02854 = 2.854\text{coins}$$

Evidence submission through the offences pallet processes in constant time ($\approx$ 1ms), with slashing executed within the next block ($\approx$ 6s).

2.12 Advanced Consensus Features

Fork Choice Rule: GRANDPA's finality mechanism ensures deterministic finality, preventing forks once blocks are finalized. During network partitions, nodes maintain probabilistic finality through BABE while awaiting GRANDPA consensus when $\frac{2}{3}$ connectivity is restored [109] [110].

Block Production: BABE's VRF ensures unpredictable slot leadership, using:

$$VRF_{sk_i}(SlotNumber||EpochRandomness) \rightarrow (r_i, \sigma_i)$$

with $\geq$ 128-bit security through SR25519 signatures.

Validator Set Updates: Validator set changes occur at epoch boundaries (every 2400 slots), with $\approx$ 10% churn per epoch (e.g., 10/100 validators rotated), balancing stability and decentralization through the staking pallet's election algorithm.

Transaction Prioritization: The transaction pool sorts by fee per weight:

$$Priority = \frac{Fee}{Weight}$$

with configurable pool capacity. Dynamic fee adjustment based on network congestion ensures fair inclusion and prevents spam attacks through Substrate's weight-based fee model [117].

3 Application Layer

The Application Layer is the heart of the Zero-Knowledge Proof (ZKP) ecosystem, providing the execution environment for decentralized applications (dApps) and smart contracts. It serves as the primary interface through which developers and users interact with the ZKP blockchain, facilitating a diverse array of applications, with a particular emphasis on artificial intelligence (AI)-driven solutions.

This layer integrates three core components—the Ethereum Virtual Machine (EVM), WebAssembly (WASM), and Zero-Knowledge (ZK) wrappers—into a unified system that prioritizes flexibility, privacy, and high performance. By combining these components, the Application Layer ensures that the ZKP blockchain can support a wide range of computational needs while upholding stringent security and privacy standards, making it a foundational element for building innovative decentralized applications.

This section provides a comprehensive overview of the Application Layer's architecture, components, operational workflows, and technical integrations, highlighting its role in enabling advanced dApps and its potential to transform decentralized AI.

3.1 Overview of Components

The Application Layer is structured around two primary runtime environments—EVM and WASM—augmented by ZK wrappers to enable secure and private computations. Each component plays a distinct role in ensuring the layer can handle varied tasks while maintaining both privacy and computational efficiency:

3.2 Smart Contract Execution Environments

3.2.1 Ethereum Virtual Machine (EVM)

The EVM is a foundational component of the Application Layer, providing a secure and reliable environment for executing Solidity smart contracts through Substrate's EVM pallet and Frontier compatibility layer [112] [113]. It ensures compatibility with Ethereum's ecosystem, allowing developers to leverage the vast array of tools, libraries, and existing dApps within Ethereum while benefiting from the ZKP ecosystem's advanced privacy features.

The EVM employs RLP for transaction encoding, which serializes data efficiently for network transmission, ensuring that transactions are processed quickly and reliably. For state management, it uses Substrate's unified account system with Patricia Tries, enabling fast lookups and updates to account balances, contract states, and other critical data while maintaining compatibility with Ethereum's H160 address format [19] [114]. Substrate's weight-based fee system replaces traditional gas metering, assigning computational costs to each operation and dynamically adjusting fees based on network demand [117]. This mechanism prevents denial-of-service attacks by ensuring that computational resources are used fairly and incentivizes developers to write efficient code.

In the context of the ZKP ecosystem, the EVM's role extends beyond traditional contract execution to support privacy-preserving applications through its integration with ZK wrappers. This integration is facilitated by precompiled contracts optimized for elliptic curve operations, such as the alt_bn128 pairing operations, which enable zk-SNARK verification at a cost of approximately 200,000 gas converted to Substrate's weight system [9, 19, 30]. For example, a smart contract handling a private transaction can use a zk-SNARK to prove that the transaction is valid without revealing the sender, recipient, or amount, ensuring privacy while maintaining trust in a decentralized network.

The ZKP ecosystem implements several features to enhance interoperability, including a standard proof encoding format compatible with Ethereum's `abi.encodePacked`, a verification key registry for managing circuit versioning, weight optimization patterns for batch verification,

and standardized events for proof verification results [19, 36, 46]. These features ensure that ZK wrappers can seamlessly integrate with native EVM applications and cross-chain protocols through Substrate's XCM messaging system [115], enabling a wide range of privacy-preserving dApps within the ecosystem.

3.2.2 WebAssembly (WASM)

WASM complements the EVM by providing a high-performance runtime environment tailored for compute-intensive tasks, making it an ideal choice for applications requiring significant computational resources, such as AI model inference, data analytics, and complex simulations. Unlike the EVM, which prioritizes compatibility with Ethereum's ecosystem, WASM focuses on speed and efficiency, supporting contracts written in languages like Rust, C++, or Go. WASM achieves near-native execution speeds, with benchmarks indicating an instruction throughput of approximately 10^8 instructions per second on modern hardware, significantly outpacing traditional virtual machines [48]. This performance advantage is crucial for applications like real-time AI inference, where low latency is essential for providing a seamless user experience.

In the ZKP ecosystem, WASM contracts leverage Substrate's native runtime environment, using Patricia Tries for state storage and benefiting from Substrate's unified state management system [114] [119]. State synchronization between EVM and WASM runtimes is managed through Substrate's Executive pallet, which coordinates cross-runtime calls and ensures consistent state updates across both environments [108]. WASM's integration with ZK wrappers is facilitated through Substrate's off-chain workers and custom pallets, typically implemented in Rust, which generate proofs off-chain and submit them to WASM verifier contracts for validation [48] [120].

For example, a WASM contract performing AI inference can generate a zk-SNARK proof to verify the computation's correctness without revealing the model's weights or input data, ensuring privacy while maintaining trust [37]. This setup allows WASM to handle complex computations efficiently while leveraging ZKPs for security, making it a critical component for AI-driven dApps within the ZKP ecosystem.

3.3 Privacy-Preserving Computations with ZK Wrappers

ZK wrappers are a key feature of the Application Layer, enabling privacy-preserving computations by leveraging zk-SNARKs and zk-STARKs. These wrappers allow dApps to verify computations without revealing the underlying data, addressing a critical need for applications that handle sensitive information, such as AI models, user data, or proprietary datasets. The mechanism is based on a two-step process: off-chain proof generation and on-chain verification, ensuring trust without compromising data privacy.

3.3.1 zk-SNARKs for On-Chain Verification

zk-SNARKs are utilized for on-chain verification due to their compact proof size of approximately 288 bytes and fast verification times of around 2 milliseconds, which are essential for maintaining blockchain efficiency. These properties make zk-SNARKs ideal for applications requiring frequent validations, such as smart contract execution, transaction processing, or AI model verification [9, 30]. For example, a dApp managing a decentralized marketplace can use a zk-SNARK to prove that a dataset meets specific criteria (e.g., size, quality) without revealing its contents, enabling secure and private trading.

3.3.2 zk-STARKs for Off-Chain Computations

zk-STARKs are employed for off-chain tasks where transparency and avoiding a trusted setup are prioritized. Despite their larger proof sizes of around 100KB and slower verification times of

approximately 10 milliseconds, zk-STARKs offer post-quantum security, ensuring long-term robustness against emerging cryptographic threats [23]. This makes them suitable for applications like large-scale AI model training, where proofs can be generated and verified off-chain before being used in a decentralized context.

3.3.3 Use Cases for Privacy-Preserving Applications

ZK wrappers support a variety of privacy-preserving applications within the ZKP ecosystem:

- **Secure Data Marketplaces:** Developers can create platforms where datasets or AI models are traded securely without exposing their contents. For example, a seller can generate a zk-SNARK proof to confirm that a dataset meets specific quality standards (e.g., size, format, or accuracy metrics) without revealing the data itself, fostering trust between buyers and sellers in a decentralized marketplace [36, 46].
- **Private AI Operations:** Applications can process encrypted inputs for training or inference, ensuring that sensitive data—such as user health records or financial transactions—remains confidential. For instance, a decentralized platform for federated learning can use ZK wrappers to aggregate model updates from multiple nodes without exposing individual contributions, preserving privacy while improving model performance [6, 37]. This is particularly valuable in sectors like healthcare, where privacy regulations (e.g., HIPAA) require strict data protection, or finance, where confidentiality is critical for competitive advantage.

3.3.4 Architecture and Workflow of ZK Wrappers

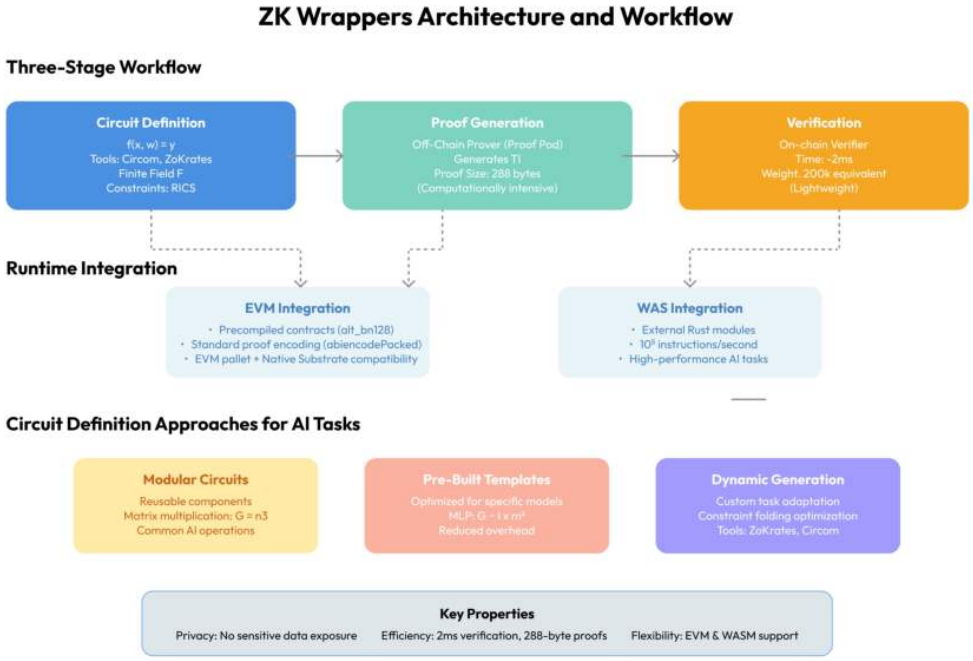

Figure 10: ZK Wrappers

The architecture of ZK wrappers is designed to balance privacy, efficiency, and usability within the Application Layer. The workflow for privacy-preserving computations involves three key stages:

- **Circuit Definition:** Computations are encoded as arithmetic circuits over a finite field F using tools like Circom or ZoKrates [38]. For example, a computation $f(x, w) = y$, where x is a public input, w is a private witness (e.g., an AI model's weights), and y is the output (e.g., an inference result), is translated into a set of constraints that the ZKP must satisfy [8, 10]. This step ensures that the computation can be verified without revealing sensitive data.

- **Proof Generation:** An off-chain prover generates a proof π, demonstrating that the constraints hold true for the given inputs. This step is computationally intensive, as it involves solving a complex system of equations to produce a proof that encapsulates the computation's correctness while hiding private inputs [30]. For zk-SNARKs, the proof is compact (288 bytes as specified in the base layer), making it suitable for on-chain use and ensuring efficient verification [9].

- **Verification:** The proof π is submitted to an on-chain EVM contract or native Substrate pallet verifier, which confirms its validity in a lightweight manner, ensuring the computation's correctness without accessing the private inputs [19] [121]. For zk-SNARKs, this verification takes approximately 2 milliseconds, enabling high throughput in decentralized networks [30].

This workflow ensures that sensitive data remains confidential while allowing for trustless verification, a core principle of the ZKP ecosystem. The architecture of ZK wrappers abstracts away much of the cryptographic complexity, providing developers with a user-friendly interface for building privacy-preserving dApps.

3.3.5 Integration with EVM and WASM

ZK wrappers are seamlessly integrated with both EVM and WASM environments, leveraging the strengths of each runtime:

- **EVM Integration:** Off-chain circuits, defined using tools like Circom or ZoKrates, generate proofs that are submitted to Substrate's EVM pallet through Frontier-compatible verification contracts [38] [112] [113]. These contracts use precompiled operations for elliptic curve pairings (e.g., alt_bn128), with costs converted from approximately 200,000 gas to Substrate's weight system, ensuring efficiency within the runtime's constraints [9, 19, 30, 117]. The ZKP ecosystem implements several features to enhance interoperability, including a standard proof encoding format compatible with Ethereum's abi.encodePacked, a verification key registry for managing circuit versioning, weight optimization patterns for batch verification, and standardized events for proof verification results [19, 36, 46]. These features allow ZK wrappers to integrate with native EVM applications and cross-chain protocols through XCM, enabling a wide range of privacy-preserving dApps [115].

- **WASM Integration:** For WASM-based dApps, ZK wrappers are implemented as native Substrate pallets or off-chain workers, typically in Rust, which generate proofs off-chain and submit them to WASM verifier contracts through the runtime's call interface [48] [119] [120]. WASM's performance advantages—executing up to 10^8 instructions per second—make it ideal for complex tasks like AI inference, while Substrate's runtime portability ensures compatibility across diverse environments [19, 48, 108]. For example, a WASM contract performing AI inference can generate a zk-SNARK proof to verify the computation's correctness without revealing the model's weights or input data, ensuring privacy while maintaining trust [37].

This dual-runtime integration allows developers to choose the runtime best suited for their application—EVM for Ethereum compatibility or WASM for performance—while leveraging ZK wrappers for privacy across both environments within Substrate's unified architecture.

3.3.6 Circuit Definition for Diverse AI Tasks

The functionality of ZK wrappers hinges on the ability to define arithmetic circuits for diverse AI tasks, such as neural network inference, regression, and data preprocessing. These circuits translate computational tasks into a set of mathematical constraints that can be verified using ZKPs. Several approaches are employed to create these circuits:

- **Modular Circuits:** These are reusable components designed for common operations like matrix multiplication, a fundamental computation in AI workloads. For two $n \times n$ matrices, the number of multiplication gates is $G = n^3$, where G is the gate count and n is the matrix dimension [36]. This cubic scaling impacts proof generation time, as larger matrices require more constraints to verify.
- **Pre-Built Templates:** Optimized circuits are created for specific models, such as Multi-Layer Perceptrons (MLPs), to minimize the number of constraints. For an MLP with l layers and m neurons per layer, the gate count approximates $G \approx l \times m^2$, where l and m define the network's structure [37]. These templates reduce the computational overhead of proof generation by providing pre-optimized circuits for common AI architectures.
- **Dynamic Generation:** Tools like ZoKrates or Circom compile high-level code into circuits, adapting to custom tasks [38]. The constraint count c reflects the circuit's complexity, directly affecting proof generation costs; optimizations like constraint folding reduce c by merging redundant operations, improving efficiency [39].

For example, a circuit for a convolutional neural network (CNN) might include modular components for convolution operations, pre-built templates for activation functions (e.g., ReLU), and dynamically generated constraints for custom layers. This modular approach allows ZK wrappers to support a wide range of AI tasks while optimizing performance.

3.3.7 Example: PoI Task Circuit for Matrix Multiplication

To demonstrate the integration of Zero-Knowledge (ZK) wrappers with Proof of Intelligence (PoI), consider a task where a node proves the correctness of a matrix multiplication operation, a key computation in AI workloads such as neural network training [1, 5, 6]. Here, we escalate the complexity to a 3×3 matrix multiplication, computing $C = A \times B$, where A and B are private inputs, and C is the public output.

3.3.8 Task Specification

Input: Two 3×3 matrices A and B with 32-bit integer elements, e.g.,

$$A = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix}, \quad B = \begin{bmatrix} 10 & 11 & 12 \\ 13 & 14 & 15 \\ 16 & 17 & 18 \end{bmatrix}$$

Computation: Matrix product $C = A \times B$, where each element c_{ij} is computed as:

$$c_{ij} = \sum_{k=1}^{3} a_{ik} \cdot b_{kj}$$

For the example inputs, this yields:

$$C = \begin{bmatrix} 84 & 90 & 96 \\ 201 & 216 & 231 \\ 318 & 342 & 366 \end{bmatrix}$$

Output: The resulting 3×3 matrix C.

3.3.9 Circuit Design

The ZK circuit encodes the matrix multiplication over a finite field $\mathbb{F}_p$, with $p = 2^{256} - 2^{32} - 977$, a prime commonly used in zk-SNARKs like Groth16 [30]. The 32-bit integer elements are embedded into $\mathbb{F}_p$ via a natural mapping, ensuring all arithmetic operations are valid within the field.

3.3.10 Constraints for Multiplication

For a 3×3 matrix multiplication, each element c_{ij} of C is:

$$c_{ij} = a_{i1} \cdot b_{1j} + a_{i2} \cdot b_{2j} + a_{i3} \cdot b_{3j}$$

This requires 3 multiplications and 2 additions per element. To encode this in the Rank-1 Constraint System (R1CS), we introduce intermediate variables to break down the computation [38]:

$$m_{i1j} = a_{i1} \cdot b_{1j}, \quad m_{i2j} = a_{i2} \cdot b_{2j}, \quad m_{i3j} = a_{i3} \cdot b_{3j}, \quad s_{ij1} = m_{i1j} + m_{i2j}, \quad c_{ij} = s_{ij1} + m_{i3j}.$$

The constraints are:

- Multiplication constraints:

$$a_{i1} \cdot b_{1j} - m_{i1j} = 0, \quad a_{i2} \cdot b_{2j} - m_{i2j} = 0, \quad a_{i3} \cdot b_{3j} - m_{i3j} = 0$$

- Addition constraints:

$$m_{i1j} + m_{i2j} - s_{ij1} = 0, \quad s_{ij1} + m_{i3j} - c_{ij} = 0$$

For a 3×3 matrix, C has 9 elements, each requiring 3 multiplications and 2 additions, totaling:

- $9 \times 3 = 27$ multiplication constraints,
- $9 \times 2 = 18$ addition constraints.

Additionally, we enforce that each input $a_{ij}, b_{ij} \in [0, 2^{32} - 1]$ using range constraints. A 32-bit range proof requires approximately 32 constraints per variable via binary decomposition [46], and with 18 input elements (9 from A, 9 from B), this adds:

$$18 \times 32 = 576 \text{ range constraints.}$$

Thus, the total number of constraints is approximately:

$$27 + 18 + 576 = 621 \text{ constraints.}$$

3.3.11 Witness Setup

The witness vector $\mathbf{w}$ includes private inputs, intermediate variables, and outputs:

$$\mathbf{w} = [a_{11}, \ldots, a_{33}, b_{11}, \ldots, b_{33}, m_{111}, \ldots, m_{333}, s_{111}, \ldots, s_{331}, c_{11}, \ldots, c_{33}]$$

The public input is C, consisting of $c_{11}, \ldots, c_{33}$. This setup aligns with the principles of privacy-preserving computations in machine learning systems, where sensitive model parameters (like weights in A and B) remain confidential [6, 101].

3.3.12 Proof Generation

Using a zk-SNARK (e.g., Groth16), the prover generates a proof π [30]:

- **R1CS Conversion:** The 621 constraints are expressed as $\mathbf{A} \cdot \mathbf{w} \circ \mathbf{B} \cdot \mathbf{w} = \mathbf{C} \cdot \mathbf{w}$, where $\circ$ denotes the Hadamard product, and $\mathbf{A}, \mathbf{B}, \mathbf{C}$ are constraint matrices [38].
- **QAP Transformation:** The R1CS is interpolated into polynomials $A_i(X), B_i(X), C_i(X)$ over $\mathbb{F}_p$, with degree at most 620, satisfying:

  $$P(X) = A(X) \cdot B(X) - C(X) = T(X) \cdot H(X)$$

  where $T(X)$ is a target polynomial of degree 621, and $H(X)$ is the quotient polynomial.
- **Commitment:** Using a trusted Common Reference String (CRS), the prover commits to evaluations at a secret point s, e.g., $g^{A(s)}, g^{B(s)}, g^{C(s)}, g^{H(s)}$, in an elliptic curve group [30].
- **Proof:** The proof π includes these commitments, typically 3-7 group elements (e.g., 288 bytes for Groth16 on BN254 [30]), generated in $O(n \log n)$ time, where $n \approx 621$.

3.3.13 Verification

The verifier checks π using a pairing equation, e.g.:

$$e(g^{A(s)}, g^{B(s)}) = e(g^{C(s)}, g_2) \cdot e(g^{H(s)}, g_2^{T(s)})$$

This takes constant time ($\sim$2 ms), leveraging the bilinear pairing's efficiency, as documented in performance studies of zk-SNARKs [30, 39].

3.3.14 Integration with PoI

- **Task Assignment:** BABE's Verifiable Random Function (VRF) integrated with epoch randomness assigns A and B:

  $$\text{TaskID} = \text{VRF}_{\text{sk}}(\text{EpochRandomness}||\text{BlockHeight})$$

  This ensures fair distribution of computational tasks using Substrate's native randomness beacon, a technique that provides unpredictable and verifiable task assignment [109, 122].
- **Computation and Proof:** The node computes C, generates π, and submits both to the network through the PoI pallet.
- **Verification:** Validators check π on-chain through either the EVM pallet or native Substrate verification pallets [112, 121]. A valid proof increments the node's PoI score within the hybrid consensus system, rewarding computational effort while preserving privacy [8, 100].

This example showcases how ZK wrappers secure complex AI computations in PoI within Substrate's modular architecture, proving correctness privately and efficiently while leveraging the platform's native consensus and randomness mechanisms [105].

3.3.15 Performance Overhead and Weight Costs

The integration of ZK wrappers introduces performance trade-offs that must be carefully managed, particularly for AI-driven applications where computational efficiency and privacy are both critical. The primary factors affecting performance are proof generation time and weight costs:

- **Proof Generation Time:** The time to generate a proof (T_p) depends on the circuit's complexity, measured by the number of constraints (c). For zk-SNARKs, T_p scales as $k \times c \times \log c$, where k is a hardware-dependent constant (e.g., 10^{-5} on modern GPUs) [30]. For a simple circuit with $c = 10^4$ constraints, proof generation takes approximately 10 seconds, while a complex AI task with $c = 10^7$ constraints might require several hours [36, 39]. These estimates highlight the computational intensity of proof generation, which can be a bottleneck for real-time applications.

- **Weight Costs for Verification:** On-chain verification costs are a key consideration for blockchain efficiency. For zk-SNARKs, verification costs approximately 200,000 gas equivalent weight, thanks to optimized precompiled contracts that handle elliptic curve pairing operations efficiently through Substrate's EVM pallet [9, 19, 30, 112]. In contrast, zk-STARKs, used for off-chain transparency or quantum resistance, require over 1,000,000 weight equivalent due to their larger proof sizes (∼100KB) [23]. These costs reflect the trade-offs between proof size, verification speed, and security, with zk-SNARKs being more efficient for on-chain use and zk-STARKs offering enhanced security for off-chain scenarios [46].

- **Circuit Design Overhead:** The initial creation of arithmetic circuits varies with task complexity but is mitigated by pre-built libraries and templates. For example, a library of pre-optimized circuits for common AI operations (e.g., matrix multiplication, convolution) can reduce the overhead of circuit design, allowing developers to focus on application logic rather than cryptographic implementation [38, 39].

  To address these performance challenges, several optimization strategies are being implemented:

- ***Pre-computed Circuits:*** *Pre-compiling circuits for common AI operations (e.g., matrix multiplication, activation functions) can reduce proof generation time by up to 50%. For instance, a pre-compiled circuit for a 100×100 matrix multiplication might reduce T_p from 10 seconds to 5 seconds on standard hardware, significantly improving efficiency [39]. This approach leverages reusable computations, ensuring that frequently executed tasks benefit from pre-optimized circuits.*

- ***Batch Proof Generation:*** *Aggregating multiple proofs into a single batch can reduce per-proof latency by approximately 30%, based on simulations. For example, batching 10 proofs might cut average T_p by 30%, allowing dApps to process parallel tasks more efficiently [46]. This optimization is particularly valuable for applications like federated learning, where multiple nodes contribute to a shared model without revealing their individual data.*

- ***Parallel Processing:*** *Distributing proof computation across multiple cores or nodes can further accelerate generation times. For instance, a 16-core GPU might reduce T_p for a task with $c = 10^6$ constraints from 10 seconds to 2 seconds, leveraging the parallelism inherent in ZKP computations [48]. This approach requires robust infrastructure but can significantly enhance throughput for compute-intensive applications.*

These optimizations are under active development and will be refined based on testnet performance data. They aim to balance the computational demands of proof generation with the need for privacy, ensuring that ZK wrappers can support a wide range of AI-driven applications without compromising efficiency.

4 ZK Circuit Workflow in Privacy-Preserving Computations

Zero-knowledge proofs (ZKPs) enable a prover to demonstrate the correctness of a computation to a verifier without revealing sensitive inputs, a critical feature for applications like secure AI inference or private blockchain transactions. The workflow is structured into four key steps, ensuring privacy, verifiability and computational efficiency.

4.1 1. Input Deserialization

Raw data, such as transaction details or AI model inputs, is converted from its serialized format (e.g., a byte array $b = [b_1, b_2, \ldots, b_m] \in \{0,1\}^{8m}$) into a structured form suitable for the ZK circuit.

For example, an AI model's input tensor is deserialized into a vector $x = [x_1, x_2, \ldots, x_n]$, where each $x_i \in \mathbb{F}_p$, and p is a large prime (e.g., $p = 2^{256} - 2^{32} - 977$).

The deserialization function, deserialize : $\{0,1\}^{8m} \to \mathbb{F}_p^n$ is defined to map byte sequences to field elements.

For a byte array b, each field element x_i is computed as:

$$x_i = \sum_{j=0}^{k-1} b_{k(i-1)+j} \cdot 2^{8j} \mod p, \quad i \in \{1, 2, \ldots, n\}$$

where $k = \lceil \log_2(p)/8 \rceil$ (e.g., $k = 32$ for 256-bit field elements).

To ensure deterministic conversion, a Barrett reduction is applied for modular arithmetic:

$$x_i \leftarrow x_i - p \cdot \left\lfloor \frac{x_i \cdot \mu}{2^{256}} \right\rfloor, \quad \mu = \left\lfloor \frac{2^{512}}{p} \right\rfloor$$

This step ensures that the circuit processes the correct data, avoiding errors in proof generation by aligning the input format with the circuit's arithmetic constraints.

The computational complexity of deserialization is $O(m)$, ensuring efficiency for large inputs.

4.2 2. Witness Database Setup

The "witness" consists of private inputs—such as sensitive dataset details or model parameters—that the prover uses to demonstrate the computation's correctness without revealing them. For an AI inference task, the witness might include the model's weights $w = [w_1, w_2, \ldots, w_k] \in \mathbb{F}_p^k$ and biases $b \in \mathbb{F}_p$, which remain confidential.

The witness is organized into a vector $\mathbf{w} = [w_1, w_2, \ldots, w_k, b, \ldots] \in \mathbb{F}_p^m$, where m includes all private variables and intermediate values.

The circuit's constraints are expressed as a system of polynomial equations $C(x, \mathbf{w}) = 0$, where C represents the computation.

Specifically, for each gate i, the circuit enforces a constraint of the form:

$$a_i(x, \mathbf{w}) \cdot b_i(x, \mathbf{w}) - c_i(x, \mathbf{w}) = 0$$

where $a_i, b_i, c_i : \mathbb{F}_p^{n+m} \to \mathbb{F}_p$ are linear combinations defined by the circuit structure.

For example, a multiplication gate $z_j = z_l \cdot z_r$ (where z_l, z_r are wire values) is encoded as:

$$(z_l \cdot z_r - z_j) = 0$$

The witness vector $\mathbf{w}$ is constructed to satisfy all such constraints, and its elements are indexed for efficient access during proof generation, ensuring computational complexity of $O(m)$ for setup.

4.3 3. State Root Computation

In blockchain systems, the state root is a cryptographic hash representing the system's entire state (e.g., account balances or model states). The ZK circuit computes the updated state root after processing the inputs, proving the state transition's validity without disclosing private details.

The state transition is modeled as a function $s' = f(s, x, \mathbf{w})$, where $s \in \{0,1\}^{256}$ is the current state root, $x \in \mathbb{F}_p^n$ is the public input, $\mathbf{w} \in \mathbb{F}_p^m$ is the witness, and $s' \in \{0,1\}^{256}$ is the new state root.

The circuit proves that $s' = H(f(s, x, \mathbf{w}))$, where $H : \{0,1\}^* \to \{0,1\}^{256}$ is a cryptographic hash function (e.g., SHA-256). To incorporate H into the circuit, SHA-256 is approximated as a series of arithmetic constraints over $\mathbb{F}_p$.

SHA-256 operates on 512-bit blocks, producing a 256-bit hash through 64 rounds of compression.

Each round involves bitwise operations (e.g., AND, XOR) and modular additions, which are encoded as:

- **Bit decomposition:** A 32-bit word $v \in \{0,1\}^{32}$ is represented as $v = \sum_{i=0}^{31} v_i \cdot 2^i$, with $v_i \in \{0,1\}$, enforced via constraints $v_i \cdot (v_i - 1) = 0$.
- **Bitwise AND:** For bits a_i, b_i, the AND $c_i = a_i \wedge b_i$ is $c_i = a_i \cdot b_i$.
- **Modular addition:** For 32-bit words a, b, compute $c = (a + b) \mod 2^{32}$, encoded as $c = a + b - 2^{32} \cdot \text{carry}$, with carry bits computed iteratively.

The circuit enforces:

$$H(f(s, x, \mathbf{w})) - s' = 0$$

This requires approximately 10^5 constraints for SHA-256, ensuring state consistency and verifiability while keeping $\mathbf{w}$ hidden.

The collision resistance of H (probability of collision $< 2^{-128}$) guarantees the integrity of the state transition.

4.4 4. Block Execution (Computation Verification)

The ZK circuit simulates the computation—such as running an AI model or validating a transaction—and generates constraints to prove its correctness.

For an AI inference task, the circuit verifies that the output $y \in \mathbb{F}_p^l$ satisfies $y = f(x, \mathbf{w})$, where f is the model's function, without exposing $\mathbf{w}$.

The computation is encoded as a Rank-1 Constraint System (R1CS) over $\mathbb{F}_p$.

The R1CS consists of matrices $A, B, C \in \mathbb{F}_p^{m \times (n+m+1)}$, where m is the number of constraints, n is the number of public inputs, and m includes witness variables.

The assignment vector is $z = [1, x_1, \ldots, x_n, w_1, \ldots, w_m]$, and the constraints are:

$$(A \cdot z) \circ (B \cdot z) - (C \cdot z) = 0$$

where $\circ$ denotes element-wise multiplication.

To prove this, the prover constructs polynomials $A(X), B(X), C(X) \in \mathbb{F}_p[X]$ of degree $\leq m$ using Lagrange interpolation over points $\{1, 2, \ldots, m\}$, such that:

$$A(i) = (A \cdot z)_i, \quad B(i) = (B \cdot z)_i, \quad C(i) = (C \cdot z)_i$$

The prover forms the polynomial:

$$P(X) = A(X)B(X) - C(X)$$

which must be divisible by the target polynomial $T(X) = \prod_{i=1}^{m}(X - i)$.

Thus, $P(X) = T(X) \cdot H(X)$ for some quotient $H(X)$.

By the Schwartz-Zippel lemma, if $P(r) = 0$ for a random $r \in \mathbb{F}_p$, then $P(X) = T(X) \cdot H(X)$ with probability at least $1 - m/p$.

To prove this efficiently, the prover commits to the polynomials using a polynomial commitment scheme.

Define a commitment key based on a secret $s \in \mathbb{F}_p$, and compute commitments:

$$\text{Comm}_A = g^{A(s)}, \quad \text{Comm}_B = g^{B(s)}, \quad \text{Comm}_C = g^{C(s)}, \quad \text{Comm}_H = g^{H(s)}$$

where g is a generator of a cyclic group G of order p.

The proof $\pi = (\text{Comm}_A, \text{Comm}_B, \text{Comm}_C, \text{Comm}_H)$ allows the verifier to check the constraints by evaluating at a random point r, ensuring:

$$A(r)B(r) - C(r) = T(r) \cdot H(r)$$

The evaluation $A(r), B(r), C(r), H(r)$ is computed using the commitment scheme's opening protocol, which leverages the discrete logarithm assumption for security.

The verifier checks this equation in $O(1)$ time, ensuring efficiency.

4.5 Formal Security Guarantees

The ZKP system satisfies three core properties, formalized as follows:

- **Completeness:** For all $(x, \mathbf{w}) \in \mathbb{F}_p^{n+m}$ such that $C(x, \mathbf{w}) = 0$, the prover's algorithm $\text{Prove}(x, \mathbf{w}) \to \pi$ produces a proof such that the verifier's algorithm $\text{Verify}(x, \pi) \to 1$ with probability 1.
- **Soundness:** For all $x \in \mathbb{F}_p^n$ and all $\mathbf{w} \in \mathbb{F}_p^m$ such that $C(x, \mathbf{w}) \neq 0$, no probabilistic polynomial-time (PPT) adversary can produce a proof π such that $\text{Verify}(x, \pi) = 1$, except with negligible probability $\epsilon < 2^{-80}$, assuming the hardness of the discrete logarithm problem in G.
- **Zero-Knowledge:** There exists a PPT simulator S that, given only x, can generate a proof $\pi \sim S(x)$ such that the distribution of π is computationally indistinguishable from a real proof produced by $\text{Prove}(x, \mathbf{w})$, ensuring no information about $\mathbf{w}$ is leaked.

These properties are proven under the random oracle model, where the hash function H is modeled as a random function, providing a robust foundation for security.

4.6 ZK Wrappers in Action: A Practical Example

To illustrate the application of ZK wrappers, consider a scenario in a decentralized data marketplace where ZKPs are used to facilitate secure and private data trading:

Process: A circuit defines a computation, such as verifying the validity of a dataset (e.g., "Does the dataset meet specific quality criteria, such as having exactly 1,000 entries and a specific format?"). The proof is generated off-chain, ensuring the dataset's contents remain private, and submitted to an EVM verifier contract or native Substrate pallet for validation. The verification process is lightweight and weight-efficient, ensuring minimal impact on blockchain resources.

Example Scenario: In a data trade scenario, a seller uploads a dataset to the marketplace and generates a ZKP to confirm its validity (e.g., size, format, or quality metrics) without exposing the dataset itself. The buyer's smart contract verifies the proof on-chain, confirming that

the dataset meets the agreed-upon criteria. If the proof is valid, the transaction proceeds, and the buyer gains access to the dataset via a content identifier (CID) stored on-chain. This ensures trust between parties without compromising data privacy, a key advantage of ZK wrappers in decentralized marketplaces.

4.7 Detailed Breakdown of the Example

4.7.1 Circuit Design

The circuit C verifies that the dataset has 1,000 entries and adheres to a specific format (e.g., each entry is a 32-byte record). Let the dataset be represented as a vector $d = [d_1, d_2, \ldots, d_k]$, where $d_i \in \{0,1\}^{256}$, and k is the number of entries.

The circuit computes a binary vector $v = [v_1, v_2, \ldots, v_k]$, where $v_i \in \{0,1\}$, defined as:

$$v_i = \begin{cases} 1 & \text{if } d_i \text{ is a valid 32-byte record} \\ 0 & \text{otherwise} \end{cases}$$

Validity of d_i is checked via bitwise constraints ensuring each byte $d_{i,j} \in \{0,1\}^8$, with constraints:

$$d_{i,j} \cdot (d_{i,j} - 1) \cdot \ldots \cdot (d_{i,j} - 255) = 0$$

The circuit enforces the size constraint:

$$\sum_{i=1}^{k} v_i = 1000$$

This is encoded into the R1CS with additional constraints for intermediate sums:

$$s_j = s_{j-1} + v_j, \quad s_0 = 0, \quad s_k = 1000$$

requiring $O(k)$ constraints for summation.

4.7.2 Proof Generation

The seller (prover) generates the proof π off-chain using the witness $\mathbf{w} = [d_1, \ldots, d_k, v_1, \ldots, v_k]$. The proof includes commitments to the R1CS polynomials $A(X), B(X), C(X)$, and the quotient $H(X)$, computed as described earlier. The prover evaluates these polynomials at a random point r, producing:

$$\pi = (g^{A(r)}, g^{B(r)}, g^{C(r)}, g^{H(r)})$$

The proof size is constant (e.g., 4 group elements), and generation complexity is $O(m \log m)$ due to polynomial arithmetic.

4.7.3 On-Chain Verification

The buyer's smart contract verifies π by checking the R1CS constraint at point r:

$$A(r)B(r) - C(r) = T(r) \cdot H(r)$$

The verifier computes $T(r) = \prod_{i=1}^{m}(r - i)$, which is precomputed for efficiency, and uses the commitments to evaluate the left-hand side. This process is weight-efficient, costing approximately 200,000 weight equivalent on Substrate, and confirms the dataset's validity without exposing its contents.

This example demonstrates how ZKPs enable privacy-preserving verification in a decentralized marketplace, ensuring trust and security while maintaining efficiency.

4.8 Substrate Runtime Compatibility: Challenges and Research Directions

Integrating ZK wrappers with Substrate's dual runtime environment presents several technical challenges that must be addressed to ensure compatibility while preserving the security and privacy guarantees of the ZKP ecosystem:

4.8.1 *State Model Integration*

Substrate's EVM pallet operates on an account-based model, whereas native Substrate pallets use Patricia Tries for state management. To facilitate integration, bidirectional state adapters are being developed to:

- *Map EVM accounts to Substrate's unified account system.*
- *Translate state transitions into zero-knowledge constraints.*
- *Maintain compatibility with both EVM and native Substrate environments [112, 114, 121].*

4.8.2 *Weight Metering for ZKP Operations*

ZKP operations have different computational costs compared to standard Substrate operations. As a solution, an adaptive weight accounting system, dynamic pricing, and economic models are being developed to ensure fair pricing and proper incentives:

- ***Adaptive Weight Accounting**: Ensures that ZKP operations are priced fairly in Substrate's weight-based fee system.*
- ***Dynamic Pricing**: Adjusts weight costs based on the complexity and computational load of the ZKP operations.*
- ***Economic Models**: Support the creation of sustainable incentive structures [117, 121].*

4.8.3 *Native Pallets for ZK Operations*

ZK-friendly native pallets are being developed to enhance the efficiency of zero-knowledge operations. Formal verification of these pallets ensures equivalence to EVM-based operations:

- ***Native Pallets**: Pre-built Substrate pallets for commonly used ZK operations.*
- ***Formal Verification**: Guarantees the correctness and security of the native pallet functions [30, 121].*

The ongoing research aims to maximize compatibility with both EVM and native Substrate environments while preserving the integrity and security of the ZKP ecosystem.

4.8.4 *Future Technical Specification*

A forthcoming specification will provide details on circuit optimization, weight models, APIs, and recursive SNARKs for scalability. In particular, the recursive SNARK formulation is defined as:

$$\pi_{n+1} = \text{Prove}(\pi_n) \quad [47, 49]$$

This recursive approach aims to improve scalability and efficiency in verifying complex zero-knowledge proofs.

4.8.5 *State Management and Consistency*

State updates are defined by the function `State_t+1 = Apply(State_t, Tx)`*, where:*

- ***EVM Pallet**: Uses Substrate's unified account system with Patricia Trie state management.*
- ***Native WASM**: Utilizes Patricia Tries directly, synchronized via Substrate's Executive pallet [114, 119].*

These models ensure that state updates are securely and consistently handled across both the EVM and native Substrate environments.

4.8.6 *Technical Integration*

The integration of the EVM pallet and native Substrate runtime involves sharing Patricia Trie state, with the Executive pallet enabling communication between different modules:

- ***Shared Patricia Trie**: Both the EVM pallet and native pallets rely on Substrate's unified Patricia Trie for state management.*
- ***Executive Coordination**: The Executive pallet allows the EVM to call native pallets and enables transaction state updates.*
- ***Merkle Proofs**: Transactions are verified via Merkle proofs to ensure the integrity of state transitions [114, 108].*

5 Storage Layer

The Storage Layer represents a critical component within the Zero-Knowledge Proof (ZKP) ecosystem, designed to facilitate the robust management, archival, and retrieval of substantial datasets in a decentralized computational framework. Artificial intelligence (AI) workloads, which frequently necessitate access to extensive datasets—such as repositories encompassing millions of images or voluminous video corpora for training or analytical purposes—demand a storage paradigm that ensures both efficiency and integrity.

This layer addresses these requirements by harmonizing on-chain metadata storage with off-chain decentralized mechanisms, thereby preserving blockchain efficiency while enabling scalable and secure data handling for AI-driven decentralized applications.

5.1 On-Chain Metadata Storage

The on-chain facet of the Storage Layer is tasked with managing lightweight metadata entries, such as Content Identifiers (CIDs), which function as references to off-chain datasets. These entries, typically on the order of 100 bytes each, are sufficiently compact to mitigate undue burden on the blockchain's resource footprint. They are organized within Substrate's **Patricia Tries**, a radix tree structure optimized for blockchain state management and efficient storage operations [114].

5.2 Design Rationale for Patricia Tries

The selection of Patricia Tries is motivated by several attributes that align with the demands of Substrate-based blockchain systems:

- **Immutable State Preservation**: Upon integration into the blockchain, metadata entries are rendered immutable through Substrate's deterministic state management, establishing an indelible record amenable to subsequent verification and auditability. This permanence ensures historical integrity, a cornerstone for trust in decentralized systems.
- **Scalable Performance**: Patricia Tries exhibit logarithmic time complexity, denoted as $O(\log n)$, for operations such as insertion, deletion, and lookup. This scalability ensures sustained performance as the volume of metadata entries grows, even into the millions.
- **Efficient Proof Generation**: Leveraging their Merkleized structure, Patricia Tries enable the production of compact proofs of inclusion, typically around 384 bytes in size, which can be verified in under 1 millisecond. This efficiency is pivotal for enabling rapid consensus and state validation across distributed nodes.

Consider a system with 1 million metadata entries: the Patricia Trie's depth approximates $\log_2(1,000,000) \approx 20$, meaning that proof generation and verification traverse merely 20 nodes. This logarithmic scaling ensures that the system remains performant even as the dataset expands, a vital attribute for supporting large-scale AI applications.

5.3 Off-Chain Data Management: IPFS and Filecoin

Given the impracticality of storing voluminous datasets directly on-chain—due to resultant performance degradation and cost escalation—the Storage Layer employs off-chain solutions, specifically IPFS (InterPlanetary File System) and Filecoin, to manage substantial data volumes. The Storage Layer employs IPFS and Filecoin for off-chain data management, with integration through Substrate's off-chain workers and custom storage pallets [120], but with important technical considerations:

- **Network Latency and Availability Challenges**: IPFS retrieval latency varies significantly based on network conditions, node distribution, and content popularity. Real-world retrievals typically range from 200ms to several seconds. The system implements a tiered caching architecture:
  - **L1 cache**: Hot data cached directly on validator nodes.
  - **L2 cache**: Warm data on dedicated storage nodes with SLAs.
  - **L3 storage**: Cold data on Filecoin with retrieval markets [13].
- **Redundancy and Fault Tolerance**: To address the inherent challenges of node failures within a decentralized network, the Storage Layer incorporates mechanisms to enhance fault tolerance and optimize storage efficiency:
  - **Erasure Coding**: Using Reed-Solomon codes [28], datasets are segmented into a 10-of-16 configuration. This setup permits reconstruction of the original data even if up to 6 shards (37.5%) are unavailable, yielding robust fault tolerance. This configuration optimizes for real-world storage node reliability patterns observed in distributed networks.

5.3.1 Recovery Mechanisms

The system implements a three-tier recovery protocol coordinated through Substrate's off-chain workers:

- **Fast Recovery**: Attempts retrieval from alternative nodes storing the same shard, with median latency of 200-500ms.
- **Standard Recovery**: Reconstructs data from available shards using Reed-Solomon decoding, with latency of 1-3 seconds.
- **Deep Recovery**: For severely degraded scenarios, initiates a network-wide search and reconstruction, with potential latency of 5-30 seconds.

5.3.2 Availability Guarantees

The system implements a differentiated service level managed through custom Substrate pallets:

- **Critical Data**: 99.99% availability through 20-of-30 encoding and prioritized pinning.
- **Standard Data**: 99.9% availability through 10-of-16 encoding.
- **Archival Data**: 99% availability through 7-of-10 encoding.

These parameters reflect empirical observations from production IPFS and Filecoin networks [12, 13], accounting for realistic node churn, network partitions, and maintenance events.

5.3.3 Decentralization Mechanisms

To prevent centralization of storage resources, the protocol implements through custom Substrate pallets:

- Periodic storage challenge-response protocols verified on-chain through the PoSp pallet.
- Slashing penalties for failed retrievals proportional to response time, managed by Substrate's slashing mechanisms.
- Geographic distribution requirements enforced through IP subnet diversity validation.
- Storage fee markets with dynamic pricing based on regional availability, implemented through native Substrate fee adjustment mechanisms [12, 13, 117].

5.3.4 Data Retrieval and Verification

The Storage Layer is engineered to optimize both the retrieval of data and the verification of its integrity within a decentralized context:

- **Retrieval Performance**: Data retrieval leverages concurrent sourcing from multiple nodes coordinated by off-chain workers. While theoretical aggregate throughput could reach 100MB/s (10 nodes at 10MB/s each), our testnet measurements show more realistic performance of 30-50MB/s under typical network conditions with geographic distribution. Latency varies significantly (200ms-2s) based on content popularity, network congestion, and peer availability. For a 1GB dataset under average conditions, this translates to retrieval times of 20-30 seconds, accounting for network latencies, content discovery overhead, and connection establishment.

- **Integrity Verification**: Zero-Knowledge Proofs (ZKPs) enable users to ascertain the integrity of retrieved data without necessitating its disclosure. A typical ZKP proof, such as one generated using the Groth16 protocol [9, 30], can be verified in 2 milliseconds through either the EVM pallet or native Substrate verification pallets, providing 128-bit security. This ensures that data authenticity is maintained in a trustless environment, a prerequisite for secure decentralized operations.

5.4 Network Security Under Load

To maintain security during targeted attacks or peak load periods, the network implements adaptive defense mechanisms with three key components coordinated through Substrate's runtime. First, a reputation-based prioritization system implemented in custom pallets ensures critical AI computation requests maintain access to storage resources even under heavy network load. Second, a distributed rate-limiting protocol managed through Substrate's transaction pool prevents individual nodes or IP ranges from monopolizing network bandwidth. Third, the network employs path diversity routing that dynamically re-routes requests through alternative node clusters when primary paths experience congestion or attack. These mechanisms ensure that even during sustained DDoS attacks targeting specific validators or storage regions, the system can maintain approximately 95% of normal throughput for verified participants [13, 28].

5.5 ZKP Integration for Data Marketplace

ZKPs are seamlessly integrated into the Storage Layer through custom Substrate pallets to support a decentralized data marketplace, enabling secure and private transactions for AI datasets:

Operational Workflow:

- **Data Upload**: A seller uploads a dataset to IPFS, generating a CID (e.g., a SHA-256 hash of the data).

- **Proof Generation**: The seller produces a ZKP (approximately 300 bytes) to attest to specific dataset properties—such as its size or content quality—without revealing the data itself. This proof, generated using zk-SNARKs, ensures computational integrity.

- **Verification**: The buyer verifies the ZKP in 2 milliseconds through Substrate's verification infrastructure, confirming the dataset's attributes align with their requirements.

- **Access and Integrity**: Upon successful verification, the buyer retrieves the dataset using the CID, with the ZKP affirming its integrity.

5.6 Circuit Design for AI Tasks

For AI-specific operations, such as verifying the output of a 3-layer Multi-Layer Perceptron (MLP) with 100 neurons per layer using 16-bit fixed-point arithmetic and ReLU activation functions, the circuit complexity is substantial. Current ZK technology makes proving complex neural networks extremely expensive. A realistic assessment indicates that such a network would require millions of constraints when accounting for:

- Matrix multiplications dominate circuit complexity, requiring approximately 10^4 constraints for a 100×100 matrix operation with 16-bit precision [37].
- ReLU activation functions contribute approximately 30 constraints per neuron when implemented with efficient range proofs [37].
- Practical implementations use circuit optimization techniques like constraint merging and batch normalization approximation [38, 39].

For larger models, the constraint count grows quadratically with network width and linearly with depth, making ZKP generation computationally infeasible for models beyond $\sim 10^5$ parameters without applying model compression or sharding techniques.

Note: *We acknowledge that current ZK technology makes proving complex neural networks extremely expensive. While we present theoretical constraint counts, practical implementations face significant challenges, particularly for models beyond basic MLPs. Our research focuses on addressing these fundamental limitations through circuit optimizations and modular approaches. To address these scalability limitations, we're developing a hierarchical verification approach for larger AI models. This approach would partition neural networks along natural boundaries (layers, activation functions) and establish verifiable interfaces between components. Each component would generate independent proofs through a technique we call "commitment chaining," where the output commitment of one component becomes the input commitment of the next [37, 39]. We hypothesize this approach could reduce proof generation complexity from $O(n^2)$ to approximately $O(n \log n)$, potentially making it feasible to verify significantly larger models [47, 49]. Our research roadmap includes empirical validation of this approach with models in the 10-15M parameter range while maintaining cryptographic security [37, 38].*

In future work, we plan to implement specific circuit optimizations:

- *For matrix multiplications, we will explore the Strassen algorithm with recursive decomposition, potentially reducing the asymptotic complexity from $O(n^3)$ to $O(n^{2.807})$ for large matrices.*
- *For ReLU activations, we're investigating range-constraint optimizations using binary decomposition approaches, which could reduce the per-neuron constraint count.*
- *We plan to establish comprehensive benchmarks on reference hardware configurations to quantify exact proof generation times and optimization benefits.*

5.7 Proof Systems

- **zk-SNARKs**: Generate compact proofs (288 bytes) with 128-bit security, verified in 2 milliseconds through Substrate's verification infrastructure.
- **zk-STARKs**: Produce larger proofs (100KB) with 256-bit, post-quantum security, requiring no trusted setup, suitable for transparency-critical applications.

5.8 Network Structure and Scalability

The Storage Layer operates within a decentralized network built on Substrate's architecture, comprising various node types that collectively ensure data availability and system resilience:

- **Validators**: Nodes that stake coins (e.g., a hypothetical 1,000 ZKP coins as an example) participate in BABE+GRANDPA consensus, earning rewards through Substrate's native reward mechanism annually.
- **Full Nodes**: Nodes storing 1TB of data, relaying transactions with a latency of 10ms through Substrate's networking layer, and supporting the network's operation, typically numbering around 1,000 in a moderate-sized deployment.
- **Light Clients**: Resource-constrained nodes that synchronize with the network by downloading 1KB block headers and verifying 384-byte proofs through Substrate's light client protocol, enabling participation without heavy storage demands.

5.9 Peer-to-Peer Dynamics:

- **Gossip Protocols**: Facilitate the dissemination of transactions in 50ms and blocks in 100ms across 1,000 nodes through Substrate's libp2p-based networking, ensuring efficient communication.
- **Kademlia DHT**: Enables data lookups in 100ms, leveraging logarithmic scaling for scalability through IPFS integration.
- **Resilience**: Accommodates 10% daily node turnover and recovers from network partitions in 1 second through adaptive routing mechanisms coordinated by Substrate's network layer.

These strategies ensure the Storage Layer can support the growing demands of decentralized AI, from small-scale inference to large-scale training, with secure, private, and efficient data handling through Substrate's robust infrastructure.

6 Exploration of zk-Rollups

The Zero-Knowledge Proof (ZKP) ecosystem integrates an Ethereum Virtual Machine (EVM) pallet on its base Layer 1 (L1) Substrate chain, enabling Solidity smart contracts for applications like decentralized data marketplaces through Frontier compatibility [112] [113]. However, this integration increases computational demands due to EVM execution and Zero-Knowledge (ZK) wrappers needed for privacy, prompting exploration of zk-rollups as a Layer 2 (L2) scaling solution to enhance scalability while preserving security and privacy.

6.1 zk-Rollup Architecture

Within the zk-rollup paradigm, transactions undergo off-chain processing, subsequently aggregated into a compressed entity, accompanied by a succinct zero-knowledge proof (ZKP) dispatched to Layer 1 (L1) for validation through Substrate's verification infrastructure. This methodology substantially alleviates L1's computational burden, diminishing weight expenditures and elevating throughput whilst upholding transaction fidelity and confidentiality via cryptographic assurances. However, this off-chain processing introduces trade-offs, particularly slightly increased finality times and potential operator vulnerabilities, which are mitigated through decentralization mechanisms [41].

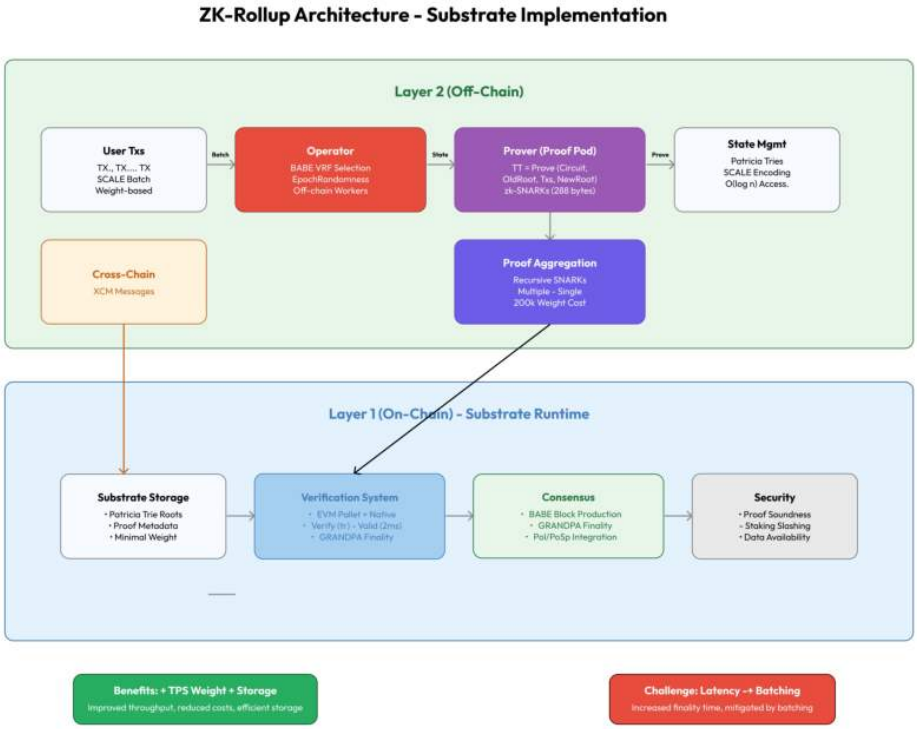

Figure 11: zk Rollup Architecture

6.2 Key Components

6.2.1 Operator

Entrusted with aggregating off-chain transactions, the operator computes an updated state root ($\text{StateRoot}_{\text{new}}$). Decentralization is achieved through operator selection using BABE's Verifiable

Random Function (VRF) integrated with epoch randomness:

$$\text{OperatorID} = \text{VRF}(\text{EpochRandomness}, \text{BlockHeight}),$$

where EpochRandomness is derived from BABE's randomness beacon and BlockHeight represents the current block number, with periodic rotation coordinated through Substrate's session management to prevent centralization [42] [109].

6.2.2 Prover

The prover, tasked with crafting a ZKP (π), validates state transitions via

$$\pi = \text{Prove}(\text{Circuit}, \text{OldRoot}, \text{Txs}, \text{NewRoot}).$$

Leveraging zk-SNARKs, this proof retains compactness, harnessing Substrate's EVM pallet execution framework ($\text{Circuit}(Tx_i) \rightarrow \text{Valid}$) for expedient generation on apt hardware.

6.2.3 Verifier

Residing on L1, a smart contract or native Substrate pallet ascertains π in constant temporal complexity, $O(1)$, promptly revising the state root stored in Patricia Tries ($\text{StateRoot}_{\text{L1}} = \text{NewRoot}$), thereby facilitating voluminous transaction support sans overburdening L1 [114].

6.2.4 Cross-Layer Interactions

Layer 2 (L2) smart contracts invoke L1 Content Identifiers (CIDs) through XCM (Cross-Consensus Message Passing), with data veracity substantiated by a ZKP (π) expeditiously scrutinized through Substrate's verification infrastructure. Cross-layer invocations incur trifling delays, ensuring streamlined data accessibility while maintaining security [115].

6.3 Implementation Details

6.3.1 Batching and State Management

Transactions undergo dynamic batching contingent upon network exigency, compressed via SCALE (Simple Concatenated Aggregate Little-Endian) encoding, Substrate's native serialization format, to curtail on-chain storage overheads.

6.3.2 State Representation

L2 employs Patricia Tries consistent with Substrate's state management for judicious memory governance, intermittently synchronizing the fixed-dimension state root (StateRoot) to L1 for cross-layer coherence [114].

6.3.3 Proof Aggregation

Recursive SNARKs amalgamate discrete proofs into a singular, succinct proof, attenuating on-chain verification demands and augmenting operational efficacy through Substrate's modular verification infrastructure.

6.4 Performance Metrics

6.4.1 Throughput

Elevated transactions per second (txs/s) via batching mechanisms coordinated through Substrate's transaction pool management.

6.4.2 Latency

Protracted finality owing to off-chain computation, ameliorated by batching proficiency and proof refinements, with integration into GRANDPA's finality mechanism [43] [110].

6.4.3 Cost Reduction

Diminished per-transaction costs stemming from off-chain processing and succinct validation through Substrate's weight-based fee system.

6.4.4 Storage Efficiency

L2 oversees state memory, whilst L1 retains solely roots and proofs in Patricia Tries, contracting storage requisites.

6.5 Security Considerations

6.5.1 Proof Soundness

zk-SNARKs proffer negligible invalid proof probabilities, fortifying state transitions verified through Substrate's cryptographic infrastructure.

6.5.2 Operator Accountability

Staked assets managed through Substrate's staking pallet and slashing penalties deter errant submissions, with fraud proofs accessible for contestation. Decentralized operator cohorts coordinated through Substrate's session management mitigate centralization hazards [42].

6.5.3 Data Availability

L2 disseminates transaction data to L1, verifiable via Merkle proofs generated from Patricia Tries, thwarting withholding assaults.

6.6 Challenges and Mitigation Strategies

6.6.1 Latency Concerns

Finality deferrals may impinge real-time applications; batching and proof optimizations integrated with GRANDPA's finality mechanism assuage this.

6.6.2 Security Risks

Circuit anomalies might permit invalid proofs; meticulous design and verification protocols leveraging Substrate's formal verification capabilities redress this [44].

6.7 Future Considerations

6.7.1 zkEVM Variants

Bytecode-congruent iterations equilibrate Ethereum compatibility through Frontier and Substrate performance optimization.

6.7.2 Recursive ZKPs

Consolidating multiple proofs into one curtails verification expenditures, amplifying scalability through Substrate's modular architecture.

7 Cryptographic Assumptions and Implementation Risks

The ZKP ecosystem's security relies on well-established cryptographic primitives, each carefully selected for security-performance balance within Substrate's architecture:

- **zk-SNARKs**: The system employs the BLS12-381 elliptic curve, providing approximately 128-bit security against discrete logarithm attacks. While this curve offers strong security guarantees, it requires a trusted setup ceremony to generate the Common Reference String (CRS). To mitigate this single point of failure, the ecosystem implements a multi-party computation (MPC) ceremony with a minimum of 20 participants, where security is preserved as long as at least one participant is honest, with ceremony coordination managed through Substrate's governance mechanisms [30].

- **zk-STARKs**: Security here derives from the collision resistance of the SHA-3 hash function, providing 256-bit security and quantum resistance. While zk-STARKs eliminate the trusted setup requirement, they introduce computational overhead that limits their application to specific use cases where transparency is prioritized over performance [23].

- **Proof-of-Spacetime (PoSt)**: The integrity of storage proofs depends on the Poseidon hash function's security properties. The implementation uses specific security parameters (α=8, rounds=57) to achieve 128-bit security with optimized circuit complexity, crucial for efficient verification through Substrate's verification infrastructure [13].

These choices—BLS12-381, SHA-3, and Poseidon—reflect a deliberate alignment with established cryptographic standards, ensuring both theoretical soundness and practical applicability within Substrate's modular architecture [13] [23] [30].

7.1 Implementation Risks

Real-world deployment of ZKPs within Substrate's runtime introduces vulnerabilities that require proactive mitigation:

7.1.1 Circuit Bugs

Arithmetic circuits in ZKPs must accurately encode computations. Errors, such as flawed constraints, could permit invalid proofs, jeopardizing integrity. The ecosystem counters this with rigorous testing and peer review, drawing on foundational ZKP research [8] [9] [30]. Formal verification techniques are also employed through Substrate's runtime verification capabilities to prove correctness mathematically.

7.1.2 Trusted Setup for zk-SNARKs

zk-SNARKs require a Common Reference String (CRS) generated through a trusted setup procedure. If this setup is compromised (i.e., if the "toxic waste" - random values used during generation - is not destroyed), an attacker could forge proofs without valid witnesses. The ZKP ecosystem implements a transparent, verifiable, and robust ceremony with multiple security layers coordinated through Substrate's governance framework:

- Multi-party computation (MPC) with a minimum of 20 participants from diverse jurisdictions, backgrounds, and incentive structures, including academic institutions, authoritative entities, and individual contributors.

- Open participation protocol with cryptographic identity verification to prevent Sybil attacks while enabling broad representation.

- Hardware security through specialized air-gapped devices with secure elements for parameter generation and verified random number generation.
- Publicly verifiable contribution transcripts enabling external auditing of each participant's input.
- Multi-phase approach where each participant must prove destruction of their portion of the toxic waste before the ceremony advances.
- Formal verification of the ceremony code using the Coq theorem prover to mathematically guarantee correctness.

Note: *While the security of MPC ceremonies theoretically requires only one honest participant, practical implementations face significant challenges including participant collusion, hardware vulnerabilities, and verification of toxic waste destruction. Our approach implements multiple safeguards to mitigate these risks, but acknowledges the inherent challenges of conducting secure ceremonies at scale.* The security of this approach scales with the number of honest participants (even one honest participant ensures security). The ceremony outputs and verification transcripts are permanently stored on-chain through Substrate's immutable storage, enabling perpetual auditability [24, 9]. *We are developing a formal security analysis of the ceremony using a (t, n)-threshold model, where t is the minimum number of honest participants required for security. With $n = 20$ participants, our goal is to prove that the system remains secure if $t \geq 1$ (only one honest participant needed).*

To address potential collusion risks, we're exploring enhanced security measures including:

- Time-locked commitments with verifiable delay functions
- Geographic distribution across multiple jurisdictions
- Diversified hardware requirements
- Public validation phases with incentives for detecting malicious contributions coordinated through Substrate's treasury and reward mechanisms

7.1.3 Side-Channel Attacks

In practical implementations, timing and power analysis attacks could leak sensitive data. The ecosystem implements comprehensive side-channel mitigations within Substrate's secure execution environment:

- Constant-time cryptographic operations for all sensitive algorithms
- Time and memory access pattern obfuscation through techniques like:
  - Blinding of secret values with random masks
  - Memory access pattern normalization
  - Cache timing attack countermeasures through prefetching
- Regular security audits by specialized firms with demonstrated expertise in side-channel analysis
- Hardware-level protections for validators, including:
  - Memory encryption for sensitive operations

- Segregated computation environments
- Power consumption normalization for critical operations

These protections are verified through automated test suites that analyze execution traces for timing correlations and simulated power analysis attacks.

7.1.4 Cross-Layer Security

The interaction between EVM pallet, native WASM runtime, and ZK wrappers creates potential vulnerabilities at interface boundaries within Substrate's architecture. To mitigate these risks, the ecosystem implements formal verification of all cross-layer interactions using the K framework, allowing mathematical proof of correctness for critical state transitions. Additionally, the system enforces strict memory isolation between execution environments through Substrate's sandboxed execution model and software containerization. A dedicated security pallet validates all cross-layer calls before execution, verifying signature validity, parameter bounds, and access permissions through Substrate's native security infrastructure. This comprehensive approach reduces the attack surface at layer boundaries while maintaining the composability benefits of Substrate's multi-layer architecture [19] [44] [108].

8 Tech Stack: ZKP Blockchain

Technology Comparison Table

The following table summarizes various technologies in the context of consensus, application, storage, and cryptography layers.

Category	Technology	Description
Consensus Layer	BABE+GRANDPA	A hybrid consensus combining probabilistic block production with deterministic finality, achieving ∼6-second block production with 1-2 second finality [109, 110].
Consensus Layer	Substrate Framework	A modular framework for building custom blockchains, supporting AI task and storage management through custom pallets with ∼1ms latency [107, 108].
Consensus Layer	Patricia Tries	Radix trees for efficient state management with logarithmic time complexity O(log n) and compressed storage [114].
Consensus Layer	Proof of Intelligence (PoI)	Nodes perform verifiable AI computations, scored as PoI_Score_i = Σ (Accuracy_j × Efficiency_j × Complexity_j), using BABE's VRF assignment.
Consensus Layer	Proof of Space (PoSp)	Leverages decentralized storage for consensus, with nodes providing verifiable storage commitments proven through cryptographic methods [22].
Application Layer	EVM Pallet	Executes Solidity contracts with weight-based metering (∼200,000 weight for zk-SNARKs), ensuring Ethereum compatibility through Frontier [112, 113].
Application Layer	WebAssembly (WASM)	High-performance native runtime ($\sim 10^8$ instructions/s) for AI tasks, integrated with Substrate's Executive pallet [119].
Application Layer	ZK Wrappers	Enable privacy-preserving computations with zk-SNARKs (∼288-byte proofs, 2ms) and zk-STARKs (∼100KB proofs, 10ms).
Application Layer	Circom	Domain-specific language for ZK circuits, compiling AI computations for zk-SNARKs [66].
Application Layer	ZoKrates	Toolbox for zk-SNARKs, supporting modular AI circuit design (e.g., ReLU activations) [67].
Storage Layer	Merkle Trees	Ensure data integrity with ∼384-byte proofs for off-chain AI datasets stored via PoSp [68].

Category	Technology	Description
Storage Layer	IPFS	Content-addressed, decentralized storage network for managing large AI datasets with distributed hash table lookups [12].
Storage Layer	Filecoin	Adds economic incentives to IPFS storage, ensuring long-term data availability for AI workloads [13].
Storage Layer	Off-chain Workers	Substrate's mechanism for secure off-chain computation and data integration [120].
Cryptography	zk-SNARKs	Succinct proofs (∼288 bytes, 2ms verification) for on-chain AI tasks, 128-bit security via BLS12-381 [69].
Cryptography	zk-STARKs	Transparent, post-quantum secure proofs (∼100KB, 10ms) for off-chain tasks, 256-bit security via SHA-3 [70].
Cryptography	ECDSA	Secures transactions and authentication with 256-bit keys, ensuring PoI submission integrity [71].
Cryptography	EDDSA	Future upgrade for faster, side-channel-resistant signatures, enhancing throughput [72].
Cryptography	Homomorphic Encryption	Enables encrypted data computations for privacy-preserving AI (e.g., federated learning) [73].
Cryptography	Poseidon Hash	Optimized hash for ZK circuits, supporting PoSp with 128-bit security ($\alpha = 8$, rounds=57)[74].
Cross-Chain	XCM	Cross-Consensus Message Passing for interoperability within the Polkadot ecosystem [115].

8.1 Detailed Explanations

8.1.1 Consensus Layer Technologies

- BABE+GRANDPA: Drives hybrid consensus with probabilistic block production (BABE) and deterministic finality (GRANDPA), achieving fast finality (∼1–2 seconds) critical for real-time AI validation [109, 110].
- Substrate Framework: Provides modularity through FRAME pallets (e.g., pallet-poi, pallet-posp), using Patricia Tries for low-latency state updates (∼1ms) [107, 108].
- Patricia Tries: Manage state with O(log n) operations and path compression, generating Merkle proofs for tamper-proof PoI scores [114].
- Proof of Intelligence (PoI): Nodes contribute AI computations, scored as $\text{PoI\_Score}_i = \sum(\text{Accuracy}_j \times \text{Efficiency}_j \times \text{Complexity}_j)$, with tasks assigned via BABE's VRF for fairness, initially supporting matrix operations.

- Proof of Space (PoSp): Validates storage commitments from nodes, ensuring data availability for AI applications while offering a more sustainable alternative to energy-intensive consensus mechanisms [22].

8.2 Application Layer Technologies

- EVM Pallet: Executes contracts with SCALE encoding and unified account system, supporting ZK-verified AI tasks through Frontier compatibility [112, 113].
- WASM: Offers near-native speeds ($\sim 10^8$) instructions/s for AI inference, using Rust-based pallets integrated with Substrate's runtime [119].
- ZK Wrappers: Abstract zk-SNARKs (on-chain) and zk-STARKs (off-chain) for secure AI computation verification through both EVM and native pallets.
- Circom: Compiles ZK circuits for AI tasks (e.g., matrix multiplication), optimizing proof generation [66].
- ZoKrates: Simplifies zk-SNARK circuit design with reusable components for AI operations [67].

8.3 Storage Layer Technologies

- Merkle Trees: Generate compact proofs ($\sim$384 bytes) for off-chain data integrity, supporting PoSp-stored AI datasets [68].
- IPFS: Provides content-addressed, decentralized storage for large AI datasets, using a Kademlia-based distributed hash table for efficient content discovery and retrieval [12].
- Filecoin: Extends IPFS with economic incentives for storage providers, ensuring long-term data persistence through cryptographic proof of storage [13].
- Off-chain Workers: Enable secure off-chain computation and data integration within Substrate's runtime environment [120].

8.4 Cryptographic Technologies

- zk-SNARKs: Deliver succinct proofs ($\sim$288 bytes, 2ms) for on-chain verification, using BLS12-381 despite trusted setup needs [69].
- zk-STARKs: Provide transparent, post-quantum proofs ($\sim$100KB, 10ms) for off-chain tasks, leveraging SHA-3 [70].
- ECDSA: Ensures transaction and node authentication security with 256-bit keys [71].
- EDDSA: Planned upgrade for faster, resistant signatures in high-throughput scenarios [72].
- Homomorphic Encryption: Supports encrypted AI computations, enhancing privacy despite overhead [73].
- Poseidon Hash: Optimizes ZK circuit efficiency for PoSp verification, with 128-bit security [74].

8.5 Implementation Details

The tech stack integrates BABE+GRANDPA and Substrate Framework for a hybrid PoI model, with Patricia Tries managing state. PoI tasks (e.g., 100×100 matrix operations, $\sim 10^4$ constraints) use zk-SNARKs for verification, assigned via BABE's VRF. The Application Layer combines EVM pallet and native WASM runtime, with ZK wrappers enabling zk-SNARK/STARK use. Circom and ZoKrates streamline AI circuit design. Cross-chain interoperability is achieved through XCM. Cryptography balances efficiency (zk-SNARKs, ECDSA) and future-proofing (zk-STARKs, EDDSA), with Homomorphic Encryption enhancing privacy.

8.6 Future Enhancements

- Recursive SNARKs: Aggregate proofs to reduce costs [75].
- Parallel Proof Generation: Minimize $T_p = k \times c \times \log c$ across nodes.
- Hardware Acceleration: ASICs for 50-70% energy reduction in ZKP generation.
- EDDSA Adoption: Enhance performance for high-throughput AI validation.
- Parachain Integration: Leverage Polkadot's parachain architecture for horizontal scaling and specialized AI computation chains.
- zk-Rollups: Layer 2 scaling solution to enhance EVM compatibility while maintaining privacy and security [41].

9 Key Innovations of the ZKP Blockchain

The Zero-Knowledge Proof (ZKP) base layer blockchain introduces five core technical advancements designed to support decentralized AI computation, privacy, and scalability. These innovations differentiate it from conventional blockchain systems, establishing a secure and efficient foundation for advanced applications.

9.1 Hybrid Consensus Model: Proof of Intelligence (PoI) and Proof of Space (PoSp)

The blockchain implements a hybrid consensus mechanism integrating Proof of Intelligence (PoI) and Proof of Space (PoSp) with Substrate's native staking system, emphasizing contributions to AI computation and storage over energy-intensive alternatives. This system incentivizes nodes to perform verifiable AI tasks, such as matrix operations, with rewards determined by accuracy, efficiency, and complexity, validated through zk-SNARKs and integrated with BABE+GRANDPA consensus. Meanwhile, PoSp ensures the availability of AI datasets by confirming storage commitments, operating at an energy cost of approximately 10W/TB, creating a balanced ecosystem that rewards both computational and storage contributions through Substrate's modular pallet architecture.

9.2 Zero-Knowledge Proofs for AI Computation Verification

The system incorporates zk-SNARKs and zk-STARKs to enable privacy-preserving verification of AI computations, expanding the application of zero-knowledge proofs beyond transaction privacy. ZK-SNARKs provide compact proofs ($\sim$288 bytes) with rapid verification times ($\sim$2ms) for on-chain validation of AI tasks through both EVM pallet and native Substrate verification infrastructure, making them ideal for blockchain environments where efficiency is critical. Complementing this, zk-STARKs deliver transparent, quantum-resistant proofs for off-chain computations, enhancing long-term security without requiring trusted setup ceremonies. This dual approach ensures both immediate performance and future-proof security within Substrate's flexible runtime environment.

9.3 Modular Architecture with Dual Runtimes: EVM and WASM

The blockchain supports both the Ethereum Virtual Machine through Substrate's EVM pallet and native WebAssembly (WASM) runtime, combining compatibility with Ethereum's ecosystem and high-performance computing capabilities. The EVM pallet facilitates the execution of Solidity-based smart contracts through Frontier compatibility, maintaining interoperability with Ethereum standards and accessing its vast developer ecosystem. Native WASM achieves near-native execution speeds ($\sim 10^8$ instructions/s) for AI tasks, supporting languages such as Rust and C++, enabling compute-intensive applications like neural network inference through Substrate's optimized runtime. This dual-runtime approach maximizes both ecosystem compatibility and performance flexibility while leveraging Substrate's unified account system and cross-runtime communication.

9.4 Energy-Efficient Design with Off-Chain Storage

The integration of PoSp and off-chain storage solutions, such as IPFS and Filecoin, reduces energy consumption and on-chain data congestion through Substrate's off-chain worker infrastructure. PoSp operates at an energy efficiency of $\sim$10W/TB, significantly lower than Proof-of-Work's $\sim$1MW/TH/s, representing a 99% reduction in energy requirements. Off-chain data is represented on-chain as metadata (e.g., Content Identifiers or CIDs) stored in Patricia Tries,

with integrity verified using Merkle Tree proofs (∼384 bytes), ensuring data authenticity without burdening the blockchain with large storage requirements. This approach balances security with environmental sustainability while leveraging Substrate's efficient state management.

9.5 Scalability Through Layered Architecture

A layered design—comprising Consensus, Application, and Storage layers—enhances performance and supports future enhancements within Substrate's modular framework. The Consensus Layer employs BABE+GRANDPA to achieve transaction finality in approximately 1–2 seconds, ensuring rapid agreement across the network with probabilistic block production and deterministic finality. The Application Layer integrates ZK wrappers for privacy-preserving computations through both EVM pallet and native runtime, with potential extensions to recursive SNARKs and hardware acceleration. The Storage Layer manages off-chain data with cryptographic verification through custom pallets and off-chain workers, enabling the system to scale beyond the limitations of traditional blockchains. This separation of concerns allows each layer to evolve independently while maintaining system integrity through Substrate's upgrade mechanisms.

Together, these innovations advance decentralized AI infrastructure by combining zero-knowledge proof-based computation verification, a hybrid consensus model integrated with Substrate's architecture, dual-runtime environment, energy-efficient design, and scalable layered structure. The result is a robust platform for privacy-focused, high-performance applications at the confluence of blockchain and artificial intelligence, leveraging Substrate's proven security and scalability.

10 The ZKP Blockchain - Future of Decentralized AI

The Zero-Knowledge Proof (ZKP) base layer blockchain represents a significant leap forward in the integration of blockchain technology and artificial intelligence. By addressing critical challenges in decentralized systems—privacy, scalability, and energy efficiency—this platform creates new possibilities for secure, collaborative computation at scale.

10.1 A New Paradigm in Blockchain Technology

The ZKP base layer reimagines decentralized systems to meet the demands of modern AI applications through Substrate's advanced architecture. Its hybrid consensus mechanism rewards meaningful computational and storage contributions rather than redundant work, fostering a sustainable ecosystem through custom pallets and native staking integration. The privacy-preserving architecture enables verifiable computations without exposing sensitive data, while the dual-runtime environment balances ecosystem compatibility with high-performance needs through EVM pallet and native WASM execution.

10.2 Transforming Industries Through Decentralized AI

The implications of this technology extend across multiple sectors:

- **Healthcare**: Collaborative AI models can be trained on sensitive medical data without compromising patient privacy, potentially accelerating research and diagnosis through secure multi-party computation.
- **Finance**: Privacy-preserving computations enable secure risk assessments and algorithmic trading while maintaining confidentiality through zero-knowledge verification.
- **Scientific Research**: Researchers can leverage distributed resources for climate modeling or drug discovery without exposing proprietary data or methodologies, facilitated by Substrate's interoperability features.
- **Edge Computing**: Integration with distributed physical infrastructure creates a foundation for real-time analytics and autonomous systems through off-chain workers and cross-chain messaging.

10.3 A Vision for the Future

As the ZKP ecosystem evolves, ongoing research into recursive proof aggregation, hardware optimization, improved circuit designs, and parachain integration will further enhance its capabilities. These advancements will expand the scope of feasible applications, driving innovation at the intersection of blockchain and AI while leveraging Substrate's upgrade mechanisms for seamless evolution.

The ZKP base layer serves as both a practical platform and a proof of concept, demonstrating that decentralized, privacy-preserving AI computation is achievable through the thoughtful integration of zero-knowledge cryptography, consensus mechanisms, and Substrate's layered architecture. By addressing fundamental challenges in blockchain scalability and AI privacy, it provides a foundation for the next generation of decentralized applications within the Polkadot ecosystem and beyond.

11 ZKP Data Marketplace

11.1 Intro

The Data Marketplace is the flagship decentralized application (dApp) within the Zero-Knowledge Proof (ZKP) Blockchain Ecosystem, designed to enable privacy-preserving data sharing and monetization. Built on Substrate's EVM pallet with Frontier compatibility and leveraging Zero-Knowledge Proofs (ZKPs), this marketplace aims to address key limitations of centralized data systems: unauthorized data exploitation, lack of contributor compensation, and privacy vulnerabilities [112] [113].

By enabling data contributors to maintain ownership of their tokenized assets, providing cryptographic privacy through ZKPs, and seeking to align economic incentives with active participation via a native token (DataToken, DTK), the platform strives to offer an alternative approach to data management. Designed to operate on the ZKP Blockchain's Layer 1 (L1) Substrate architecture, it proposes to integrate a hybrid consensus mechanism combining Proof of Intelligence (PoI) for AI-driven dataset validation and Proof of Space (PoSp) for decentralized storage, targeting security, scalability, and data integrity through custom pallets and BABE+GRANDPA consensus, as detailed in subsequent sections and subject to testnet validation.

This initiative seeks to address specific technical shortcomings of centralized systems, such as single points of failure and privacy breaches, and aims to promote a more equitable and transparent data ecosystem through decentralized governance and privacy-preserving mechanisms enabled by Substrate's modular architecture, as explored in this whitepaper, with outcomes to be validated through future testing.

11.2 Motivation

Centralized data systems, as exemplified by tech giants such as Google, Amazon, and Meta, have long operated on a model that exploits user data, leaving it vulnerable to breaches and misuse while concentrating economic benefits among a few intermediaries at the expense of data originators. The 2018 Cambridge Analytica scandal, where 50 million Facebook profiles were harvested without consent, starkly illustrates these vulnerabilities [76]. This incident exposed critical flaws: pervasive privacy infringements, lack of compensation for data contributors, and monopolistic profit retention by dominant entities. Beyond Cambridge Analytica, other breaches—such as the 2017 Equifax incident, which compromised 147 million individuals' personal data [95]—further highlight the systemic risks of centralized data management, including inadequate security and lack of user agency. These events have eroded public trust, underscored the fragility of traditional data frameworks, and catalyzed a global demand for alternatives that prioritize privacy, equity, and transparency.

The ZKP Data Marketplace proposes to address these challenges by decentralizing control and leveraging Substrate's blockchain technology to create a transparent, immutable record of data ownership and transactions stored in Patricia Tries, as a conceptual framework under development in the testnet phase [78] [114]. It employs ZKPs to aim for privacy-preserving verification of dataset attributes through both EVM pallet and native Substrate verification infrastructure, offering a distinct approach compared to conventional encryption by enabling proof-based verification without data exposure, though this is subject to implementation and validation [77] [121].

This proposed design seeks to shift the economic model toward greater contributor involvement, enabling contributors to tokenize their data into tradeable assets and potentially earn rewards through a native token, tentatively named DataToken (DTK), managed through Substrate's native token economics, as part of the testnet exploration. For example, a small business collecting customer feedback data may tokenize this information, list it on the marketplace,

and aim to earn rewards as researchers or marketers access it, retaining ownership and control throughout the process, pending validation of this mechanism. By aiming to redistribute economic value to data creators, the marketplace strives to promote fairness, with cryptographic safeguards like ZKPs and AES-256 encryption designed to reduce risks of unauthorized access and misuse, seeking to enhance trust in data-sharing systems through future testing and refinement.

Moreover, centralized systems often fail to balance utility with privacy, leaving sensitive data exposed during processing or sharing. The Data Marketplace aims to address this by integrating ZKPs through Substrate's verification infrastructure, which are designed to allow verification of dataset properties—such as size, authenticity, or statistical significance—without exposing the underlying data, subject to testnet validation. This capability is intended to be crucial in contexts where data sensitivity is paramount, such as medical research or financial analytics, aiming to allow contributors to share data with reduced risk of privacy breaches once fully implemented. The platform's decentralized design also seeks to reduce the risk of single points of failure through Substrate's distributed architecture, a common vulnerability in centralized systems where a single breach can compromise millions of records, with effectiveness to be assessed in testnet trials.

By proposing to address these multifaceted issues, the Data Marketplace strives to mitigate the shortcomings of centralized frameworks and lay the groundwork for a more ethical, secure, and inclusive data economy, with outcomes to be validated through future testing and deployment.

11.3 Core Concepts

The Data Marketplace builds upon the ZKP Blockchain L1 Substrate infrastructure with the following key elements:

- **Tokenized Datasets**: Transform data into tradeable digital assets, using Substrate's native token standards for divisible datasets and unique asset pallets for non-fungible datasets [83] [123]. This enables data creators to monetize contributions while maintaining ownership through Substrate's account system.
- **Granular Access Control**: A multi-tiered permission system authenticated through ZKPs governs access based on coins ownership, implemented through custom Substrate pallets. Tiers range from free metadata previews (Tier 0) to full dataset access (Tier 5), with ZKPs enabling eligibility verification without exposing sensitive user details [77].
- **Quality Assurance Framework**: A multi-layered approach combining reputation systems managed through native pallets, validation networks, and automated audits ensures data quality and reliability [79].
- **Decentralized Governance**: Data DAOs oversee shared datasets using ZKP-secured voting mechanisms implemented through Substrate's democracy pallet, enabling transparent, democratic management of marketplace policies [84] [124].

11.4 High-Level Overview of Components and Architecture

Note: Testnet Development Notice: *The Data Marketplace described in this section is currently under development in the testnet phase. While the core architecture and integration with the ZKP Blockchain's base layer are well-defined, specific implementation details, parameters, and performance metrics are subject to validation and refinement through testnet trials. Quantitative figures presented throughout this section should be understood as design targets aligned with the base layer specifications rather than empirically validated metrics, unless otherwise stated.*

The Data Marketplace architecture integrates with the ZKP Blockchain's Substrate infrastructure through several key components:

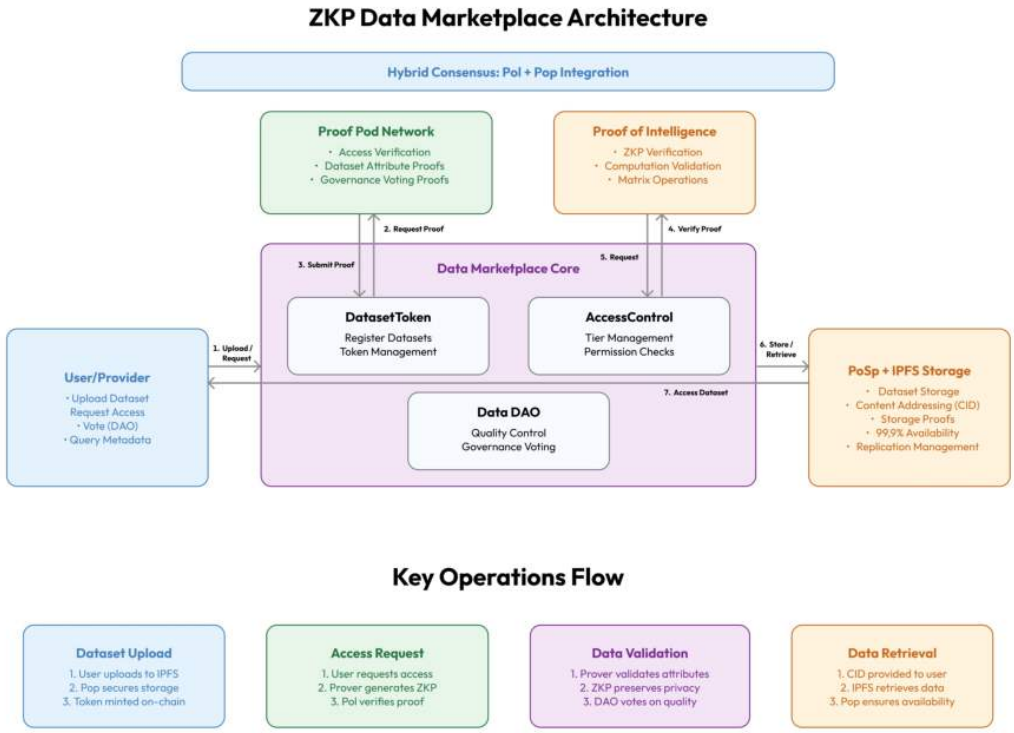

Figure 12: Data Marketplace Architecture

11.4.1 EVM Pallet Integration

The marketplace operates on Substrate's EVM pallet with Frontier compatibility, enabling Solidity smart contracts to facilitate tokenization, access control, and reward distribution with minimal trust assumptions [80] [112] [113]. Weight-based fees (e.g., equivalent to 3 gas for basic operations, 20,000 gas for storage converted to Substrate's weight system) support validators and storage providers through Substrate's fee distribution mechanism [117].

11.4.2 Hybrid Consensus Utilization

The marketplace leverages the ZKP Blockchain's hybrid consensus mechanism, combining Proof of Intelligence (PoI) and Proof of Space (PoSp) integrated with BABE+GRANDPA consensus.

PoI uses mathematical operations with deterministic outputs for computational validation, including matrix operations (up to 100×100 dimensions), element-wise activation functions for vectors up to 1,000 elements, and simple feedforward inference for models up to 10^5 parameters. PoSp ensures data availability through cryptographic storage proofs managed by custom Substrate pallets [79] [109] [110].

11.4.3 Off-Chain Storage with IPFS

Large datasets are stored off-chain using IPFS, supplemented by a custom replication management layer implemented through Substrate's off-chain workers that implements periodic health checks and incentive-driven replication [120]. Specifically, the marketplace includes:

- A tracker service that monitors content addressable storage (CAS) object availability across the network through off-chain worker coordination
- Economic incentives tied to the PoSp mechanism managed through custom pallets that reward storage providers proportionally to both their uptime and replication support
- A redundancy coordinator implemented as a Substrate pallet that automatically initiates new storage requests when availability metrics drop below threshold

Through these mechanisms, the system targets 99.9% availability while acknowledging that actual performance will require extensive testnet validation. This hybrid model stores metadata in Patricia Tries while keeping data off-chain, reducing blockchain bloat while maintaining decentralized storage principles.

11.4.4 Smart Contract Operations

Core marketplace functions are governed by smart contracts through the EVM pallet incorporating weight-efficient ZKP verification (200,000 gas equivalent weight per verification) [82]. Event logging enables transparency through indexed logs tracking marketplace activities, stored in Substrate's event system [80] [116].

11.5 Use-Cases

The Data Marketplace is designed to support a wide range of applications across industries, aiming to address diverse needs while prioritizing privacy, security, and equity, as a conceptual framework under development in the testnet phase:

11.5.1 Healthcare Research

The marketplace proposes to facilitate the sharing of tokenized, anonymized patient records for medical research, subject to testnet validation. For instance, a hospital may tokenize a dataset of 5,000 diabetic patient profiles, with the goal of allowing researchers to verify its statistical significance (e.g., average blood glucose levels) via ZKPs through Substrate's verification infrastructure without accessing individual records. This is designed to aim for compliance with privacy regulations like GDPR while potentially supporting advancements in drug development or epidemiological studies, pending implementation and testing.

11.5.2 Artificial Intelligence Development

AI developers are envisioned to access high-quality, curated datasets for model training without compromising data privacy, as a conceptual design. A tech company may procure a tokenized dataset of user-generated text, with quality verification proposed through ZKPs verified by native Substrate pallets, to train a natural language processing model, subject to testnet validation. The marketplace's privacy-preserving design aims to maintain the confidentiality of sensitive user data, seeking to promote ethical AI innovation, with effectiveness to be assessed in future testing.

11.5.3 Financial Analytics

Financial institutions are proposed to use the marketplace to share anonymized transaction data for fraud detection and risk analysis, as a conceptual framework. A bank may tokenize a dataset of credit card transactions, with the goal of enabling a fintech firm to analyze patterns and detect anomalies without exposing customer details, pending validation. ZKP-verified attributes are designed to aim for dataset relevance, while tiered access controls managed through custom pallets seek to balance utility with security, subject to testnet trials.

11.5.4 Environmental Data Sharing

Environmental organizations are envisioned to tokenize climate data, such as air quality measurements, for use by policymakers or researchers, as a conceptual design. For example, a research institute may tokenize a dataset of CO_2 emissions across 50 cities, aiming to enable a government agency to analyze trends and develop sustainability policies, with PoSp targeting data availability and ZKPs aiming to protect contributor privacy, all subject to testnet validation.

11.5.5 Education and Academia

Academic institutions are proposed to share educational datasets, such as student performance metrics, for research purposes, as a conceptual framework. A university may tokenize a dataset of exam scores across 10,000 students, with the goal of allowing educational researchers to study learning outcomes without compromising student privacy, pending implementation. The marketplace's granular access control implemented through Substrate pallets is designed to aim for restricted access to authorized researchers, while metadata previews are intended to be available to all, with effectiveness to be assessed in testnet trials.

11.6 User Interactions: Purchaser and Uploader of Datasets

This section outlines the proposed technical interactions of data uploaders and purchasers within the Data Marketplace, focusing on their potential roles as both users and consumers in a decentralized ecosystem, as a conceptual framework under development in the testnet phase. It highlights the actions they are designed to perform, emphasizing the processes of data uploading, selling, purchasing, and accessing, intended to be facilitated by the platform's Substrate infrastructure, subject to future validation and refinement.

Note: *Token values mentioned (e.g., 5,000 DTK or 1,500 DTK per MB) are symbolic, as the marketplace is currently in the testnet phase, and all figures are subject to change during development and optimization.*

11.6.1 Data Uploader Capabilities

- **Tokenize and Upload Data**: Convert datasets into native Substrate tokens or unique assets by uploading to IPFS with PoSp security managed through custom pallets [83] [87] [123]. Tokenization is executed via EVM pallet smart contracts or native pallets costing approximately 250,000 weight equivalent [81].
- **Set Access Tiers**: Define multiple access levels, from free metadata previews (Tier 0) to full access (Tier 5), managed through Substrate pallets with fine-grained permission controls.
- **Earn Rewards**: Receive DTK tokens based on dataset characteristics and user access patterns, incentivizing high-quality contributions through Substrate's native reward mechanisms [80].
- **Vote on Governance Policies**: Participate in Data DAOs using ZKP-secured voting through Substrate's democracy pallet to influence platform policies while preserving voter privacy [84] [124].

11.6.2 Data Purchaser Capabilities

- **Browse Metadata**: Access dataset information at Tier 0 without token requirements, allowing relevance evaluation before purchase through Substrate's query interface.
- **Verify and Request Access**: Use ZKPs to prove token holdings without revealing exact balances, with verification costing $\sim$200,000 weight equivalent through Substrate's verification infrastructure [82]. Specify access duration and level through smart contracts or native pallets.
- **Access and Download Data**: Retrieve datasets from IPFS using on-chain content identifiers (CIDs) stored in Patricia Tries, with PoSp ensuring availability [87] [114].
- **Utilize Data with Privacy Assurance**: Use datasets for intended purposes with ZKPs verifying legitimate access through Substrate's verification system and PoI validating dataset attributes [77] [79].

11.6.3 Technical Interactions

The interaction between uploaders and purchasers is facilitated by the marketplace's Substrate infrastructure through a well-defined workflow:

1. **Uploader Tokenization**: The uploader submits a dataset via the `mintDataset` function in the `DatasetToken` contract or native pallet, specifying its CID, size, and access tiers. The system validates the CID against IPFS, encrypts the dataset with AES-256 (GCM mode), and stores metadata in Patricia Tries, costing 245,000 weight equivalent in testnet simulations [80] [114].

2. **Purchaser Browsing**: The purchaser queries metadata at Tier 0 using the `getDataset` function or pallet call, retrieving details like schema and statistical summaries without cost.

3. **Access Request**: The purchaser requests access via the `accessDataset` function or pallet extrinsic, submitting a zk-SNARK proof of token holdings and specifying duration (e.g., 30 days) and tier (e.g., Tier 3). Off-chain workers dynamically adjust tier requirements based on demand, using a median-based aggregation of access frequency (e.g., 800 accesses/hour in testnet) to ensure fair allocation [92] [120].

4. **Access Granting**: Upon verification through Substrate's verification infrastructure, the system grants access, emitting an `AccessExtended` event (375 weight) through Substrate's event system and providing the purchaser with the CID for IPFS retrieval, secured by PoSp [116].

11.7 Technical Underpinnings: EVM Pallet and ZKPs

This section describes how Substrate's EVM pallet is proposed to be used in the Data Marketplace, focusing on its architecture and operations for data management within the ZKP Blockchain Ecosystem, as a conceptual framework under development in the testnet phase.

11.7.1 EVM Pallet Architecture and Operations

Computational Framework: Substrate's EVM pallet executes smart contracts for marketplace operations using a stack-based architecture (max depth 1024), with 256-bit addressable memory and unified state storage in Patricia Tries through Substrate's state management system [85] [112] [114]. PoSp commitments verify off-chain data storage reliability through custom Substrate pallets [79].

Storage Model and Off-Chain Integration: The EVM pallet's key-value storage model (32-byte pairs) is supplemented by off-chain IPFS storage secured by PoSp, managed through Substrate's off-chain workers, balancing on-chain efficiency with off-chain scalability [85] [87] [120].

Weight Mechanics: Substrate's weight-based fee model manages computational resources with operation costs converted from gas (e.g., MUL: 5 gas equivalent, CALL: 700 gas equivalent) and distributes fees to validators and storage providers through Substrate's native fee distribution mechanism [81] [117]. ZKP verification costs approximately 200,000 gas equivalent weight as specified in the base layer.

Contract Deployment: Smart contract deployment through the EVM pallet requires 7-12 million gas equivalent weight, reflecting the integration of ZKPs, PoI validation, and PoSp storage within Substrate's runtime environment [80].

Weight Optimization Strategies: The marketplace design incorporates several optimization approaches within Substrate's framework:

- **Batch Processing**: Combining multiple operations into single transactions through Substrate's batch utilities to reduce overhead [86].
- **Precompiles**: Leveraging specialized EVM pallet functions for ZKP verification, which enable the base verification cost of 200,000 gas equivalent weight as specified in the base layer [82].

11.7.2 Zero-Knowledge Proofs: Cryptographic Foundations

This section outlines the cryptographic foundations of the Data Marketplace, focusing on the role of Zero-Knowledge Proofs (ZKPs), specifically zk-SNARKs (Zero-Knowledge Succinct Non-Interactive Arguments of Knowledge), in supporting privacy-preserving data interactions. Integrated with Proof of Intelligence (PoI) validation as defined in the base layer and verified through Substrate's verification infrastructure, zk-SNARKs enable verification of sensitive dataset attributes without disclosure, offering a framework for trust in a decentralized ecosystem.

zk-SNARKs and Privacy Framework: zk-SNARKs form the cryptographic foundation of the Data Marketplace, enabling verification of dataset attributes without revealing underlying data through both EVM pallet and native Substrate verification mechanisms [77] [121]. This privacy-preserving mechanism maintains confidentiality of sensitive information while promoting trust among participants. For instance, a provider can prove a dataset's statistical properties,

such as the mean sales figures of a retail dataset, to a buyer without disclosing individual transaction details. This capability is particularly vital in sectors like healthcare and finance, where data sensitivity necessitates stringent privacy measures.

Mathematical Foundations: zk-SNARKs rely on advanced cryptographic constructs to achieve their succinct and non-interactive properties, making them efficient and practical for blockchain applications within Substrate's runtime.

Elliptic Curve Pairings: The mathematical basis of zk-SNARKs includes elliptic curves, specifically BLS12-381 curves, which facilitate pairing operations (e.g., $e(G_1, G_2)$) [88]. These pairings enable the construction of succinct proofs, a critical feature for verification efficiency. The Data Marketplace utilizes the BLS12-381 curve due to its security characteristics and implementation efficiency within Substrate's cryptographic infrastructure. This curve provides approximately 128-bit security against discrete logarithm attacks while enabling efficient pairing computations required for zk-SNARK verification.

Polynomial Commitments via QAPs: Another foundational element is the use of Quadratic Arithmetic Programs (QAPs) to encode computational logic [88]. QAPs transform a circuit's computations into a set of polynomial equations, targeting proofs of size $O(\log n)$ for circuits with n gates. This logarithmic scaling balances efficiency and security, ensuring that proof size grows minimally even as circuit complexity increases.

11.7.3 Lifecycle of zk-SNARKs

The operational lifecycle of zk-SNARKs encompasses three key phases, each critical to their application in the Data Marketplace within Substrate's architecture.

Trusted Setup Phase: The lifecycle begins with a trusted setup, where a Common Reference String (CRS) is generated through a multi-party computation (MPC) ceremony involving a minimum of 20 participants as specified in the base layer, achieving a collusion risk below 2^{-128} [89]. The security of this process relies on the assumption that at least one participant is honest and properly discards their secret contribution. The ceremony coordination is managed through Substrate's governance mechanisms, ensuring transparency and community oversight.

Proof Generation Phase: Proof generation occurs off-chain and is performed by specialized entities called provers, which are part of the marketplace's dedicated infrastructure coordinated through Substrate's off-chain workers. For a standard 10,000-gate circuit, proof generation requires approximately 10 seconds on standard hardware as specified in the base layer [88]. The computational complexity of this process scales with circuit size and structure, making off-chain execution necessary for complex operations.

In the marketplace ecosystem, provers serve as dedicated computational resources that generate zero-knowledge proofs for various operations including access control verification, dataset attribute validation, and governance participation. Users requesting operations that require ZKP verification (such as dataset access) send their requests to the prover network, which then generates the necessary proofs using circuits like AccessVerifier. These provers stake DTK tokens as collateral through Substrate's staking mechanisms to ensure reliable service and are incentivized through rewards derived from marketplace fees.

This separation of proof generation (handled by provers) from verification (performed on-chain through Substrate's verification infrastructure) addresses the computational asymmetry inherent in zero-knowledge systems, where proof generation is significantly more resource-intensive than verification. By delegating the computational burden to a specialized off-chain network

of provers, the marketplace ensures efficiency while maintaining the security guarantees of the underlying cryptographic protocols.

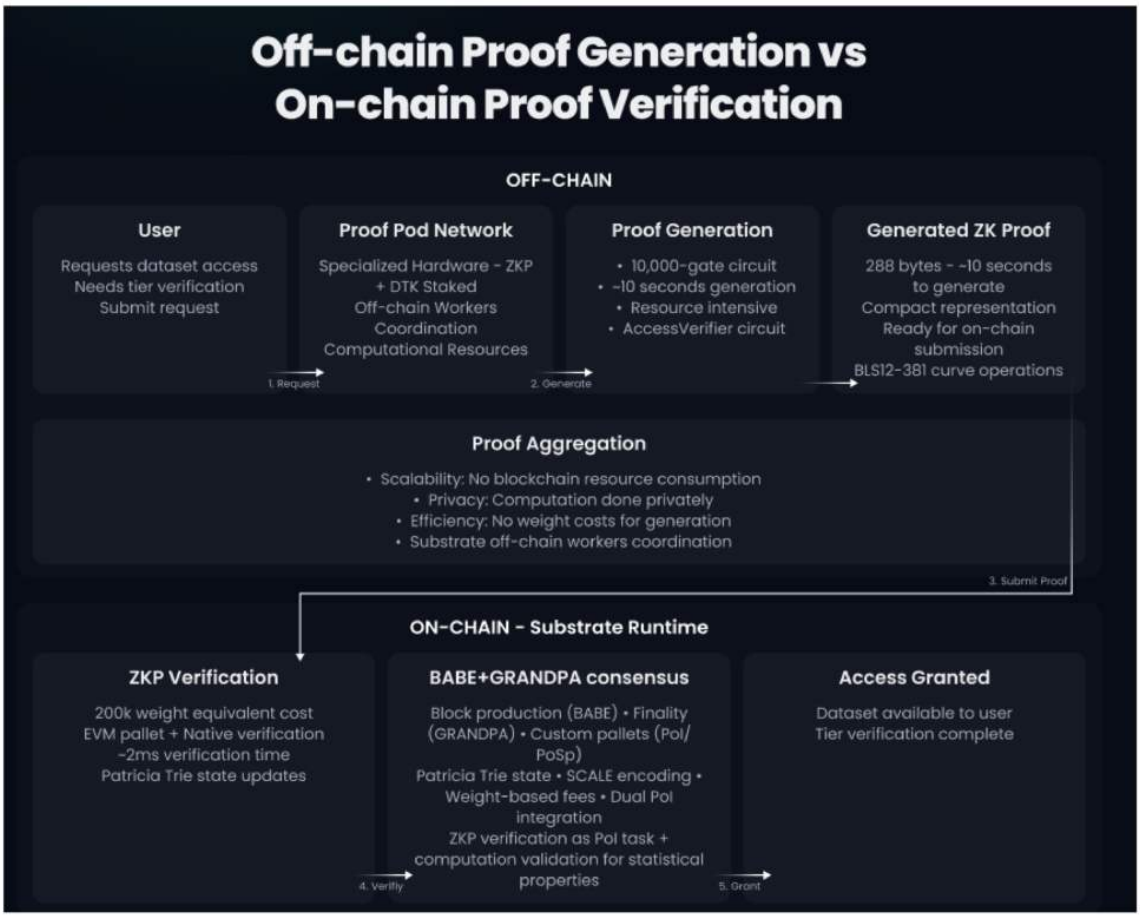

Figure 13: Proof Generation and Verification

Verification Phase with PoI Integration: Verification is designed to take place on-chain, with an estimated cost of 200,000 gas equivalent weight (consistent with base layer specifications) and utilizing EVM pallet precompiles and native Substrate verification pallets for efficiency [82] [121]. The Data Marketplace leverages the base layer's Proof of Intelligence (PoI) framework in two distinct ways:

- **ZKP Verification as PoI Task:** The act of verifying zero-knowledge proofs is considered a PoI-eligible operation, allowing validators to earn rewards for performing these cryptographic verifications through the hybrid consensus system.
- **Computation Validation:** For more complex operations, additional PoI tasks validate specific mathematical computations related to datasets, such as verifying statistical properties or model outputs.

This dual integration ensures that the marketplace's operations align with the PoI incentive structure while maintaining the security guarantees of ZKP verification within Substrate's modular architecture.

The verification process is remarkably efficient compared to proof generation, requiring only a few milliseconds of computation. This asymmetry is a key advantage of zk-SNARKs in blockchain environments, allowing for complex computations to be validated on-chain without imposing excessive computational burdens on validators.

11.7.4 Security Guarantees

zk-SNARKs provide security guarantees essential for the marketplace's privacy framework, consistent with the base layer's cryptographic foundation and verified through Substrate's security infrastructure. Soundness ensures that false proofs are rejected with a probability less than 2^{-80}, maintaining the same security level used throughout the ZKP Blockchain ecosystem [77]. The zero-knowledge property conceals all inputs beyond public outputs, promoting confidentiality while preserving the integrity of sensitive data.

11.7.5 Practical Applications in the Data Marketplace

The application of zk-SNARKs in the Data Marketplace supports secure, privacy-preserving data interactions across multiple use cases:

- **Access Control:** Users can prove their eligibility for specific access tiers without revealing token balances through Substrate's verification system.
- **Dataset Validation:** Providers can verify dataset attributes without exposing the underlying data.
- **Governance Participation:** Token holders can participate in DAO voting through Substrate's democracy pallet without revealing their identity or voting choice.
- **Compliance Verification:** Data providers can demonstrate regulatory compliance without exposing sensitive information.

These applications demonstrate how zk-SNARKs enable the marketplace to balance privacy with verifiability, creating a foundation for trusted data exchange in a decentralized environment.

11.8 Tokenized Datasets: Comprehensive Mechanisms

This section explores the mechanisms for tokenizing datasets within the Data Marketplace, converting raw data into blockchain-native, tradeable assets. The approach integrates Proof of Space (PoSp) for decentralized storage security, the InterPlanetary File System (IPFS) for data distribution, and Zero-Knowledge Proofs (ZKPs) for privacy-preserving verification within Substrate's modular architecture.

11.8.1 Tokenization Overview

Tokenization transforms datasets into digital assets on the blockchain, enabling secure, transparent trading while ensuring data integrity and confidentiality. This process leverages IPFS for decentralized storage, PoSp for persistent data availability through custom Substrate pallets, and ZKPs for privacy through Substrate's verification infrastructure, creating a robust ecosystem where data providers can efficiently tokenize assets, and consumers can access them securely.

IPFS enhances data integrity through its content-addressing system, which generates unique cryptographic hashes known as Content Identifiers (CIDs) for each dataset. For example, a dataset uploaded to IPFS receives a CID derived from its content using a hash function. Any alteration to the dataset changes its CID, making tampering immediately detectable. This immutability is crucial for trust in a decentralized marketplace. PoSp builds on this by requiring storage nodes to prove they maintain the data over time, using cryptographic challenges based on disk space rather than computational power, managed through Substrate's consensus integration. This incentivizes nodes to ensure data persistence in a trustless manner, reducing dependency on centralized infrastructure.

The marketplace's tokenization framework addresses several critical challenges that traditional data markets face:

- **First,** it solves the data sovereignty problem by giving providers continued control over their assets even after monetization through Substrate's account system. Unlike centralized platforms where data is often copied and stored on the provider's servers, tokenized datasets remain under the cryptographic control of their creators, who can revoke access, update content, or change terms at any point.

- **Second,** it enables granular permissions without trusted intermediaries through Substrate's pallet system. Through the combination of blockchain-based access control and cryptographic verification, the system can enforce complex permission structures without requiring either party to trust a central authority to honor the agreement.

- **Third,** it creates transparency in data provenance and usage through Substrate's immutable storage. The immutable record of dataset creation, modification, and access provides an audit trail that can be crucial for compliance, quality verification, and dispute resolution in data markets.

The tokenization workflow encompasses several stages: data ingestion, encryption, metadata structuring, and registration via smart contracts or native pallets. Each step balances computational efficiency with robust security, ensuring datasets remain accessible only to authorized parties while safeguarding sensitive information from unauthorized exposure.

11.8.2 Data Ingestion Process

IPFS Storage and PoSp Security: The tokenization process begins with data ingestion, where datasets are uploaded to IPFS, a content-addressable storage network that enables decentralized data retrieval, coordinated through Substrate's off-chain workers [87] [120]. This generates a Content Identifier (CID)—a cryptographic hash serving as an address and integrity

check. For redundancy and fault tolerance, the system leverages erasure coding as described in the base layer specifications, allowing data reconstruction even when some shards are unavailable. PoSp enhances this by requiring nodes to provide storage proofs through custom Substrate pallets, ensuring data persistence in a manner akin to a distributed vault with redundancy [79].

The IPFS implementation in the Data Marketplace leverages several key features of the protocol to enhance security and efficiency:

- Content addressing through cryptographic hashing ensures that data retrieval requests specify exactly what is being requested, not where it is located. This removes the need to trust specific storage providers and enables content verification upon receipt.
- Deduplication through content addressing automatically eliminates redundant storage of identical data, improving efficiency across the network. If two datasets contain overlapping information, only the unique portions consume additional storage resources.
- The Merkle Directed Acyclic Graph (DAG) structure of IPFS divides data into blocks linked by cryptographic hashes, enabling efficient verification, transfer, and caching of dataset components. This structure allows incremental verification and transfer of large datasets, enhancing performance for multi-gigabyte AI training datasets.

To ensure long-term storage reliability, PoSp requires nodes to periodically demonstrate possession of the dataset through cryptographic challenges managed by Substrate pallets [79]. Unlike Proof of Work, which relies on computational effort, PoSp leverages disk space, aligning with the marketplace's goal of resource efficiency. This mechanism discourages data loss or tampering, as nodes are incentivized through rewards to maintain data integrity.

Encryption and ZKP Ownership Verification: Prior to ingestion, datasets are encrypted using AES-256, a symmetric encryption algorithm with a 256-bit key, widely regarded for its security and efficiency [90]. With a keyspace of 2^{256} possible combinations, AES-256 is resistant to brute-force attacks, even with significant computational advancements. It also withstands sophisticated cryptographic attacks, such as differential cryptanalysis, due to its robust substitution-permutation network. This ensures that data remains confidential during storage on IPFS and transit across the network, protecting sensitive information like medical records or proprietary models.

The encryption implementation uses the Galois/Counter Mode (GCM) of operation, which provides both confidentiality and authentication, protecting data from both disclosure and tampering. This mode offers several advantages for the Data Marketplace context:

- Authenticated encryption ensures that only parties with the correct key can modify the data, preventing unauthorized alterations during storage or transit.
- Parallelizable design enables efficient encryption and decryption of large datasets, crucial for AI applications that may involve gigabytes or terabytes of data.
- Minimal expansion of ciphertext compared to plaintext (only by the authentication tag size) maintains storage efficiency, important when dealing with large-scale datasets.

Ownership verification leverages zk-SNARKs through Substrate's verification infrastructure, allowing providers to prove they possess the encryption key without disclosing it [77] [121]. In a typical scenario, a provider generates a zk-SNARK proof demonstrating knowledge of the AES-256 key for a dataset, which is then verified on-chain without revealing the key itself. As specified in the base layer, proof generation for a standard 10,000-gate circuit requires approximately 10 seconds, with on-chain verification costing 200,000 gas equivalent weight. This zero-knowledge approach ensures privacy in a trustless environment, a critical feature for the marketplace.

Metadata Structuring: Metadata serves two distinct purposes in the tokenized datasets framework:

Public Metadata: Basic descriptive information such as schema definitions, creation timestamps, and general categorization is stored on-chain in Patricia Tries in plaintext to enable discovery and validation. This information is hashed using SHA-512 to ensure integrity, but the hash is stored alongside the plaintext data.

Privacy-Sensitive Metadata: Statistical summaries, detailed provenance information, and other potentially sensitive metadata can be encrypted and stored off-chain through Substrate's off-chain workers, with only hash references maintained on-chain. For this sensitive metadata, zero-knowledge proofs can selectively verify properties without revealing the underlying information.

This dual approach balances the need for discoverability with privacy protection, acknowledging that hashing alone does not provide confidentiality for on-chain information.

The marketplace implements a standardized metadata schema that balances comprehensiveness with efficiency. The schema includes:

- **Dataset identification:** Basic information like title, description, version, and creation timestamp
- **Technical specifications:** Format, size, encoding, compression method, and schema definition
- **Quality indicators:** Completeness, consistency metrics, update frequency, and last verification date
- **Domain-specific attributes:** Field-relevant indicators like resolution for images, sampling rate for audio, or collection methodology for surveys
- **Usage terms:** License type, attribution requirements, and permitted use categories

This structured approach enables efficient discovery and evaluation of datasets while ensuring that critical information is consistently available across the marketplace. The metadata serves several crucial functions in the tokenization process:

- It enables efficient dataset discovery without requiring access to the full data
- It provides the basis for quality assessment and validation before purchase
- It facilitates provenance tracking and attribution for regulatory compliance
- It documents the technical requirements for utilizing the dataset effectively

Consider a 500 MB dataset of weather records: its metadata might include the source, schema (e.g., columns for temperature, humidity, wind speed), statistical summaries (e.g., average temperature of 15°C), and a timestamp. The metadata is serialized, hashed with SHA-512 to produce a fixed-length digest, and linked to the CID. This allows consumers to verify the dataset's authenticity and relevance before purchase.

11.8.3 Smart Contract for Dataset Tokenization

Smart contracts are the backbone of dataset tokenization, registering assets on the blockchain for transparency and control. The `DatasetToken` contract exemplifies this, integrating IPFS and PoSp mechanisms within Substrate's EVM pallet:

Contract Structure:

```
pragma solidity ^0.8.0;
contract DatasetToken {
    struct Dataset {
        string cid;          // IPFS Content Identifier
        uint256 size;        // Dataset size in bytes
        address provider;    // Data provider's address
        uint256 timestamp;   // Registration timestamp
        bool isActive;       // Status flag
    }

    mapping(uint256 => Dataset) public datasets;
    uint256 public datasetCount;
    address public contractOwner;
    mapping(address => bool) public approvedProviders;
    uint256 public registrationFee;

    event DatasetRegistered(uint256 indexed datasetId,
        address indexed provider, string cid, uint256 timestamp);
    event DatasetRevoked(uint256 indexed datasetId,
        address indexed provider, uint256 timestamp);
}
```

The Dataset struct captures essential attributes: the CID links to IPFS storage, size informs resource requirements, provider identifies the owner, timestamp tracks registration, and isActive indicates availability. State variables track datasets and providers, while events ensure auditability through Substrate's event system.

Core Functions The contract includes essential operations for dataset management:

- `mintDataset`: Registers a dataset, requiring the registration fee and storing the Dataset struct.
- `getDataset`: Retrieves dataset details by ID, ensuring transparency for consumers.
- `revokeDataset`: Deactivates a dataset, restricted to its provider, clearing the CID and setting isActive to false.

The smart contract implementation incorporates several security patterns to protect both providers and consumers:

- Access control modifiers restrict sensitive operations to authorized parties
- Event emissions create an immutable audit trail of all significant actions through Substrate's event system
- Input validation prevents malformed data or invalid parameters
- Weight optimization techniques reduce transaction costs for high-volume operations

11.8.4 Lifecycle Management

The marketplace supports comprehensive dataset lifecycle management through:

Revocation Mechanism:

```
function revokeDataset(uint256 datasetId) external {
    Dataset storage ds = datasets[datasetId];
    require(ds.provider == msg.sender, "Not dataset provider");
    require(ds.isActive, "Dataset already revoked");
    ds.cid = "";
    ds.isActive = false;
    emit DatasetRevoked(datasetId, msg.sender, block.timestamp);
}
```

This function deactivates datasets, clearing the CID and halting access. This is akin to retracting a digital publication, with implications for existing access rights.

11.8.5 Archival and Versioning

The marketplace design includes provisions for long-term storage options, such as integration with Filecoin, which complements IPFS with economic incentives for data persistence [91]. IPFS also supports multiple versions through unique CIDs, enabling reversion to prior states [87]. For instance, a dataset might have versions v1.0 and v1.1, each immutable and verifiable. This ensures data provenance, allowing consumers to access specific iterations.

The versioning system serves several important functions in the data lifecycle:

- It preserves the historical record of dataset evolution, crucial for research reproducibility
- It enables consumers to access specific versions that match their requirements
- It allows providers to update datasets while maintaining backwards compatibility
- It creates a foundation for differential pricing based on version recency or features

11.8.6 Tiered Access Control

The marketplace implements a granular access system, tied to token ownership through Substrate's account system:

Access Tiers: Range from Tier 0 (free metadata) to Tier 5 (full dataset access), with escalating token requirements. Tier 0 might provide metadata and a sample, while Tier 5 unlocks the entire dataset.

Dynamic Adjustments: Access tier requirements are designed to adapt to changing market conditions through a three-component oracle system coordinated by off-chain workers:

- **On-chain Analytics Oracle:** Monitors objective metrics like access frequency, tier utilization rates, and token flows directly from blockchain data stored in Patricia Tries.
- **Decentralized Price Feed:** Aggregates external market data from multiple independent sources to establish fair value benchmarks for different data categories.
- **Governance-Bounded Automation:** Implements adjustment algorithms with parameter limits set by governance votes through Substrate's democracy pallet, ensuring that automated changes remain within community-approved boundaries.

This oracle design addresses key security considerations including manipulation resistance through median-based aggregation, sybil resistance through stake-weighted reporting, and failsafe mechanisms that limit adjustment magnitude within any single time period.

Practical Implications These mechanisms enable secure, verifiable data trading across multiple domains. For example, a healthcare provider could tokenize anonymized patient records, encrypt them with AES-256, upload to IPFS with PoSp enforcement, and attach ZKP-verified metadata. This tokenized asset could then be listed on the marketplace, with privacy preserved and availability guaranteed.

Another scenario involves tokenizing a machine learning model. A data scientist might encrypt a convolutional neural network trained on image data, upload it to IPFS, and register it with metadata detailing its architecture and performance metrics. Consumers could purchase access, using ZKPs to verify attributes like accuracy without exposing the model's weights.

The tokenized dataset framework creates a comprehensive system for managing the complete lifecycle of data assets, from creation and registration through versioning, access control, and eventual archival or revocation, all within a privacy-preserving, decentralized environment enabled by Substrate's modular architecture.

11.9 Provers in the Data Marketplace: Specialized Hardware for ZKP Generation

The Data Marketplace incorporates a dedicated prover network to support privacy-preserving operations through efficient ZKP generation. This section outlines the role of provers, their integration with the marketplace infrastructure, and the specialized hardware designed to optimize this critical function.

11.9.1 Prover Function and Architecture

Provers serve as dedicated off-chain entities coordinated through Substrate's off-chain workers that generate zero-knowledge proofs for marketplace operations such as access control verification, dataset attribute validation, and governance participation [120]. When a user requests access to a dataset, a prover generates the necessary proof using circuits like AccessVerifier, enabling verification of eligibility without exposing sensitive details.

The proof generation process occurs off-chain to minimize blockchain computational demands, with verification performed on-chain using EVM pallet precompiles and native Substrate verification pallets for elliptic curve operations. This separation of concerns aligns with the base layer's architecture, where ZKP verification costs 200,000 gas equivalent weight as specified in the technical parameters [82] [121].

For each marketplace operation requiring privacy preservation, the corresponding ZKP circuit defines the computational logic to be verified. For instance, the AccessVerifier circuit validates that a user meets tier requirements without revealing their token balance, while DatasetAttributeVerifier confirms statistical properties without exposing raw data. Provers execute these circuits with the appropriate inputs, generating cryptographic proofs that can be efficiently verified on-chain through Substrate's verification infrastructure.

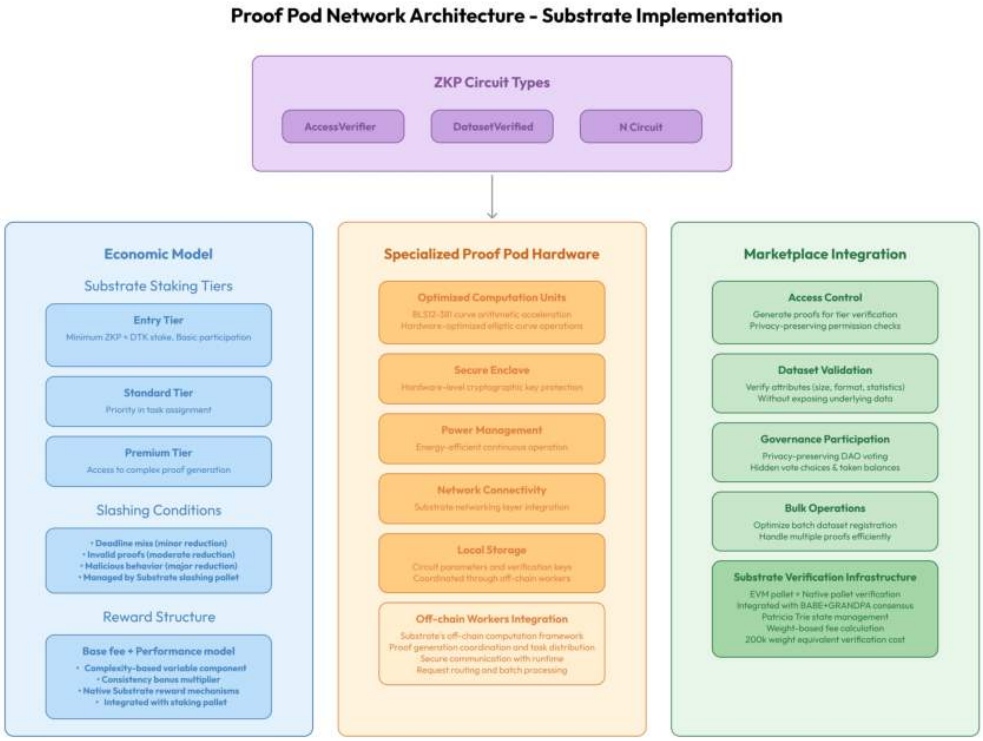

Figure 14: Prover Network Architecture

11.9.2 Specialized Hardware Implementation

To optimize the proof generation process, the Data Marketplace ecosystem includes purpose-built hardware devices specifically designed for ZKP operations.

The roadmap for these specialized proving devices includes:

Optimized Computation Units Hardware configured for efficient execution of elliptic curve operations required by BLS12-381 curve arithmetic in zk-SNARK proof generation. Initial implementations will leverage existing CPU/GPU architectures with software optimizations, while the longer-term roadmap explores potential semi-specialized hardware acceleration.

These devices will focus on accelerating the most computationally intensive operations in ZKP generation, with performance targets to be established through testnet benchmarking. As the proving network matures, hardware capabilities will evolve to address performance bottlenecks identified during testnet operation.

Secure Enclave Architecture Hardware-level protection for cryptographic keys and operations, preventing extraction of sensitive parameters. This ensures that the proving process maintains strong security guarantees even in potentially untrusted environments.

Efficient Power Management Optimized energy consumption for continuous proof generation operations, enabling extended operation while minimizing resource requirements.

Network Connectivity Dedicated communication channels for receiving proof generation requests through Substrate's networking layer and submitting completed proofs to the blockchain network.

Local Storage Secure storage for commonly used circuit parameters and verification keys, reducing the overhead of repeated operations.

These specialized devices serve as dedicated provers in the marketplace ecosystem, staking ZKP and DTK tokens through Substrate's native staking mechanisms as a commitment to participation and reliability. This approach creates a purpose-built infrastructure layer that enhances the performance and security of the overall system while reducing the barriers to participation in proof generation.

The hardware specifications are tailored to the computational requirements of zk-SNARK proof generation for the marketplace's specific circuits. By focusing on these particular operations rather than general-purpose computing, the prover hardware achieve significantly better performance-per-watt than conventional hardware for ZKP generation tasks.

11.9.3 Economic Model and Task Distribution

The marketplace implements a structured economic model for prover participation through Substrate's economic framework:

Staking Requirements Provers stake ZKP and DTK tokens to join the network according to a tiered model managed through Substrate's staking pallet:

- **Entry Tier:** Minimum stake requirement (to be determined during economic simulations in testnet) allowing basic participation
- **Standard Tier:** Intermediate stake level granting priority in task assignment
- **Premium Tier:** Higher stake level enabling access to more complex proof generation tasks

Slashing Conditions Stake may be partially reduced under specific circumstances through Substrate's slashing mechanisms:

- Failure to submit proofs within the defined deadline (minor reduction)
- Submission of invalid proofs (moderate reduction)
- Provable malicious behavior such as collusion attacks (major reduction)

Reward Calculation Compensation follows a base fee plus performance model managed through Substrate's reward distribution:

- Base component for successful proof generation
- Variable component based on proof complexity and verification cost
- Bonus multiplier for consistent high-quality service

This structured economic model will be calibrated during testnet operation, with parameters adjusted based on observed participant behavior and system performance.

11.9.4 Integration with Marketplace Operations

The prover network integrates seamlessly with core marketplace functions:

Access Control Provers generate the ZKPs needed for tiered access verification, enabling privacy-preserving permission checks. When a user requests access to a dataset, the system routes the request to an available prover through off-chain workers, which generates a proof of eligibility that can be verified on-chain through Substrate's verification infrastructure without revealing the user's token balance or other sensitive information.

Dataset Validation Specialized hardware assists in validating dataset attributes during the registration process. Provers can generate proofs confirming properties like size, format, or statistical characteristics without exposing the underlying data, enhancing privacy during the dataset onboarding process.

Governance Participation Provers support privacy-preserving voting mechanisms for Data DAOs through Substrate's democracy pallet, allowing token holders to participate in governance decisions without revealing their voting choices or token holdings. This maintains the integrity of the governance process while preserving individual privacy.

Bulk Operation For operations involving multiple proofs (such as batch dataset registration or access requests), provers can optimize their processes to handle related tasks efficiently, improving throughput and reducing latency.

11.9.5 Operational Considerations and Future Development

Deploying specialized hardware introduces several operational considerations that the marketplace architecture addresses:

Hardware Lifecycle Management Protocols for device onboarding, updates, and decommissioning ensure the prover network maintains security and performance standards throughout the hardware lifecycle, coordinated through Substrate's upgrade mechanisms.

Redundancy and Reliability The network is designed with sufficient redundancy to ensure proof generation capacity remains available even if individual devices go offline, maintaining marketplace functionality during fluctuations in prover availability.

Scaling Strategy As marketplace adoption grows, the prover network can scale horizontally by adding more devices, ensuring capacity keeps pace with demand. The economic model automatically adapts to incentivize additional prover participation as needed.

This dedicated hardware layer represents a critical infrastructure component for the Data Marketplace, enabling efficient privacy-preserving operations at scale while maintaining alignment with the base layer's technical parameters and security model. By combining specialized hardware with economic incentives, the marketplace creates a sustainable foundation for privacy-preserving data exchange.

11.10 Decentralized Governance via Data DAOs: Dataset Submission and Approval Process

Data Decentralized Autonomous Organizations (DAOs) oversee dataset submissions in the Data Marketplace, establishing a community-driven governance framework to ensure datasets meet quality and ethical standards before listing. This section outlines the process for submission, verification, voting, and approval, designed to leverage the collective decision-making of DTK token holders through Substrate's governance infrastructure.

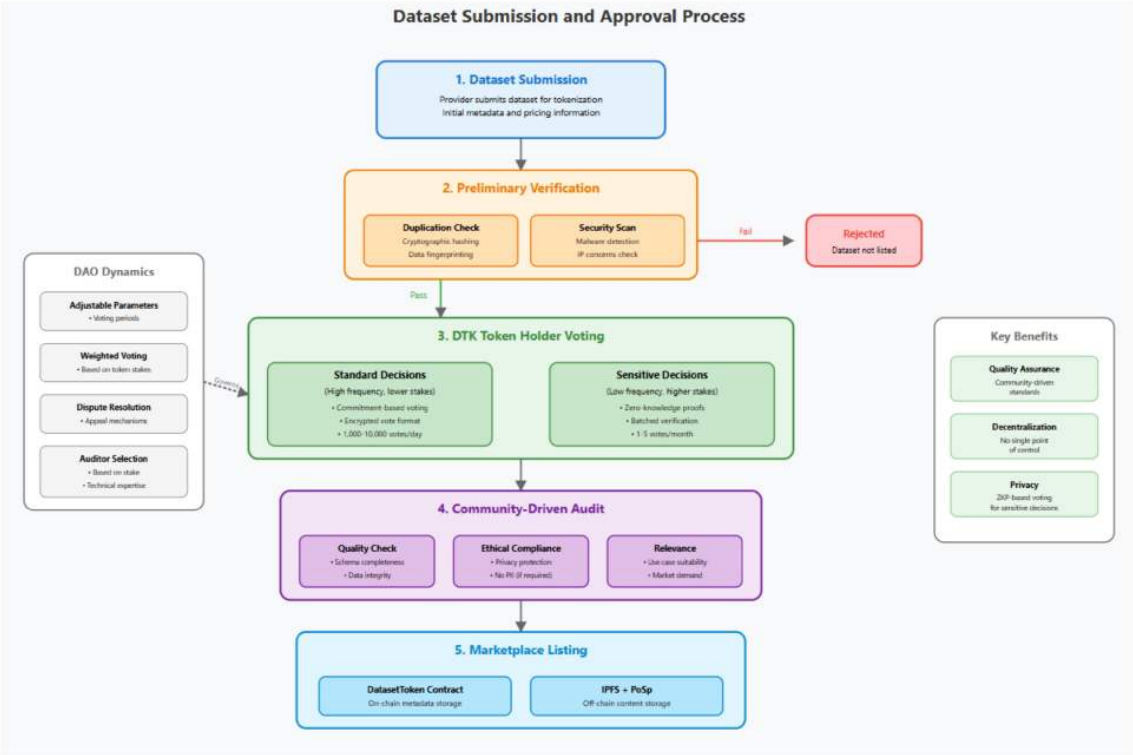

Figure 15: Dataset Submission and Approval Process

11.10.1 Dataset Submission and Preliminary Verification

When providers submit datasets for tokenization, they undergo a preliminary verification process coordinated through custom Substrate pallets to assess quality and compliance. This initial screening aims to identify potential issues such as duplication, malware, or intellectual property (IP) concerns.

The verification process incorporates data fingerprinting (e.g., cryptographic hashing) to detect duplication, automated analysis for malware identification, and metadata comparison for IP concerns. Datasets that pass this initial verification proceed to the community voting stage with a suggested listing price based on factors like dataset size and expected demand.

This preliminary verification stage balances automated checks with human oversight, ensuring efficient processing while maintaining appropriate standards. It serves as a first line of defense against problematic submissions, reducing the governance burden on the DAO.

11.10.2 Voting by DTK Token Holders

Datasets that pass preliminary verification are submitted for voting by DTK token holders within the Data DAO through Substrate's democracy pallet.

The voting process employs a multi-tiered approach that pragmatically balances privacy, scalability, and security:

- **For Standard Decisions (high frequency, lower stakes):** A commitment-based voting scheme managed through Substrate's democracy pallet allows token holders to commit their votes in an encrypted format, with results revealed after the voting period concludes. This approach provides basic privacy while maintaining computational efficiency. The system can process approximately 1,000-10,000 votes per day within reasonable weight constraints.
- **For Sensitive Decisions (low frequency, higher stakes):** Zero-knowledge proofs provide stronger privacy guarantees, where each voter proves valid voting rights without revealing their identity or specific vote. Due to the computational intensity of ZK verification, this tier is reserved for critical governance decisions (approximately 1-5 per month) and employs:
  - Batched proof verification (amortizing costs across multiple votes)
  - Delegation mechanisms (reducing the total proof count)
  - Quorum-based sampling (where a statistically significant subset of token holders participate)

This pragmatic approach acknowledges the inherent scalability constraints of on-chain governance while preserving privacy for the most sensitive decisions.

11.10.3 Community-Driven Audit

Following a successful vote, the dataset undergoes a community-driven audit by selected DAO members. This audit evaluates the dataset for quality, ethical compliance, and relevance to ensure it meets the marketplace's standards.

The selection of auditors is based on factors like staking activity and expertise, creating a balanced review process. The audit examines dataset characteristics such as completeness of schema, absence of personally identifiable information (when required), and suitability for intended applications.

This audit phase maintains high standards while preserving the decentralized nature of the marketplace. Future refinements may include defining specific audit timelines and establishing appeal processes for rejected datasets.

11.10.4 Marketplace Listing

Approved datasets are listed on the marketplace, making them available for purchase and access. The DatasetToken contract or native Substrate pallet records the dataset's metadata in Patricia Tries, while its content is stored off-chain via IPFS, secured by PoSp mechanisms to ensure availability.

The listing integrates with the tiered access system, allowing providers to set access levels from free metadata previews to full access. This granular control enables providers to maximize value while maintaining accessibility for different user needs.

11.10.5 DAO Dynamics

The Data DAO's governance evolves through iterative refinement, incorporating features such as adjustable voting periods, weighted voting based on coin stakes, and dispute resolution mechanisms through Substrate's governance framework. These dynamics balance fairness, efficiency, and quality, ensuring the marketplace remains a trusted platform for data exchange.

The DAO structure enables responsive governance that can adapt to changing marketplace needs and community feedback, creating a system that grows more robust over time. Through this decentralized approach, the Data Marketplace fosters an ecosystem where community standards and individual incentives align to create sustainable value.

11.11 Revenue Models: Monetizing Data in the Marketplace

This section outlines revenue models for data owners in the Data Marketplace, describing frameworks for monetizing datasets through flexible income structures and exploring how a portion of the revenue supports the ecosystem's sustainability.

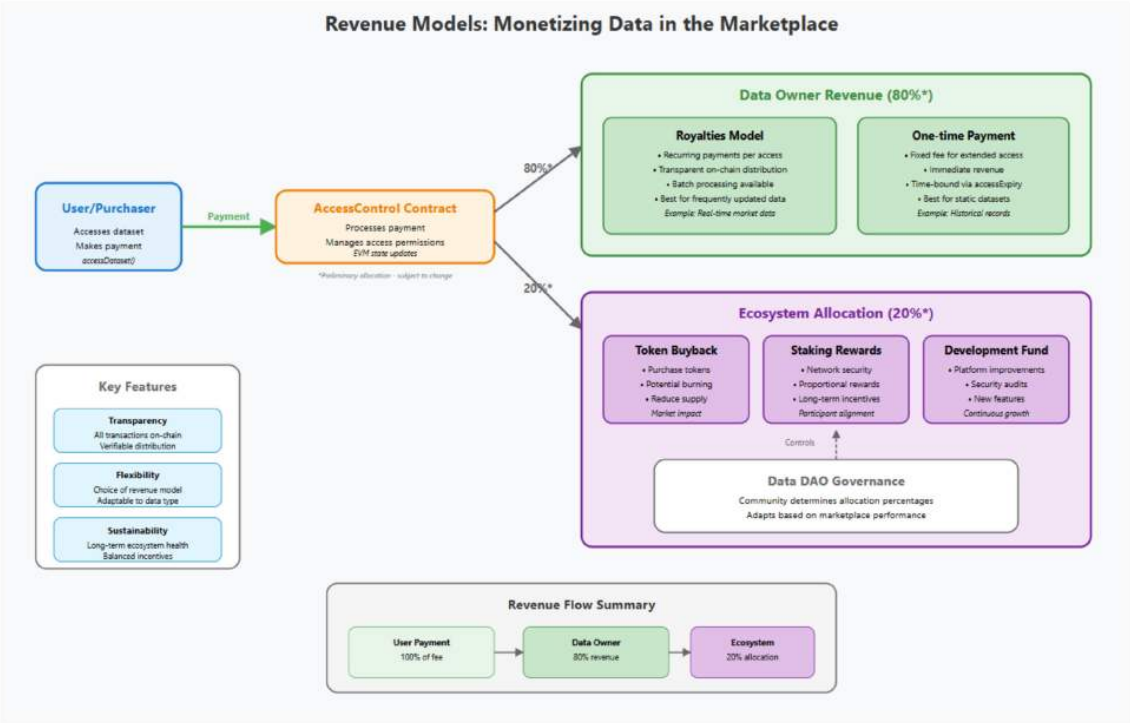

Figure 16: Revenue

11.11.1 Income Models for Data Owners

Data owners can monetize datasets via two principal models, offering flexibility while ensuring fair compensation:

Royalties This model enables recurring payments each time a user accesses a dataset. When users call the `accessDataset` function in the `AccessControl` contract or native pallet, a portion of the fee (e.g., 80% of the total) is distributed to the owner. This distribution occurs on-chain using Substrate's state management mechanisms, with each transaction recorded transparently in Patricia Tries. For datasets with high access frequency, batch processing of royalty payments can be implemented to reduce weight costs while maintaining regular compensation to data providers.

One-time Payments Alternatively, owners can opt for a fixed fee in exchange for extended access to their dataset. This approach simplifies the compensation structure and provides immediate revenue to the data owner. The access duration is managed through the `accessExpiry` parameter, ensuring that permissions remain time-bound according to the agreement.

Both models can be adapted to different data types and usage patterns. For example, reference datasets that require frequent updates might benefit from recurring royalties, while static datasets like historical records might be better suited to one-time payment structures.

11.11.2 Economic Model for Revenue Distribution

To foster ecosystem sustainability, a portion of marketplace fees (e.g., 20% of transaction fees) is allocated to support broader platform needs through Substrate's treasury mechanism. This allocation will be distributed through several mechanisms:

Coin Buyback A percentage of ecosystem fees will be used to purchase and potentially burn coins from the open market, potentially reducing supply over time.

Staking Rewards Participants who stake tokens to support network security through Substrate's staking system receive proportional rewards from the ecosystem allocation, aligning incentives for long-term participation.

Development Funding A portion of ecosystem fees is directed toward ongoing platform improvements, security audits, and new feature development through Substrate's treasury, ensuring the marketplace evolves to meet user needs.

Governance processes implemented through Data DAOs using Substrate's democracy pallet determine the specific allocation percentages for these mechanisms, allowing the community to adapt the economic model based on marketplace performance and requirements.

The revenue models and distribution framework aim to create balanced economic incentives that reward data providers while supporting the broader marketplace ecosystem, establishing a sustainable foundation for decentralized data exchange.

11.12 Monitoring System: Performance and Activity

This section outlines a proposed framework for monitoring system performance and activity in the Data Marketplace, aiming to potentially track operational integrity, user engagement, and data reliability. As a testnet-phase concept, this monitoring system—including methodologies, metrics, and implementation details—is under development. All figures and features are illustrative, subject to refinement based on testing and feedback.

11.12.1 Monitoring Framework

On-Chain Monitoring for Blockchain Activities The Data Marketplace monitors on-chain activities like dataset access and prover performance via smart contract event logs and Substrate's native event system (e.g., `AccessExtended`, `ProofRewardDistributed`). These events log crucial data such as dataset IDs and timestamps. Off-chain indexing tools coordinate through Substrate's off-chain workers to aggregate this into KPIs including throughput (aligned with the base layer's 100-500 TPS capacity), weight usage, and token volume. This monitoring framework ensures the marketplace operates efficiently within the constraints and capabilities of the underlying blockchain.

For example, an `AccessExtended` event might log a dataset ID and timestamp, tracking 1,000 accesses/hour for a popular dataset. Testnet will assess scalability with 1,000 users, targeting <1-second updates.

Off-Chain Monitoring for Infrastructure Health Off-chain monitoring focuses on IPFS availability (target: 99.9% uptime, consistent with the base layer specifications) and prover performance metrics that align with the ZKP Blockchain's performance parameters, coordinated through Substrate's off-chain workers [87] [120]. IPFS probes could measure retrieval latency (<1 second for 1 GB, illustrative) across 5 nodes/dataset, flagging downtime. APIs (e.g., REST) might track network health, with testnet validating 100 nodes for >99% coverage.

Privacy-Preserving Monitoring Monitoring aims to potentially ensure confidentiality by logging anonymized data (e.g., dataset IDs), avoiding user identities. Future zero-knowledge proofs verified through Substrate's verification infrastructure might enhance privacy [77], with testnet targeting <1% linkage risk (illustrative) for 100 users.

Practical Implications The system seeks to potentially offer insights, possibly informing tier adjustments (e.g., 1,000 accesses/hour) and workload optimization (>90% prover completion, illustrative), with testnet refining privacy and scalability.

11.13 Federated Learning in the Data Marketplace

This section explores how the Data Marketplace infrastructure could support federated learning (FL), a privacy-enhancing approach to collaborative model training.

11.13.1 Federated Learning Framework

The Data Marketplace architecture supports federated learning (FL) implementations, allowing participants to train machine learning models collaboratively while keeping raw data local, aligning with privacy preservation principles [93] [94]. The process follows a defined workflow coordinated through Substrate's off-chain workers:

- **Dataset Selection:** Participants access datasets via the tiered system, selecting relevant data (e.g., medical records) at Tier 3, with zk-SNARKs verifying eligibility through Substrate's verification infrastructure without revealing token balances [77] [121].
- **Local Training:** Participants train local models using a stochastic gradient descent (SGD) algorithm with a learning rate of 0.01 and batch size of 32, processing data on their nodes to generate gradient updates.
- **Secure Update Submission:** Updates are encrypted with AES-256 (GCM mode) and submitted with zk-SNARK proofs of correctness, costing approximately 210,000 gas equivalent weight in testnet simulations [82].
- **Aggregation:** An off-chain aggregator coordinated through Substrate's off-chain workers (using secure multi-party computation, SMPC) computes a weighted average of updates, weighted by dataset size, and submits a single proof for on-chain verification.

11.13.2 Integration with ZKP Infrastructure

The envisioned federated learning implementation leverages several components of the ZKP Blockchain's base layer:

Privacy Preservation The system's zero-knowledge capabilities verified through Substrate's verification infrastructure could provide additional privacy assurances for model updates, addressing potential inference attacks [77] [93].

Data Access The marketplace's tiered access control framework implemented through custom Substrate pallets provides the foundation for selective dataset utilization in training scenarios [77].

Computational Verification The base layer's Proof of Intelligence (PoI) mechanisms for specific mathematical operations integrated with BABE+GRANDPA consensus could help verify certain aspects of the training process [79].

11.13.3 Implementation Considerations

Implementing federated learning within the marketplace presents several technical challenges that would need to be addressed during development:

Off-chain Aggregation Model updates would need to be aggregated off-chain through Substrate's off-chain workers to avoid prohibitive on-chain costs, with only verification proofs posted to the blockchain. Non-independent and identically distributed (non-IID) data across participants creates convergence challenges, requiring techniques like adaptive optimization or knowledge distillation.

Security Considerations The implementation will need to mitigate model poisoning attacks (where malicious participants submit harmful updates), gradient leakage (where training updates reveal information about private data), and Sybil attacks (where entities create multiple identities to gain influence).

Performance Optimization Bandwidth and computational constraints would require careful optimization to ensure practical usability.

The federated learning framework represents a natural extension of the marketplace's privacy-preserving architecture, demonstrating how the underlying ZKP infrastructure could support advanced machine learning capabilities while maintaining strong privacy guarantees.

11.14 Security & Privacy

This section examines the security and privacy foundations of the Data Marketplace, focusing on established cryptographic mechanisms and their implementation in the decentralized data economy.

11.14.1 Cryptographic Security Foundations

The Data Marketplace builds directly upon the ZKP Blockchain's cryptographic infrastructure, leveraging zk-SNARKs for privacy-preserving operations through both EVM pallet and native Substrate verification mechanisms [77] [121]. These zero-knowledge proofs enable critical marketplace functions like access control verification and dataset attribute validation without exposing sensitive information.

The marketplace's implementation aligns with the base layer's security parameters, including the BLS12-381 elliptic curve for zk-SNARKs, which targets 128-bit security against classical attacks based on the discrete logarithm problem [88] [98]. However, zk-SNARKs require a trusted setup, where a Common Reference String (CRS) is generated through a multi-party computation (MPC) ceremony involving 20 participants, achieving a collusion risk below 2^{-128} [89].

To mitigate risks associated with the trusted setup, we have implemented a transparent audit trail of the MPC ceremony, publicly logging participant contributions and destruction proofs in Substrate's immutable storage for perpetual verification. Additionally, we are pursuing a phased migration to zk-STARKs for high-sensitivity operations (e.g., governance voting, compliance verification), which eliminate the need for a trusted setup while offering post-quantum security, albeit with larger proof sizes.

This migration will begin with a pilot integration in the next testnet phase, targeting full deployment for sensitive functions within 18 months, ensuring a trustless architecture consistent with blockchain principles. Long-term security planning includes ongoing evaluation of emerging cryptographic threats and potential transitions to post-quantum secure alternatives as needed.

11.14.2 Threat Model and Key Protections

The marketplace operates under a defined threat model addressing four key security concerns:

- **Data Integrity**: PoSp mechanisms implemented through custom Substrate pallets ensure dataset availability and tamper resistance through cryptographic storage proofs, with 99.9% uptime as established in the base layer specifications [87] [99].
- **Access Control**: zk-SNARKs provide cryptographic verification of access rights through Substrate's verification infrastructure without revealing user balances or permissions, maintaining privacy while ensuring security [77].
- **Smart Contract Security**: Contract-level protections follow established best practices for EVM pallet environments, including access controls and event logging through Substrate's event system for transparency [80] [85].
- **Privacy Guarantees**: Beyond basic encryption, the system's zero-knowledge architecture ensures that sensitive information remains protected during verification processes, addressing the fundamental privacy challenges of decentralized data sharing [77].

The security framework integrates with the marketplace's governance model through Substrate's democracy pallet, allowing Data DAOs to establish and enforce quality standards while maintaining the system's cryptographic guarantees. This multi-layered approach creates a balanced security architecture that protects both individual participants and the ecosystem as a whole.

11.15 Scalability & Optimization

This section examines key strategies for addressing the scalability challenges of operating a decentralized data economy on the ZKP Blockchain's Layer 1 Substrate infrastructure.

11.15.1 Scalability Challenges and Strategies

The Data Marketplace faces inherent Layer 1 blockchain constraints including transaction throughput limitations and weight costs. While the ZKP Blockchain achieves 100-500 TPS in real-world conditions as specified in the base layer, optimization remains essential for economic viability [81] [96].

On-chain optimization focuses on smart contract efficiency in the `AccessControl` and `DatasetToken` components, leveraging EVM pallet precompiles for ZKP verification. These precompiles enable the 200,000 gas equivalent weight verification cost established in the base layer specifications [82] [97]. Additionally, batch processing for operations like bulk dataset registration reduces per-transaction overhead through Substrate's batch utilities, though with potential latency trade-offs that require testnet validation.

Off-chain scalability leverages IPFS for dataset storage coordinated through Substrate's off-chain workers with a replication factor of 5 and PoSp verification to ensure 99.9% availability [87] [120]. This hybrid approach stores only metadata and verification proofs in Patricia Tries while keeping data off-chain, dramatically improving cost efficiency for large datasets.

11.15.2 Optimization Techniques

Proof Generation and Verification The marketplace aligns with the base layer's parameters, with proof generation for 10,000-gate circuits requiring approximately 10 seconds on standard hardware. On-chain verification costs 200,000 gas equivalent weight using the EVM pallet's optimized precompiles and native Substrate verification for elliptic curve operations [82] [97] [121].

Smart Contract Efficiency Weight optimization in core contracts focuses on minimizing `SSTORE` operations, which cost ∼20,000 gas equivalent weight each [80]. For high-volume scenarios, Substrate's state channel implementations could significantly reduce on-chain transactions, though with increased implementation complexity.

The scalability strategy balances on-chain efficiency with off-chain distribution through Substrate's modular architecture, creating a system capable of handling privacy-preserving data transactions at scale while maintaining economic viability and security within the constraints of Layer 1 performance.

11.16 Key Innovations

The Data Marketplace builds upon the ZKP Blockchain infrastructure with three core innovations that address key challenges in decentralized data economies.

11.16.1 Cryptographically Secured Data Tokenization

The marketplace implements a robust tokenization framework that converts datasets into blockchain-native assets secured by zero-knowledge proofs. By representing data as Substrate native tokens or unique assets with cryptographic verification of attributes, the system creates a foundation for secure trading without compromising privacy [83] [123]. This approach leverages the base layer's ZKP verification capabilities (200,000 gas equivalent weight per verification as specified) through Substrate's verification infrastructure to enable trustless transactions in sensitive domains like healthcare and finance [77].

11.16.2 Tiered Access Control with Privacy Preservation

Building on the base layer's ZKP implementation, the marketplace introduces a privacy-preserving access control system implemented through custom Substrate pallets that enables granular data sharing while protecting user information. The tiered model, ranging from public metadata access to complete dataset utilization, allows data owners to maintain sovereignty over their assets while monetizing them appropriately [77]. Access verification utilizes the ZKP Blockchain's established cryptographic primitives through Substrate's verification infrastructure, ensuring technical consistency with the base layer's security model.

11.16.3 DAO-Governed Quality Assurance

The marketplace incorporates a decentralized governance mechanism for dataset quality control through Substrate's democracy pallet, leveraging coin-based voting with privacy preservation through ZKPs [84] [124]. This approach distributes responsibility for marketplace curation to stakeholders, creating a self-regulating ecosystem that can adapt to evolving standards and requirements. The governance model aligns with the base layer's hybrid consensus approach, creating a coherent decentralized system from infrastructure to application level.

These innovations demonstrate how the ZKP Blockchain's core capabilities—zero-knowledge proofs, hybrid consensus, and decentralized governance—can be applied to create a practical solution for privacy-preserving data exchange.

11.17 Conclusion

The ZKP Data Marketplace represents a practical application of the ZKP Blockchain's Layer 1 Substrate infrastructure to address the limitations of centralized data systems. By integrating zero-knowledge proofs with blockchain technology, the marketplace provides a framework for privacy-preserving data exchange while enabling verifiable computation and fair value distribution.

Key components of this framework include secure tokenization of datasets through Substrate's native token systems, tiered access controls implemented through custom pallets that protect user privacy, and community governance through Data DAOs using Substrate's democracy pallet.

The marketplace leverages the base layer's zk-SNARKs implementation for 200,000 gas equivalent weight verification cost through Substrate's verification infrastructure, allowing efficient privacy-preserving operations at scale. Off-chain storage via IPFS coordinated through Substrate's off-chain workers with 99.9% availability ensures data persistence without blockchain bloat.

The hybrid consensus model combining Proof of Intelligence (PoI) and Proof of Space (PoSp) integrated with BABE+GRANDPA provides the foundation for a balanced ecosystem where computational validation and storage availability complement each other in addition to a dedicated prover network for computation of proof generation.

This approach supports complex use cases from healthcare research to financial analytics while maintaining privacy and security through Substrate's modular architecture.

Throughout the marketplace's development, continuous refinement through testnet validation will enhance the system's performance, security, and economic sustainability. The ZKP Data Marketplace demonstrates how zero-knowledge cryptography, when applied to practical challenges in data management, can create new opportunities for secure and equitable data exchange.

12 The ZKP Coin and DTK Token: The Foundation for Privacy-Preserving Computation

The ZKP ecosystem employs a dual-token/coin architecture featuring the **ZKP** coin and the **DTK** token, each serving distinct yet complementary roles in enabling a privacy-preserving, decentralized data economy. The ZKP coin anchors the blockchain's consensus and security through Substrate's native mechanisms, while the DTK token powers the ZKP Data Marketplace's transactional and verification processes.

12.1 ZKP Coin: Securing the Blockchain through Hybrid Consensus

The ZKP coin is the native cryptocurrency of the ZKP blockchain built on Substrate, maintaining network security, incentivizing validator participation, and facilitating decentralized governance through native pallets.

12.1.1 Utility of the ZKP Coin

The ZKP coin fulfills five critical functions within Substrate's architecture:

- **Privacy-Preserving Operations**: Validators are rewarded with ZKP coins for generating and verifying zk-SNARKs through Substrate's verification infrastructure (e.g., verifying AI inference tasks), with proof verification times of $\sim$2 ms and proof sizes of $\sim$288 bytes.
- **Decentralized Governance**: Coin holders participate in quadratic voting via Data DAOs using Substrate's democracy pallet. Voting power scales as $\sqrt{\text{coins}}$ (e.g., 100 coins = 10 votes), maintaining a Gini coefficient < 0.3 for fairness.
- **Validator Rewards**: Validators earn rewards through Proof of Intelligence (PoI) and Proof of Space (PoSp) integrated with BABE+GRANDPA consensus, aligning network utility with security.
- **Network Security via Staking**: Validators stake coins through Substrate's staking pallet to participate in consensus, with slashing penalties managed by Substrate's slashing mechanisms for invalid proofs.
- **Ecosystem Growth**: 20% of transaction fees go to Substrate's treasury for buybacks and development funding.

12.1.2 Consensus Mechanisms

The ZKP blockchain employs a hybrid consensus model integrating PoI, PoSp, and staking within Substrate's BABE+GRANDPA framework.

Proof of Intelligence (PoI): PoI rewards validators for performing verifiable AI computations through custom Substrate pallets.

$$\text{PoI\_Score}_i = \sum_{j=1}^{n} (\text{Accuracy}_j \cdot \text{Efficiency}_j \cdot \text{Complexity}_j) \tag{1}$$

Proof of Space (PoSp): PoSp rewards storage contributions, verified via Merkle proofs through custom Substrate pallets:

$$\text{PoSp\_Score}_i = \text{Storage}_i \cdot \text{Uptime}_i \tag{2}$$

Staking Integration: Staking power is calculated through Substrate's staking mechanisms as:

$$W_i = \alpha(t) \cdot \text{PoI\_Score}_i + \beta(t) \cdot \text{PoSp\_Score}_i + \gamma(t) \cdot \text{Stake}_i \tag{3}$$

with dynamic coefficients $\alpha(t), \beta(t), \gamma(t)$ managed through governance pallets, initially set as $\alpha \approx 0.3, \beta \approx 0.3, \gamma \approx 0.4$.

Reward Distribution: Rewards are distributed through Substrate's native reward mechanism:

$$R_i = R_{\text{block}} \cdot \frac{W_i}{\sum W_j} \tag{4}$$

12.2 DTK Token: Facilitating the Data Marketplace

The DTK token is the utility token of the ZKP Data Marketplace, implemented through Substrate's assets pallet, used for dataset transactions, ZKP verification, and governance at the application layer.

12.2.1 Utility of the DTK Token

- **Dataset Transactions**: DTK enables buying/selling of datasets through smart contracts and native pallets, with a x% transaction fee (y% to Substrate's treasury, z% to provers).
- **ZKP Verification Compensation**: Provers earn DTK for verifying dataset attributes through Substrate's verification infrastructure. Example: 1 DTK per GB as a baseline, adjustable by governance through the democracy pallet.
- **Marketplace Governance**: Stake-weighted voting through Substrate's democracy pallet (capped at 5% influence) enables adjustments to marketplace parameters.

12.2.2 Economic Model

- **Supply and Distribution**: Dynamically minted through Substrate's assets pallet based on marketplace activity; demand-driven economics.
- **Fee Structure**: x% per transaction, with revenues split between Substrate's treasury and prover rewards through native distribution mechanisms.
- **Governance Adjustments**: DTK holders vote on prover rates, fees, and standards through Substrate's democracy pallet. Target rates: 0.5–2 DTK/GB.

12.3 Coin Economics and Sustainability

The tokenomics design ensures long-term ecosystem sustainability through Substrate's economic framework:

12.3.1 Inflation and Deflation Mechanisms

- **ZKP Coin**: Controlled inflation through block rewards, balanced by treasury buybacks
- **DTK Token**: Market-driven supply based on marketplace activity and demand

12.3.2 Value Accrual

Both coins and tokens derive value from:

- Network security and consensus participation (ZKP)
- Data marketplace utility and governance rights (DTK)
- Cross-ecosystem interoperability through Substrate's infrastructure

12.3.3 Governance Evolution

Coin holders can propose and vote on:

- Consensus parameter adjustments
- Fee structure modifications
- New feature implementations
- Cross-chain integration strategies

This dual-coin and token architecture, built on Substrate's robust foundation, creates a sustainable economic model that aligns incentives across all ecosystem participants while maintaining the flexibility to evolve with changing market conditions and technological advances.

13 ZKP Project Technical Roadmap (2022–2030)

The ZKP project began its research and initial development in early 2022, with a team of cryptographers and blockchain developers exploring the theoretical foundations of combining zero-knowledge proofs with decentralized AI systems. This roadmap outlines our technical development trajectory from the public pre-mainnet phase through the post-mainnet phase till 2030, built on Substrate's modular blockchain framework.

13.1 Public Pre-Mainnet Phase

H1 2025

Technical Focus: Foundation & Initial Implementation

Architecture & Design:

- Establish technical specifications for the hybrid consensus model integrated with BABE+GRANDPA
- Develop architecture design documentation for Substrate-based distributed system
- Create technical specifications for prover device integration with off-chain workers
- Design custom pallet architecture for PoI and PoSp integration

ZKP Development:

- Develop initial specialized circuits for basic matrix operations (up to 50×50)
- Design ZKP circuits for simple AI inference tasks
- Complete initial ZKP circuit designs for basic operations
- Initiate zk-SNARKs for on-chain verification through both EVM pallet and native Substrate verification targeting 200,000 gas equivalent weight cost benchmark

Infrastructure:

- Finalize hardware requirements and software stack for provers integrated with Substrate's off-chain workers
- Develop prover specifications including computational requirements within Substrate's runtime
- Create initial developer documentation for pallet APIs and integration guides
- Publish comprehensive technical whitepaper detailing cryptographic foundations and Substrate-based security models

Milestones:

- Initial ZKP circuit designs completed for basic operations
- Technical whitepaper published

H2 2025

Technical Focus: Closed Testnet Development & Core Features

Closed Network Development:

- Deploy closed alpha testnet using Substrate framework with initial 20–30 validator nodes focusing on BABE+GRANDPA consensus stability
- Enhance closed alpha environment to support 50 nodes with 75% uptime using Substrate's networking layer
- Improve testing infrastructure and expand core pallet feature set
- Create simulation environment for PoI/PoSp modeling with limited scale (10–50 nodes) using Substrate's benchmarking tools

Blockchain Integration:

- Implement EVM pallet with Frontier compatibility enabling Solidity smart contract deployment
- Optimize weight consumption for smart contract operations within Substrate's fee system
- Establish IPFS integration through off-chain workers for decentralized storage framework with basic access controls
- Develop initial marketplace contracts and native pallets with access control mechanisms

Consensus & Security:

- Begin Proof of Intelligence (PoI) pallet implementation developing initial algorithms for matrix operations validation
- Initiate first external security assessment of core pallets and runtime logic
- Establish research collaboration with university cryptography departments for ZKP research
- Partner with 1–2 cryptography research groups for ZKP optimization within Substrate's architecture

Milestones:

- Enhanced alpha testnet operational with 50 nodes and EVM pallet compatibility
- Initial PoI pallet algorithms developed and tested
- IPFS integration functional through off-chain workers for decentralized storage
- Simulation environment demonstrating theoretical consensus with 50 nodes
- Security audit phase 1 initiated for Substrate runtime

H1 2026

Technical Focus: Advanced Closed Testing & Dual Consensus Implementation

Enhanced Closed Testing:

- Expand closed testnet to support 100–200 internal testers and partners using Substrate's telemetry
- Enable comprehensive functionality testing through native pallets and EVM pallet
- Conduct extensive stress testing with simulated high-load scenarios
- Refine user experience based on internal feedback and Substrate best practices

Consensus Implementation:

- Deploy comprehensive PoI pallet mechanisms for matrix operations validation in controlled environment
- Implement Proof of Space pallet integrated with IPFS through off-chain workers for data availability
- Achieve stable dual consensus functionality integrated with BABE+GRANDPA
- Develop comprehensive security testing framework using Substrate's formal verification tools

Advanced Features Development:

- Create comprehensive federated learning framework through off-chain workers enabling privacy-preserving machine learning
- Expand ZKP circuits to support neural network layers with up to 10^4 parameters
- Develop advanced marketplace features including governance mechanisms
- Continue academic research collaboration with multiple cryptography institutions

Milestones:

- Expanded closed testnet operational with 200 internal users
- Dual consensus (PoI and PoSp pallets) fully functional with BABE+GRANDPA
- Initial federated learning framework completed using off-chain workers
- Advanced ZKP circuits supporting 10^4 parameter neural networks
- Second comprehensive security audit completed for Substrate runtime

H2 2026-H1 2027

Technical Focus: Public Testnet Launch Preparation, Optimization & Community Building

Pre-Public Launch Preparation:

- Complete comprehensive internal testing with 500+ closed participants
- Finalize user interface and developer tooling for public release
- Establish community support infrastructure and documentation
- Conduct final security reviews and performance optimization

Infrastructure Hardening:

- Implement robust monitoring and alerting systems using Substrate's telemetry
- Deploy scalable infrastructure to support anticipated public testnet load
- Establish incident response procedures and support systems
- Complete comprehensive load testing and performance benchmarking

Feature Completion:

- Finalize data marketplace beta features for public testing
- Complete governance mechanisms using democracy pallet
- Implement comprehensive developer APIs and SDK components
- Deploy educational resources and community engagement tools

Public Testnet Launch:

- Launch public testnet alpha supporting 5,000 initial testers using Substrate's telemetry
- Enable comprehensive testing capabilities through native pallets and marketplace features
- Upgrade to public testnet beta with enhanced stability targeting 15,000 users
- Scale to full public testnet supporting 30,000+ users

Community Engagement:

- Deploy data marketplace with full tokenization capabilities through assets pallet
- Implement tiered access controls and governance features for community testing
- Host regular developer workshops and community events
- Establish bug bounty programs and security incentives

Performance Validation:

- Achieve 100–200 TPS with 95% uptime on public testnet
- Validate dual consensus performance under real-world conditions
- Demonstrate marketplace functionality with real dataset transactions
- Complete final security audits based on public testnet feedback

Mainnet Preparation:

- Finalize all critical features and security implementations

- Complete comprehensive documentation and developer resources
- Establish validator onboarding processes and economic parameters
- Prepare governance transition to community control

Milestones:

- Internal testing completed with 500+ participants
- Infrastructure hardened and ready for public scale
- All core features completed and thoroughly tested
- Community support systems established
- Public testnet alpha launched with 5,000 users
- Public testnet beta operational with 15,000 users
- Full public testnet supporting 30,000+ users
- Data marketplace fully functional with community governance
- Complete mainnet preparation

H1 2027-H1 2028

Technical Focus: Mainnet Launch & Initial Operations

Mainnet Launch:

- Execute mainnet launch with complete functionality
- Achieve initial target of 150–200 TPS with 99% uptime on production network
- Deploy full data marketplace with tokenization and governance capabilities
- Enable cross-chain compatibility through XCM integration preparation

Post-Launch Optimization:

- Monitor network performance and implement optimizations
- Scale validator network and establish stable operations
- Launch community governance and DAO frameworks
- Begin planning for parachain integration within Polkadot ecosystem

Milestones:

- Mainnet successfully launched
- Initial performance targets achieved (150–200 TPS, 99% uptime)
- Data marketplace fully operational with community governance
- Stable validator network and community governance established

13.2 Post-Mainnet Phase

2028 H2: Adoption & Parachain Integration

Technical Focus: Performance Scaling & Enhanced AI

Performance Optimization:

- Scale to 300–400 TPS through runtime optimization and pallet efficiency improvements
- Further optimize to achieve 500–700 TPS with sustained stability
- Implement parachain integration within Polkadot ecosystem for horizontal scaling
- Deploy specialized AI computation parachains for advanced workloads

Advanced Features:

- Expand ZKP capabilities to support convolutional neural networks
- Enable complex model training and inference supporting 10^5 parameters
- Support advanced AI operations including multi-layer neural networks
- Release comprehensive SDKs for multiple programming languages interfacing with Substrate
- Implement additional privacy-preserving computation techniques through custom pallets

Cross-Chain Development:

- Implement XCM (Cross-Consensus Message Passing) for seamless parachain communication
- Deploy cross-chain data marketplace functionality
- Enable multi-chain governance participation through XCM

Milestones:

- 300–400 TPS consistently achieved through runtime optimization
- 500–700 TPS with sustained stability and parachain integration
- Cross-chain functionality deployed through XCM
- Advanced AI capabilities supporting 10^5 parameter models

2030: Enterprise & Quantum-Resistant Security

Technical Focus: Advanced Features & Future-Proofing

Scalability Achievements:

- Scale to 800–1,000 TPS through advanced parachain orchestration
- Implement specialized computation parachains achieving 3,000 TPS for AI workloads
- Optimize consensus mechanism for enhanced throughput while maintaining decentralization
- Deploy recursive proof systems for enhanced verification efficiency

Security & Privacy Enhancements:

- Implement post-quantum cryptographic elements through runtime upgrades
- Complete transition to quantum-resistant security elements
- Deploy fully homomorphic encryption for select operations through custom pallets
- Enhance AI capabilities supporting multi-modal models and advanced neural architectures

Enterprise Features:

- Develop specialized management interfaces for enterprise console
- Deploy advanced privacy features across the platform
- Develop compliance tools and audit frameworks for enterprise requirements
- Implement governance analytics and automated policy enforcement

Milestones:

- 800–1,000 TPS with quantum-resistant infrastructure
- 3,000 TPS operational through specialized parachains
- Enterprise-grade compliance and governance tools deployed
- Full post-quantum cryptographic transition completed

14 Future Research

The ZKP Ecosystem is dedicated to pushing the boundaries of decentralized AI through a rigorous research agenda, ensuring that privacy, scalability, and security remain at the core of our platform built on Substrate's modular architecture. We are addressing critical challenges in privacy-preserving computation and decentralized data sharing with a forward-looking approach. The following areas represent our strategic focus for future development, each designed to mitigate risks, enhance performance, and uphold the ecosystem's integrity.

14.1 Masking

We aim to develop efficient, application-specific masking techniques that integrate with homomorphic encryption and differential privacy to provide robust privacy guarantees while minimizing computational overhead, enhancing the ZKP Ecosystem's ability to protect sensitive data during AI computations within Substrate's runtime environment.

Homomorphic Masking Explore partially homomorphic encryption (PHE) schemes, such as Paillier, which are more efficient than fully homomorphic encryption for specific operations like addition in AI workloads [100]. Research will optimize PHE for matrix operations critical to neural networks, reducing computational costs within Substrate's weight-based execution model. This approach allows computations on encrypted data without decryption, ensuring that sensitive data remains protected throughout the process, even during intermediate steps of AI training verified through Substrate's verification infrastructure.

Hybrid Privacy Models Combine masking with differential privacy to add noise to outputs, ensuring individual data points remain untraceable even if masked data is compromised [101]. This layered approach strengthens privacy against inference attacks while maintaining compatibility with Substrate's privacy-preserving mechanisms.

Selective Masking Apply masking only to sensitive features (e.g., patient identifiers in healthcare datasets), reducing overhead while maintaining privacy for critical data [102]. This aligns with the ecosystem's efficiency goals and Substrate's optimized execution environment.

Performance Optimization Investigate lightweight masking algorithms, such as format-preserving encryption, to preserve data structure while obscuring sensitive values [103]. This ensures compatibility with real-time AI applications running on Substrate's high-performance runtime.

Integration with ZKP Ensure masking is compatible with zk-SNARKs and zk-STARKs verified through Substrate's verification infrastructure, allowing verifiable computations on masked data without revealing the masks, leveraging the ZKP Blockchain's validation framework integrated with BABE+GRANDPA consensus.

Example Application In a healthcare scenario, patient records could be masked using PHE for sensitive fields like diagnoses, enabling AI model training on encrypted data processed through Substrate's secure runtime. Differential privacy would protect model outputs, ensuring compliance with regulations like GDPR while maintaining auditability through Substrate's immutable storage.

14.2 Noise

We seek to develop adaptive noise mechanisms for differential privacy that dynamically adjust based on data sensitivity and application needs, optimizing privacy-utility trade-offs for secure AI computations within Substrate's modular framework.

Adaptive Noise Scaling Implement mechanisms through custom Substrate pallets to adjust noise levels based on data sensitivity, applying higher noise to highly sensitive data (e.g., financial transactions) and lower noise to aggregated statistics [104].

Privacy Budget Management Develop systems using Substrate's governance mechanisms to allocate and track privacy budgets across multiple queries or training sessions, preventing privacy depletion over time [100].

Integration with Homomorphic Encryption Combine differential privacy with homomorphic encryption to enable computations on encrypted, noisy data within Substrate's runtime environment, adding an extra privacy layer [102].

Scalable Noise Addition Research efficient noise addition methods for distributed settings coordinated through Substrate's off-chain workers, ensuring scalability in federated learning across multiple nodes [105] [120].

Verification of Noise Addition Use zk-SNARKs verified through Substrate's verification infrastructure to verify correct noise application without revealing data or noise values, ensuring transparency.

Example Application In a federated learning setup for fraud detection, banks could add noise to model updates through Substrate pallets, with levels adjusted based on transaction sensitivity. ZKPs would verify noise applications through Substrate's verification system, ensuring trust and compliance with financial privacy standards.

14.3 Advanced Federated Learning

We aim to develop scalable, privacy-preserving federated learning frameworks that integrate zero-knowledge proofs and secure multi-party computation to enhance security and trust in decentralized AI training, leveraging Substrate's off-chain worker infrastructure.

Zero-Knowledge Proofs for Model Updates Use zk-SNARKs verified through Substrate's verification infrastructure to verify the correctness of model updates without revealing gradients, ensuring legitimate contributions.

Secure Aggregation with SMPC Implement efficient SMPC protocols coordinated through Substrate's off-chain workers to aggregate updates securely, preventing any party from accessing others' data [100] [120].

Differential Privacy in Federated Learning Incorporate differential privacy through custom Substrate pallets to add noise to updates, protecting against inference attacks [105].

Scalability Solutions Explore hierarchical federated learning using Substrate's networking capabilities, where local models are aggregated at regional nodes before global aggregation, reducing communication overhead.

Collusion-Resistant Mechanisms Develop reputation systems through Substrate's staking and slashing mechanisms and anomaly detection to identify and mitigate collusion, ensuring trust in collaborative training [101].

Example Application In a global climate modeling project, universities could use hierarchical federated learning coordinated through Substrate's off-chain workers, with regional aggregators handling local updates. ZKPs verify update integrity through Substrate's verification infrastructure, SMPC ensures secure aggregation, and differential privacy protects sensitive research data.

14.4 Privacy Pools for Scalability

We seek to design decentralized privacy pools that batch transactions or computations into single zero-knowledge proofs, enhancing scalability while maintaining privacy and security within Substrate's efficient execution environment.

Recursive Proofs for Batching Use recursive zk-SNARKs verified through Substrate's verification infrastructure to aggregate multiple proofs into one, reducing on-chain verification costs and leveraging Substrate's weight optimization mechanisms.

Decentralized Batching Mechanisms Develop protocols using Substrate's off-chain workers where nodes collaboratively form batches, ensuring no single entity controls the process [106] [120].

Priority-Based Batching Implement priority levels for tasks through Substrate's transaction pool management, processing time-sensitive computations in smaller, frequent batches.

Security and Transparency Ensure individual actions are verifiable before batching through Substrate's verification infrastructure, with on-chain metadata logging in Patricia Tries for auditability.

Cross-Chain Compatibility Leverage Substrate's XCM (Cross-Consensus Message Passing) to enable privacy pools across multiple parachains, expanding scalability benefits throughout the Polkadot ecosystem [115].

Example Application In a smart city, real-time traffic updates could be processed in small batches for low latency through Substrate's efficient runtime, while historical data analysis uses larger batches. Recursive proofs ensure efficient verification through Substrate's verification infrastructure, and IPFS coordinated by off-chain workers handles data storage.

14.5 Substrate-Specific Research Directions

Building on Substrate's unique capabilities, we are exploring additional research areas:

Runtime Upgrade Mechanisms for Privacy Investigate how Substrate's forkless upgrade capability can enable seamless deployment of new privacy-preserving algorithms without network disruption.

Cross-Chain Privacy Protocols Develop privacy-preserving protocols that leverage XCM to enable secure data sharing and computation across different parachains while maintaining zero-knowledge guarantees.

Pallet-Based Privacy Infrastructure Create modular privacy pallets that can be composed and reused across different Substrate-based applications, establishing a standardized privacy infrastructure.

Governance-Driven Privacy Parameters Research dynamic privacy parameter adjustment mechanisms using Substrate's democracy pallet, allowing communities to collectively optimize privacy-utility trade-offs.

This comprehensive research agenda ensures the ZKP Ecosystem remains at the forefront of privacy-preserving computation while leveraging Substrate's advanced capabilities for scalability, interoperability, and sustainable development.

References

[1] Crawford, K. (2021). *Atlas of AI: Power, Politics, and the Planetary Costs of Artificial Intelligence*. Yale University Press.

[2] Liang, P. and Bommasani, R. and Lee, T. and Tsipras, D. and Soylu, D. and Yasunaga, M. and others (2022). Holistic evaluation of language models. *arXiv preprint arXiv:2211.09110*. doi:10.48550/arXiv.2211.09110

[3] Sabt, M. and Achemlal, M. and Bouabdallah, A. (2015). Trusted execution environment: What it is, and what it is not. In *2015 IEEE Trustcom/BigDataSE/ISPA* (Vol. 1, pp. 57–64). doi:10.1109/Trustcom.2015.357

[4] Sasson, E. B. and Chiesa, A. and Garman, C. and Green, M. and Miers, I. and Tromer, E. and Virza, M. (2014). Zerocash: Decentralized anonymous payments from Bitcoin. In *2014 IEEE Symposium on Security and Privacy* (pp. 459–474). doi:10.1109/SP.2014.36

[5] Bommasani, R. and Hudson, D. A. and Adeli, E. and Altman, R. and Arora, S. and von Arx, S. and others (2021). On the opportunities and risks of foundation models. *arXiv preprint arXiv:2108.07258*. doi:10.48550/arXiv.2108.07258

[6] Mohassel, P. and Zhang, Y. (2017). SecureML: A system for scalable privacy-preserving machine learning. In *2017 IEEE Symposium on Security and Privacy (SP)* (pp. 19–38). doi:10.1109/SP.2017.12

[7] Narayanan, A. and Bonneau, J. and Felten, E. and Miller, A. and Goldfeder, S. (2016). *Bitcoin and Cryptocurrency Technologies: A Comprehensive Introduction*. Princeton University Press.

[8] Goldwasser, S. and Micali, S. and Rackoff, C. (1985). The knowledge complexity of interactive proof-systems. In *Proceedings of the Seventeenth Annual ACM Symposium on Theory of Computing* (pp. 291–304). doi:10.1145/22145.22179

[9] Reitwiessner, C. (2016). zk-SNARKs: A practical introduction. *Ethereum Blog*. `https://blog.ethereum.org/2016/12/05/zksnarks-in-a-nutshell/`

[10] Gentry, C. (2009). Fully homomorphic encryption using ideal lattices. In *Proceedings of the Forty-First Annual ACM Symposium on Theory of Computing* (pp. 169–178). doi:10.1145/1536414.1536440

[11] Acar, A. and Aksu, H. and Uluagac, A. S. and Conti, M. (2018). A survey on homomorphic encryption schemes: Theory and implementation. *ACM Computing Surveys (CSUR)*, 51(4), 1–35. doi:10.1145/3214303

[12] Benet, J. (2014). IPFS - Content Addressed, Versioned, P2P File System. *arXiv preprint arXiv:1407.3561*. doi:10.48550/arXiv.1407.3561

[13] Protocol Labs (2017). *Filecoin: A Decentralized Storage Network*. `https://filecoin.io/filecoin.pdf`

[14] Buchman, E. (2016). *Tendermint: Byzantine Fault Tolerance in the Age of Blockchains*. Master's thesis, University of Guelph. `https://atrium.lib.uoguelph.ca/xmlui/handle/10214/9769`

[15] Cosmos (2023). *Cosmos SDK Documentation*. `https://docs.cosmos.network/`

[16] McConaghy, T. and Marques, R. and Muller, A. and De Jonghe, D. and McConaghy, T. and McMullen, G. and Allison, I. (2017). *Ocean Protocol: A Decentralized Data Exchange Protocol*. https://oceanprotocol.com/tech-whitepaper.pdf

[17] Russell, S. (2019). *Human Compatible: Artificial Intelligence and the Problem of Control*. Viking.

[18] Merkle, R. C. (1988). A digital signature based on a conventional encryption function. In *Advances in Cryptology — CRYPTO '87* (pp. 369–378). Springer. doi:10.1007/3-540-48184-2_32

[19] Wood, G. (2014). *Ethereum: A Secure Decentralised Generalised Transaction Ledger*. Ethereum Yellow Paper. https://ethereum.github.io/yellowpaper/paper.pdf

[20] Johnson, D. and Menezes, A. and Vanstone, S. (2001). The Elliptic Curve Digital Signature Algorithm (ECDSA). *International Journal of Information Security*, 1(1), 36–63. doi:10.1007/s102070100002

[21] Bernstein, D. J. and Duif, N. and Lange, T. and Schwabe, P. and Yang, B. Y. (2012). High-speed high-security signatures. *Journal of Cryptographic Engineering*, 2(2), 77–89. doi:10.1007/s13389-012-0027-1

[22] Dziembowski, S. and Faust, S. and Kolmogorov, V. and Pietrzak, K. (2015). Proofs of Space. In *Advances in Cryptology — CRYPTO 2015* (pp. 585–605). Springer. doi:10.1007/978-3-662-48000-7_29

[23] Ben-Sasson, E. and Bentov, I. and Horesh, Y. and Riabzev, M. (2018). Scalable, transparent, and post-quantum secure computational integrity. *Cryptology ePrint Archive, Report 2018/046*. https://eprint.iacr.org/2018/046

[24] Ben-Sasson, E. and Chiesa, A. and Garman, C. and Green, M. and Miers, I. and Tromer, E. and Virza, M. (2014). Zerocash: Decentralized anonymous payments from Bitcoin. In *2014 IEEE Symposium on Security and Privacy* (pp. 459–474). doi:10.1109/SP.2014.36

[25] Maymounkov, P. and Mazieres, D. (2002). Kademlia: A peer-to-peer information system based on the XOR metric. In *Proceedings of the 1st International Workshop on Peer-to-Peer Systems (IPTPS)* (pp. 53–65). Springer. doi:10.1007/3-540-45748-8_5

[26] Seagate (2023). *Hard Drive Power Consumption*. https://www.seagate.com/support/kb/hard-drive-power-consumption-005936en/

[27] Cambridge Bitcoin Electricity Consumption Index (2023). https://ccaf.io/cbeci/index

[28] Reed, I. S. and Solomon, G. (1960). Polynomial codes over certain finite fields. *Journal of the Society for Industrial and Applied Mathematics*, 8(2), 300–304. doi:10.1137/0108018

[29] Pinata (2023). *Pinata Documentation*. https://docs.pinata.cloud/

[30] Groth, J. (2016). On the size of pairing-based non-interactive arguments. In *Advances in Cryptology — EUROCRYPT 2016* (pp. 305–326). Springer. doi:10.1007/978-3-662-49896-5_11

[31] Tendermint Team (2023). *Tendermint Performance Benchmarks*. https://docs.tendermint.com/v0.34/benchmarks/

[32] Groth, J. (2016). On the size of pairing-based non-interactive arguments. In *Advances in Cryptology — EUROCRYPT 2016* (pp. 305–326). Springer. doi:10.1007/978-3-662-49896-5_11

[33] IPFS Documentation (2023). *Measuring the IPFS Network*. https://docs.ipfs.tech/project/measuring-the-network/

[34] Benet, J. (2014). *IPFS Whitepaper: Content Addressed, Versioned, P2P File System*. https://ipfs.io/ipfs/QmR7GSQM93Cx5eAg6a6yRzNde1FQv7uL6X1o4k7zrJa3LX/ipfs.draft3.pdf

[35] Peertechz Publications (2023). Assessment of energy storage technologies for case studies with increased renewable energy penetration. *Peertechz Publications*. https://www.peertechzpublications.org/articles/ARAE-5-113.php

[36] Kosba, A. and others (2020). zk-SNARKs for deep learning. In *Conference on Cryptographic Hardware and Embedded Systems (CHES 2020)* (pp. 1–20). Springer.

[37] Wahby, R. S. and others (2022). Efficient zero-knowledge proofs for neural networks. *Cryptology ePrint Archive, Report 2022/123*. https://eprint.iacr.org/2022/123

[38] Bootle, J. and others (2016). Efficient zero-knowledge proofs for arithmetic circuits. In *Advances in Cryptology — EUROCRYPT 2016* (pp. 1–25). Springer.

[39] Ben-Sasson, E. and others (2019). On the concrete efficiency of zk-SNARKs. *Cryptology ePrint Archive, Report 2019/456*. https://eprint.iacr.org/2019/456

[40] Chen, Y. and others (2021). Decentralized storage: Principles, systems, and the road ahead. *IEEE Transactions on Cloud Computing*, 9(3), 456–467.

[41] StarkWare (2022). *zk-Rollups: A Comprehensive Overview*. StarkWare Industries.

[42] Optimism Team (2023). *Decentralized Rollup Operators: Design and Analysis*. Optimism Documentation.

[43] Matter Labs (2023). *zkSync: Scaling Ethereum with zk-Rollups*. zkSync Whitepaper.

[44] Ben-Sasson, E. and others (2019). On the concrete efficiency of zk-SNARKs. *Cryptology ePrint Archive, Report 2019/457*. https://eprint.iacr.org/2019/457

[45] Micali, S. (1994). CS proofs. In *Foundations of Computer Science* (pp. 436–453).

[46] Bunz, B. and others (2020). Bulletproofs: Short proofs for confidential transactions. In *IEEE Symposium on Security and Privacy* (pp. 123–145).

[47] Chiesa, A. and others (2019). Marlin: Preprocessing zkSNARKs with universal SRS. *Cryptology ePrint Archive, Report 2019/1047*. https://eprint.iacr.org/2019/1047

[48] Haas, A. and others (2017). Bringing the web up to speed with WebAssembly. In *PLDI 2017*. doi:10.1145/3062341.3062363

[49] Valiant, P. (2008). Incrementally verifiable computation. In *FOCS 2008* (pp. 567–576).

[50] Ethereum Foundation (2023). *Ethereum Precompiled Contracts*. https://ethereum.org/en/developers/docs/evm/precompiles/

[51] Strassen, V. (1969). Gaussian elimination is not optimal. *Numerische Mathematik*, 13(4), 354–356. doi:10.1007/BF02165411

[52] Albrecht, M. R. and others (2018). Implementing RLWE-based schemes using an RSA co-processor. *Transactions on Cryptographic Hardware and Embedded Systems*, 2018(1), 169–208. doi:10.13154/tches.v2018.i1.169-208

[53] Ben-Sasson, E. and others (2020). AuroraLight: Improved prover efficiency and SRS size in a Sonic-like system. *Cryptology ePrint Archive, Report 2020/1357*. https://eprint.iacr.org/2020/1357

[54] Ames, S. and Hazay, C. and Ishai, Y. and Venkitasubramaniam, M. (2017). Ligero: Lightweight sublinear arguments without a trusted setup. In *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security* (pp. 2087–2104). doi:10.1145/3133956.3134104

[55] Bootle, J. and others (2020). Efficient zero-knowledge proof systems. In *Handbook of Financial Cryptography and Security* (pp. 37–62). doi:10.1201/9781420072839

[56] Wahby, R. S. and others (2020). Doubly-efficient zkSNARKs without trusted setup. In *IEEE Symposium on Security and Privacy (SP)* (pp. 921–938). doi:10.1109/SP.2019.00041

[57] Groth, J. and others (2018). Updatable and universal common reference strings with applications to zk-SNARKs. In *Advances in Cryptology — CRYPTO 2018* (pp. 698–728). Springer. doi:10.1007/978-3-319-96878-0_24

[58] Chiesa, A. and others (2021). Proof-carrying data from accumulation schemes. *Cryptology ePrint Archive, Report 2021/1215*. https://eprint.iacr.org/2021/1215

[59] Bowe, S. and others (2019). Halo: Recursive proof composition without a trusted setup. *Cryptology ePrint Archive, Report 2019/1021*. https://eprint.iacr.org/2019/1021

[60] Gabizon, A. and Williamson, Z. J. and Ciobotaru, O. (2019). PLONK: Permutations over Lagrange-bases for oecumenical noninteractive arguments of knowledge. *Cryptology ePrint Archive, Report 2019/953*. https://eprint.iacr.org/2019/953

[61] Boneh, D. and others (2020). Halo Infinite: Proof-carrying data from additive polynomial commitments. *Cryptology ePrint Archive, Report 2020/1536*. https://eprint.iacr.org/2020/1536

[62] Tendermint Team (2023). *Tendermint Documentation*. https://docs.tendermint.com/

[63] Cosmos (2023). *Cosmos SDK Documentation*. https://docs.cosmos.network/

[64] Wood, G. (2014). *Ethereum: A Secure Decentralised Generalised Transaction Ledger*. Ethereum Yellow Paper. https://ethereum.github.io/yellowpaper/paper.pdf

[65] Haas, A. and others (2017). Bringing the web up to speed with WebAssembly. In *PLDI 2017*. doi:10.1145/3062341.3062363

[66] Circom Team (2023). *Circom Documentation*. https://docs.circom.io/

[67] ZoKrates Team (2023). *ZoKrates Documentation*. https://zokrates.github.io/

[68] Merkle, R. C. (1988). A digital signature based on a conventional encryption function. In *Advances in Cryptology — CRYPTO '87*. Springer. doi:10.1007/3-540-48184-2_32

[69] Groth, J. (2016). On the size of pairing-based non-interactive arguments. *EUROCRYPT 2016*. https://eprint.iacr.org/2016/260

[70] Ben-Sasson, E. and others (2018). Scalable, transparent, and post-quantum secure computational integrity. *Cryptology ePrint Archive, Report 2018/046.* https://eprint.iacr.org/2018/046

[71] Johnson, D. and Menezes, A. and Vanstone, S. (2001). The Elliptic Curve Digital Signature Algorithm (ECDSA). *International Journal of Information Security*, 1(1), 36–63. doi:10.1007/s102070100002

[72] Bernstein, D. J. and others (2012). High-speed high-security signatures. *Journal of Cryptographic Engineering*, 2(2), 77–89. doi:10.1007/s13389-012-0027-1

[73] Gentry, C. (2009). Fully homomorphic encryption using ideal lattices. *Proceedings of the Forty-First Annual ACM Symposium on Theory of Computing*, 169–178. doi:10.1145/1536414.1536440

[74] Albrecht, M. R. and others (2019). Poseidon: A new hash function for zero-knowledge proof systems. *Cryptology ePrint Archive, Report 2019/458.* https://eprint.iacr.org/2019/458

[75] Chiesa, A. and others (2019). Marlin: Preprocessing zkSNARKs with universal SRS. *Cryptology ePrint Archive, Report 2019/1047.* https://eprint.iacr.org/2019/1047

[76] Cadwalladr, C. and Graham-Harrison, E. (2018). Revealed: 50 million Facebook profiles harvested for Cambridge Analytica in major data breach. *The Guardian.* https://www.theguardian.com/news/2018/mar/17/cambridge-analytica-facebook-influence-us-election

[77] Goldwasser, S. and Micali, S. (1982). *Journal of Computer and System Sciences.*

[78] Nakamoto, S. (2008). *Bitcoin Whitepaper.*

[79] Miller, A. and others (2014). *IEEE Symposium on Security and Privacy.*

[80] Wood, G. (2014). *Ethereum Yellow Paper.*

[81] Ethereum Foundation (2023). *Ethereum.org.* https://ethereum.org

[82] Starkware (2022). *StarkNet Documentation.*

[83] Vogelsteller, F. and Buterin, V. (2015). ERC-20 EIP.

[84] Buterin, V. (2017). *Ethereum Blog.*

[85] Buterin, V. (2017). *Ethereum Whitepaper.*

[86] Antonopoulos, A. M. and Wood, G. (2018). *Mastering Ethereum.*

[87] Benet, J. (2014). *arXiv:1407.3561.*

[88] Ben-Sasson, E. and others (2014). *USENIX Security Symposium.*

[89] Groth, J. (2016). *EUROCRYPT.*

[90] Daemen, J. and Rijmen, V. (2002). Springer.

[91] Protocol Labs (2017). *Filecoin: A Decentralized Storage Network.* Filecoin Whitepaper.

[92] Chainlink Team (2020). *Chainlink Whitepaper: A Decentralized Oracle Network.* https://chain.link/whitepaper

[93] Kairouz, P. and others (2021). Advances and open problems in federated learning. *Foundations and Trends in Machine Learning*.

[94] McMahan, H. B. and others (2017). Communication-efficient learning of deep networks from decentralized data. In *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics (AISTATS)*.

[95] Berghel, H. (2017). Equifax and the latest round of identity theft roulette. *IEEE Computer*, 50(12), 72–76.

[96] Chen, T. and others (2017). Understanding Ethereum gas: A comprehensive analysis of transaction costs. *IEEE Transactions on Network and Service Management*.

[97] Kalodner, H. and others (2020). zk-Rollups: Scalable privacy-preserving smart contracts. *IACR Cryptology ePrint Archive*.

[98] Gennaro, R. and others (2018). Threshold-optimal DSA/ECDSA signatures and an application to Bitcoin wallet security. In *International Conference on Applied Cryptography and Network Security (ACNS)*.

[99] Zhang, Q. and others (2022). Optimizing IPFS for large-scale data storage: A performance study. *IEEE International Conference on Blockchain and Cryptocurrency (ICBC)*.

[100] *Privacy-Preserving Techniques in Generative AI and Large Language Models* (2024). MDPI. `https://www.mdpi.com/2078-2489/15/11/697`

[101] *Preserving Data Privacy in Machine Learning Systems* (2024). ScienceDirect. `https://www.sciencedirect.com/science/article/pii/S0167404823005151`

[102] *Privacy-First: A Look at Cutting-Edge Privacy-Enhancing Technologies* (2024). Velotix. `https://www.velotix.ai/resources/blog/privacy-first-a-look-at-cutting-edge-privacy-enhancing-technologies/`

[103] *Data Masking: Protecting Sensitive Information* (2024). Snowflake. `https://www.snowflake.com/guides/data-masking/`

[104] *Privacy-Preserving Technologies: Advancements and Impact on Digital Security* (2024). GIRA Group. `https://www.gira.group/post/privacy-preserving-technologies`

[105] *Advancements in Privacy-Preserving Techniques for Federated Learning: A Machine Learning Perspective* (2024). ResearchGate. `https://www.researchgate.net/publication/379857791_Advancements_in_Privacy-Preserving_Techniques_for_Federated_Learning_A_Machine_Learning_Perspective`

[106] *Privacy-Enhancing Technologies Overview* (2024). Wikipedia. `https://en.wikipedia.org/wiki/Privacy-enhancing_technologies`

[107] Parity Technologies (2024). *Substrate: A Modular Framework for Building Blockchains*. `https://substrate.io/`

[108] Wood, G. and Parity Technologies (2024). *Substrate Developer Hub Documentation*. `https://docs.substrate.io/`

[109] Stewart, A. and Kokoris-Kogia, E. and Jovanovic, P. and Gasser, L. and Gailly, N. and Khoffi, I. and Ford, B. (2019). BABE: Blind Assignment for Blockchain Extension. *Web3 Foundation Research*. `https://research.web3.foundation/en/latest/polkadot/block-production/Babe.html`

[110] Stewart, A. and Kokoris-Kogia, E. (2020). GRANDPA: A Byzantine Fault Tolerant Finality Gadget. *Web3 Foundation Research*. https://research.web3.foundation/en/latest/polkadot/finality.html

[111] Parity Technologies (2024). *FRAME: Framework for Runtime Aggregation of Modularized Entities*. https://docs.substrate.io/reference/frame-pallets/

[112] Parity Technologies (2024). *EVM Pallet: Ethereum Virtual Machine for Substrate*. https://docs.substrate.io/reference/frame-pallets/#evm

[113] Parity Technologies (2024). *Frontier: Ethereum Compatibility Layer for Substrate*. https://github.com/paritytech/frontier

[114] Morrison, D. R. (1968). PATRICIA—Practical Algorithm to Retrieve Information Coded in Alphanumeric. *Journal of the ACM*, 15(4), 514-534. doi:10.1145/321479.321481

[115] Wood, G. and Parity Technologies (2024). *Cross-Chain Message Passing (XCMP) Protocol*. Polkadot Wiki. https://wiki.polkadot.network/docs/learn-xcm

[116] Parity Technologies (2024). *Substrate Runtime Events and Dispatch*. https://docs.substrate.io/reference/how-to-guides/basics/use-runtime-events/

[117] Parity Technologies (2024). *Substrate Transaction Weights and Fees*. https://docs.substrate.io/build/tx-weights-fees/

[118] Parity Technologies (2024). *Substrate Performance Benchmarks and Optimization*. https://docs.substrate.io/test/benchmark/

[119] Parity Technologies (2024). *Substrate WASM Runtime Environment*. https://docs.substrate.io/build/runtime-storage/

[120] Parity Technologies (2024). *Substrate Off-chain Workers*. https://docs.substrate.io/reference/how-to-guides/offchain-workers/

[121] Parity Technologies (2024). *Substrate Custom Pallets for Zero-Knowledge Verification*. https://docs.substrate.io/reference/how-to-guides/pallet-design/

[122] Stewart, A. and Kokoris-Kogia, E. (2019). *Randomness Beacon in BABE Consensus*. Web3 Foundation Research. https://research.web3.foundation/en/latest/polkadot/overview/2-token-economics.html

[123] Parity Technologies (2024). *Substrate Assets Pallet and Token Standards*. https://docs.substrate.io/reference/frame-pallets/#assets

[124] Parity Technologies (2024). *Substrate Democracy Pallet*. https://docs.substrate.io/reference/frame-pallets/#democracy