Praman Setu

The Internet of Blockchains

Pranavi Jayasankaran and Vidhan Mangla

Abstract

Blockchains, by design, are deterministic, closed-loop systems incapable of natively accessing external, non-deterministic data. While this isolation preserves security and consensus, it also restricts the scope of smart contracts, limiting their ability to interact with real-world events and systems. This challenge, known as the oracle problem which must be addressed for decentralised applications to achieve their full potential.

This whitepaper introduces Praman Setu, developed by GenesisCipher Labs, a next-generation Decentralised Oracle Network (DON) that acts as a secure and reliable middleware layer between blockchains and external data sources. Praman Setu employs the Proof of Quartiles consensus mechanism, a statistically driven approach designed to ensure data integrity by detecting and mitigating the influence of erroneous or extreme data points. By bridging on-chain logic with off-chain data and computational environments, Praman Setu enables trust-minimised, fault-tolerant integration of real-world information into blockchain ecosystems, removing dependence on a single point of failure. This paper further explores its architecture, services, and value propositions across diverse application domains.

1. Introduction

A blockchain is a decentralised, tamper-resistant digital ledger that organises data into cryptographically linked blocks, enabling secure recording of transactions and tracking of assets across a peer-to-peer network without reliance on a central authority [1]. The emergence of this blockchain technology has transformed the way digital value and information are exchanged. By enabling secure, transparent, and distributed systems, blockchains have provided the foundation for a wide range of applications—from financial transactions to digital identity, healthcare, and logistics—demonstrating their versatility in managing trustless interactions [2].

The decentralised nature of blockchain ensures that no single privileged entity has unilateral control, which enhances trust and reduces vulnerabilities associated with centralised systems that contain single points of failure [1][3]. However, the very properties that make blockchain secure, immutable, isolated, and consensus-driven also impose limitations on its ability to interact with external environments. This gap has necessitated the development of decentralised oracle networks, which extend blockchain functionality by enabling access to real-world data and events [4].

1.1 Smart Contracts

The introduction of smart contracts has enabled programmable and automated interactions. Historically, contracts embodied in code have run in a centralised manner that leaves them subject to alteration, termination, and even deletion by a privileged party. To fix this limitation of digital contracts, tamper proof smart contracts were introduced [3].

A smart contract is a self-executing code embedded within the blockchain that enforces rules and conditions agreed upon by participating parties. Once predefined conditions are satisfied, the contract automatically executes the corresponding actions without requiring human intervention [5].

In the same way that legal contracts are expressed in natural languages like English, smart contracts are written in specialised programming languages. The most widely adopted among these is Solidity [6], which allows developers to define the logic, state variables, and functions that govern contract execution.

Sample Smart Contract

The following is a basic smart contract written in Solidity that stores and retrieves a number.

```
// SPDX-License-Identifier: GPL-3.0
pragma solidity >=0.8.2 <0.9.0;

contract Storage {
    uint256 number;

    function store(uint256 num) public {
        number = num;
    }

    function retrieve() public view returns (uint256){
        return number;
    }
}
```

Source: Solidity Documentation, "Introduction to Smart Contracts" [6]

1.2 Oracles

Once deployed, smart contracts operate deterministically across the blockchain network. However, their deterministic nature presents a challenge: many smart contracts rely on external data inputs (such as asset prices, weather reports, or identity verification) to determine whether contractual conditions have been met. As blockchain networks lack native access to off-chain data, a middleware infrastructure is required to securely connect their on-chain logic with external information sources [7].

Oracles serve as a secure and reliable data gateway, performing the essential functions of fetching, verifying, and transmitting external information to smart contracts [8]. By providing this crucial data feed, oracles enable smart contracts to execute based on real-world inputs and outputs, vastly expanding their potential applications from decentralised finance (DeFi) and insurance to supply chain management and beyond [9].

2. The Oracle Problem

Blockchains are inherently deterministic and self-contained systems. This deterministic design ensures consensus across a distributed network, wherein every full node on the network must be able to re-execute every transaction and arrive at the exact same state change [10], but it also introduces a fundamental limitation: smart contracts cannot directly access or verify information from external environments. This limitation is commonly referred to as the oracle problem. In essence, blockchains cannot natively import data from or export data to external systems [4]. The decentralised nature of blockchains has created a foundation for building Decentralised Applications (DApps), where ensuring data integrity and consistency is of critical importance [11].

However, for decentralised applications to perform meaningful operations, they often require real world inputs such as financial market prices, sensor readings, weather conditions, or identity credentials. Unlike on-chain transactions, external data is non-deterministic. There is no cryptographic truth within the blockchain that can prove the value of an asset, the outcome of an election, or the reading of a device [2][12].

2.1 The Limitations of Centralised Oracles

Early approaches sought to bridge this gap between on-chain smart contracts and off-chain data sources through centralised oracles. Although these solutions enabled fundamental functionality, they reintroduced a Single Point of Failure (SPOF) and undermined the core principle of decentralisation in blockchain networks [13]. By relying on a centralised oracle, the smart contract, which is designed to operate without trust, became dependent on the external provider of data. This dependency meant that the overall security of a decentralised application was contingent on the integrity of its oracle mechanism [7].

This created a serious vulnerability. If an oracle were compromised, a malicious actor could inject false or manipulated data into the blockchain [14]. Since smart contract executions are immutable and irreversible, acting on corrupted inputs could result in financial losses, invalid outcomes, or systemic failures. In such cases, the blockchain itself would remain secure, but the application layer would be undermined, effectively reintroducing the very trusted third party that blockchains were designed to remove [15].

The challenge is therefore to design oracle mechanisms that can deliver data with a level of security, reliability, and resistance to manipulation that matches the guarantees of the blockchain itself.

2.2 Decentralised Oracle Networks

Decentralised Oracle Networks (DONs) were developed as a direct solution to the vulnerabilities posed by centralised oracles. A DON mitigates the risk of a single point of failure by distributing the responsibility of data sourcing and validation across a wide network of independent, geographically dispersed oracle nodes [4][16]. The operational framework of a DON is structured to establish trust through decentralisation.

When a smart contract initiates a data request, multiple nodes within the network independently retrieve information from diverse off-chain sources. These nodes then participate in a consensus protocol to agree on a unified, validated data point before it is transmitted on-chain [7]. By aggregating responses from multiple nodes and multiple data sources, the network ensures that the outcome is reliable, precise, and resistant to manipulation by any single malicious actor [4]. This mechanism reflects the core principles of blockchain, shifting reliance from a central authority to a model based on distributed, verifiable consensus [17].

However, the enhanced security and reliability of DONs introduce trade-offs. They require more complex architectures involving node coordination, consensus mechanisms, and economic incentives [18]. This complexity can increase operational costs and introduce higher latency compared to a single centralised provider. The choice between centralised and decentralised oracles therefore becomes a critical design decision, balancing trust minimisation and security against efficiency and cost [16].

Praman Setu is engineered to address these trade-offs; it provides a cost-effective, robust, and secure solution that enables the seamless integration of real-world data into blockchain smart contracts.

3. Praman Setu Overview

Praman Setu is a decentralised oracle network that sources information from multiple data providers and external APIs. This data is then aggregated and transmitted to the oracle contract, ensuring reliability and resilience against faulty inputs or exceptions. When a client initiates a transaction

request on the blockchain, the corresponding smart contract issues a call to the oracle for the required real-world data. The oracle network generates a request id, collects, validates, and returns this information to the smart contract, which subsequently executes operations based on predefined conditions. This process enables seamless integration of off-chain data into on-chain logic, ensuring secure and trust-minimised execution of decentralised applications.

3.1 Proof of Quartiles

Praman Setu introduces a novel data consensus mechanism called Proof of Quartiles, a statistical approach designed to ensure the accuracy and reliability of data delivered by the oracle network. This method leverages quartiles and the interquartile range (IQR) to filter out outliers and arrive at a robust, aggregated value. By focusing on the central tendency of the data, Proof of Quartiles effectively mitigates the impact of erroneous or malicious data points, thereby enhancing the overall security and integrity of the oracle network [19].

3.1.1 The Rationale for a Statistical Approach

Decentralised oracle networks face a critical trade-off consolidating varied data points from multiple independent sources into a single reliable value. Naive approaches like simple averaging are highly susceptible to manipulation, as a small set of malicious nodes providing outlier values can distort the outcome. Meanwhile, traditional consensus protocols, although effective in achieving agreement, tend to introduce latency and computational strain overhead [9].

Praman Setu addresses these challenges by adopting a statistical approach rooted in robust statistics. The rationale behind Proof of Quartiles is to achieve a high degree of data accuracy and resilience against outliers without incurring the computational burden of complex cryptographic consensus protocols for every data point. By systematically identifying and eliminating extreme values, the mechanism ensures that the final aggregated data reflects the true underlying reality, even in the presence of faulty or adversarial oracle data nodes. This approach is particularly effective in scenarios where a significant portion of the network is honest, but a minority might attempt to manipulate data. Moreover, Proof of Quartiles enables Praman Setu to lower operational costs by reducing the need for resource-intensive consensus computations.

3.1.2 Computations

The Proof of Quartiles mechanism operates on a set of n observations O = {o1, o2, ..., on} reported by the oracle nodes in a committee. The process unfolds as follows:

1. Sorting: The initial step involves sorting the observations in ascending order, resulting in a sorted set S = {s1, s2, ..., sn} where s1 <= s2 <= ... <= s<n.

2. Quartile Calculation: The first quartile (Q1), second quartile (Q2, the median), and third quartile (Q3) are calculated. For a data set with n elements, the positions of the quartiles are determined as follows:

- Q1 Position: $L_1 = \frac{(n+1)}{4}$
- Q2 Position: $L_2 = \frac{(n+1)}{2}$
- Q3 Position: $L_3 = \frac{3(n+1)}{4}$

If the position Li is an integer, the quartile Qi is simply the value at that position in the sorted dataset. If Li is not an integer, interpolation is used. For example, if Li = k + 0.5, then $Q_i = \frac{(s_k + s_{(k+1)})}{2}$.

3. Interquartile Range (IQR): The IQR is calculated as the difference between the third and first quartiles:

- IQR = Q3 - Q1

The IQR represents the spread of the middle 50% of the data, providing a robust measure of statistical dispersion that is less sensitive to outliers than the standard deviation.

4. Outlier Elimination: The core of the Proof of Quartiles mechanism lies in the elimination of outliers. Any observation that falls outside the range defined by the following lower and upper bounds is discarded:

- Lower Bound (LB) = Q1 - 1.5 * IQR
- Upper Bound (UB) = Q3 + 1.5 * IQR

This method, known as Tukey's fences, effectively identifies and removes extreme values that could skew the aggregated result. Observations oj are retained if LB <= oj <= UB.

5. Interquartile Mean (IQM): After eliminating the outliers, the final aggregated value is calculated as the mean of the remaining observations within the interquartile range. Let R be the set of remaining observations after outlier elimination. The Interquartile Mean (IQM) is then calculated as:

- IQM = (1 / |R|) * Σ(r_k) for all r_k in R

This Interquartile Mean (IQM) provides a robust and reliable measure of the central tendency of the data, as it is not influenced by the discarded outliers.

By employing this statistical approach, Praman Setu ensures that the data delivered to smart contracts is not only accurate but also resilient to manipulation and collusion among a minority of malicious nodes. The Proof of Quartiles mechanism provides a verifiable and transparent method for data validation, enhancing the overall security and trustworthiness of the oracle network.

4. Architecture Overview

Praman Setu's architecture has been designed to provide a seamless bridge between decentralised on-chain systems and diverse off-chain data sources. The design is structured around two complementary tiers or levels [13]. The on-chain architecture manages the interaction between the blockchain environment and the oracle system, ensuring that validated data is securely consumed by smart contracts and decentralised applications. The off-chain architecture manages the interaction between external data sources and the oracle network, handling data collection, validation, and preparation before it is relayed on-chain. These two tiers work together to provide a secure, verifiable, and efficient pipeline that enables trusted data exchange across decentralised ecosystems.

4.1 On-Chain Architecture

The on-chain architecture of Praman Setu's decentralised oracle network (DON) is primarily composed of two foundational components: smart contracts, known as Consumer Contracts, which reside on the native blockchain, and Oracle Contracts, which serve as the vital bridge connecting the deterministic, self-contained blockchain ecosystem with the probabilistic, information-rich off-chain world [4].

4.1.1 Consumer Contracts

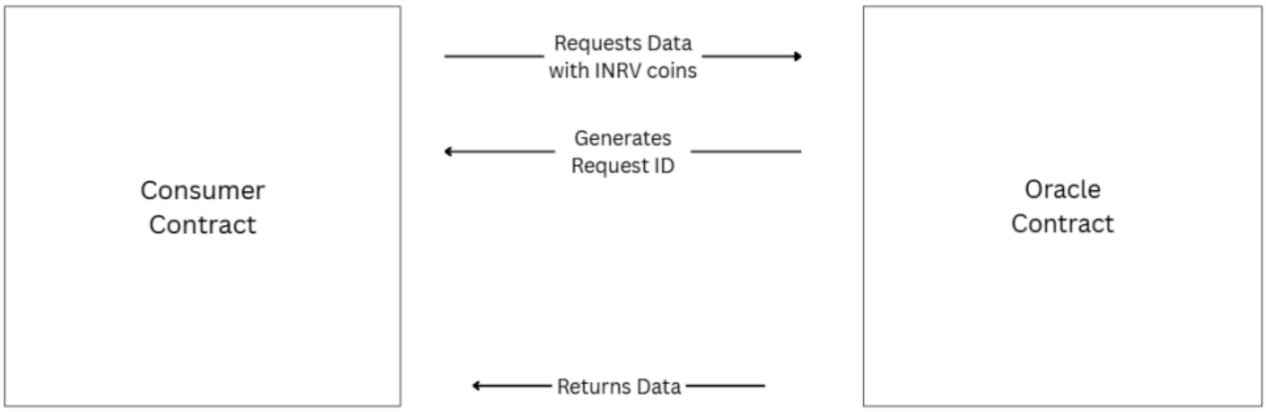
A consumer contract is a self-executing smart contract deployed on the blockchain that relies on the oracle network to access external data. For example, a DeFi lending protocol relies on accurate, real-time market prices to initiate positions, manage settlements, and liquidate undercollateralised loans in a way that is globally consistent and fair to the market [20]. A consumer contract can submit a request to the Praman Setu oracle network via oracle contract on-chain. Each request includes the necessary parameters that define the type of data to be retrieved and the corresponding action to be executed upon receipt. To ensure secure and accountable usage, every client request must be accompanied by a payment in INRV tokens.

4.1.2 Oracle Contract

An oracle request contract is just a specialised smart contract that the consumer contract interacts with by using the oracle interface to get the information it requires.

Request ID Generation

Upon validating both the request and the corresponding payment, the oracle contract generates a unique request ID. This identifier is used by the consumer contract to verify and track the data recipient throughout the fulfilment process. If the payment or request validation fails, the client is returned an error, thereby maintaining the security and reliability of the Praman Setu oracle network.



Request ID Generation

After a request is successfully registered, the oracle contract records it as an event on the blockchain. This event is broadcast across the network and detected by participating off-chain oracle nodes [20]. These nodes are then responsible for gathering data from external sources, performing aggregation using the proof of quartiles consensus mechanism, and submitting the validated results back on-chain. The processed data is subsequently made available to the consumer contract, ensuring that the smart contract operates on trusted and tamper-resistant information.

The consumer contract and oracle contract work complementarily on the chain to create a complete, end-to-end request and fulfilment cycle.

4.2 Off-Chain Architecture

The off-chain architecture of Praman Setu is designed to ensure efficient data collection, validation, and aggregation before it is transmitted to the blockchain. By leveraging decentralised oracle nodes, aggregators and off-chain services this architecture reduces on-chain computational overhead while maintaining high levels of accuracy, reliability, and trust-minimisation.

4.2.1 Oracle nodes

The oracle nodes function as the core off-chain components. They are responsible for interacting directly with external data environments. The oracle nodes retrieve data from multiple trusted sources such as APIs, websites, and databases, which may vary in quality, freshness, or update frequency. Once the information is collected, the oracle nodes.

4.2.2 Aggregators and off-chain services

Aggregators and off-chain services work closely with oracle nodes to improve the accuracy and consistency of data prior to its delivery on-chain. By consolidating oracle responses off-chain and relaying a single validated message to the oracle contract, they offer a more efficient and cost-effective solution [3]. This reduces the computational overhead on the blockchain and increases the efficiency of the decentralised networks.

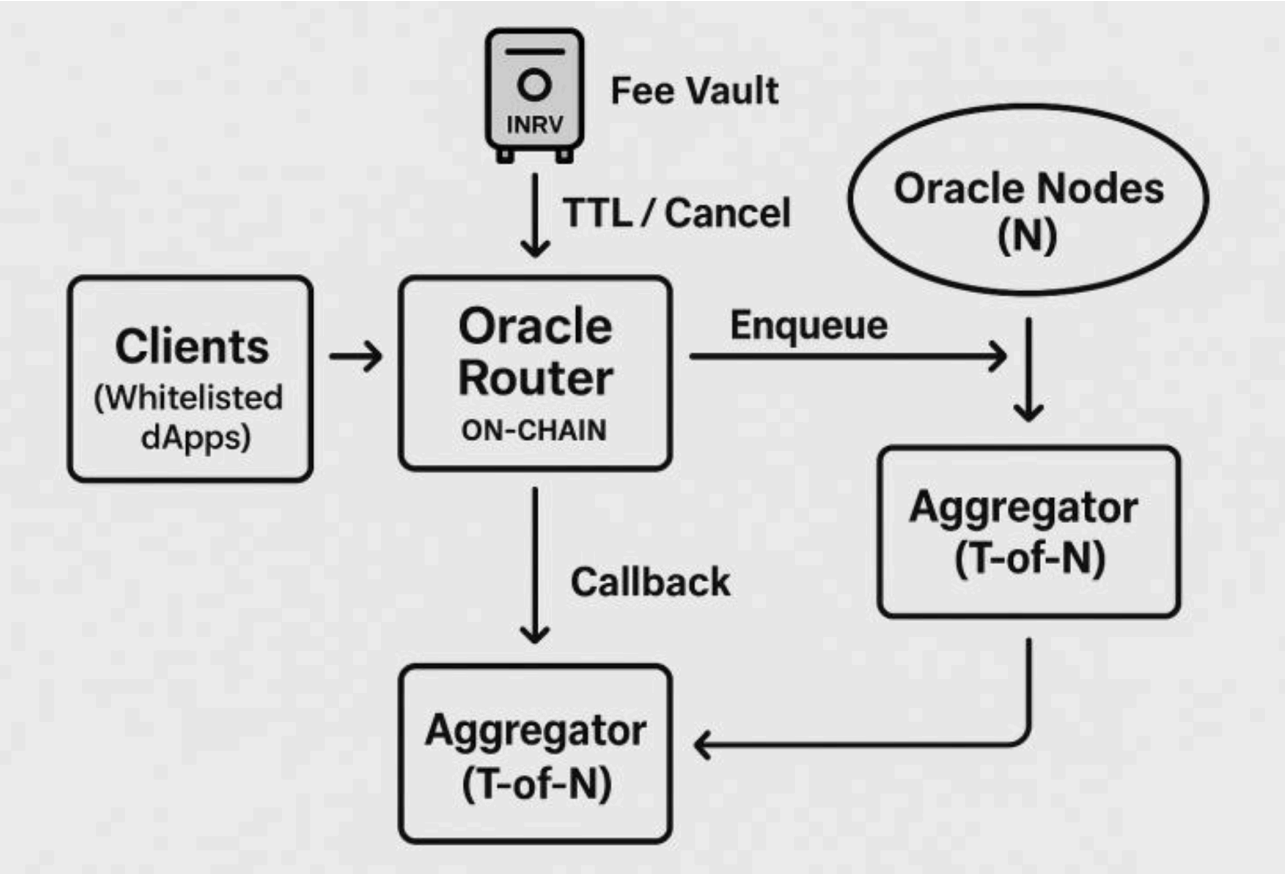
4.2.3 Data-Feeder APIs

Data-feeder APIs establish the pipeline that brings real-world information onto the blockchain. Oracle nodes utilise these APIs to collect data from a wide range of sources, including public APIs, enterprise databases, IoT sensors, and financial data aggregators. By retrieving information from multiple reputable off-chain sources, the system ensures both accuracy and resilience, avoiding reliance on a single point of failure [21]. This mechanism guarantees that decentralised applications receive timely, reliable, and high-quality data for their operations.

By combining on-chain transparency with off-chain efficiency, Praman Setu delivers a hybrid oracle model that balances trust, scalability, cost and performance without compromising on the decentralisation and immutability that blockchain promises. This dual-layered design positions Praman Setu as a next-generation oracle solution, bridging the gap between blockchains and the external world with unprecedented reliability.

5. The Request-to-Report Flow

When a client smart contract initiates a data request, Praman Setu triggers a structured sequence of events that governs the flow from request initiation to final reporting on-chain. The process unfolds as follows:



Request-to-Report Flow

- Request Initiation: A client (consumer contract) on the blockchain submits a data request to the Praman Setu Oracle Contract. The request specifies the type of data required, the preferred data sources, and any additional parameters. The client makes a payment for the request in INRV tokens.
- Request ID Generation: Upon receiving the request, the Oracle Contract validates both the parameters and the associated payment. It then generates a unique Request ID, which is returned to the client contract. This Request ID serves as the identifier for tracking the request throughout its lifecycle.
- Off-Chain Data Fetching: The Oracle nodes proceed to fetch the requested data from multiple independent external sources (data feeder APIs). This multi-source approach strengthens availability and ensures resilience against downtime or inaccuracies from any single provider.
- Data Aggregation and Off-Chain Computation: The Aggregator Module collects data retrieved from the various sources and performs off-chain computations. This process ensures the data is standardised, consolidated, and prepared for consensus-based validation.
- Callback and On-Chain Reporting: Once aggregation is complete, the Oracle Node returns the processed data to the Oracle Contract on-chain. The data is submitted together with the corresponding Request ID via a callback mechanism, ensuring that the information is tied to the correct client request.
- On-Chain validation and consensus: The Oracle Contract validates the received data and applies Praman Setu's Proof of Quartiles consensus mechanism. This step is critical for mitigating the impact of outliers and discarding potentially malicious submissions, resulting in a robust and trustworthy data set.
- Data delivery: Following validation, the Oracle Contract delivers the final, consensus-approved data to the requesting consumer contract. The client contract can then incorporate this real-world data into its execution logic, enabling secure and reliable decentralised applications.

6. Value Proposition

Praman Setu represents a compelling opportunity at the intersection of blockchain infrastructure and real-world data integration. As decentralised applications increasingly rely on reliable, accurate, and timely information, the demand for robust oracle solutions continues to expand. By addressing this need through a secure, scalable, and economically sustainable architecture, Praman Setu positions itself as a critical enabler of the Web3 ecosystem while offering investors exposure to one of the most promising growth opportunities in the blockchain space.

Engineered to overcome the inherent challenges of blockchain interoperability and data accessibility, Praman Setu delivers value through four foundational pillars: architectural evolvability through upgradable smart contracts, economic efficiency via off-chain computation, a native stablecoin and extensive versatility across high-growth industries.

6.1 Upgradable Smart Contracts

A key advantage of smart contracts on a blockchain is the security and permanence enabled by immutable code. This ensures trust by preventing malicious or arbitrary modifications to contract logic, while also guaranteeing that once deployed, contracts remain on the blockchain unless explicitly programmed otherwise [22][23]. However, this intrinsic property presents a significant challenge for long-term protocol maintenance and evolution, as it precludes the ability to patch vulnerabilities, errors, respond to environmental shifts, or introduce new functionalities post-deployment which is an essential feature for long-term adaptability [23] To reconcile the need for both security and adaptability, Praman Setu implements a structured upgradability model of smart contracts. This capability ensures that clients retain a degree of control and flexibility, enabling them to adapt to new requirements and rectify any issues that may arise.

6.2 Low Costs and Efficient Payment Systems

For a decentralised oracle network to achieve widespread adoption, it must strike a balance between technical robustness and economic sustainability. Challenges such as high transaction costs, inefficient computation, and volatility in payment mechanisms frequently impede the growth of blockchain ecosystems [24]. Praman Setu addresses these challenges through the use of INRV for payments, off-chain computations and its Proof of Quartiles consensus mechanism, which significantly reduces costs. As a result, Praman Setu is positioned as one of the most cost-efficient oracle networks in the market, designed with a focus on scalability, stability, and long-term sustainability.

6.2.1 INRV Payments

To incentivise computation within the network, a reliable mechanism for value transfer is essential [25]. Praman Setu utilises INRV, a proprietary stablecoin developed by GenesisCipher Labs, to facilitate transactions between smart contracts and blockchains. INRV is designed to provide stability, security, and efficiency across the ecosystem. By using INRV for payments, the network

ensures predictable transaction costs, mitigates volatility risks, and establishes a dependable medium of exchange for oracle services and decentralised applications.

6.2.2 Whitelist

By default, consumer contracts must pay in INRV to generate a request ID and trigger oracle responses. However, select users may be granted whitelist access, allowing them to submit requests without upfront payment. This mechanism reduces entry barriers for early adopters, accelerates ecosystem adoption, and strategically drives network growth while maintaining sustainable long-term incentives.

6.3 Off-Chain Computation

In use cases like credential-dependent APIs, oracle applications demand capabilities that extend beyond basic data transmission. Off-chain aggregation and computation are integral to enhancing efficiency, scalability, and cost-effectiveness. By processing and consolidating raw data from multiple sources before transmitting it on-chain, the network not only reduces operational overhead but also strengthens its value proposition for large-scale adoption [3].

This design delivers several advantages:

1. Reduced On-Chain Computation: Since blockchain environments are resource-intensive and costly due to consensus and immutability requirements, performing computation off-chain minimises on-chain workload, resulting in faster and cheaper transactions.

2. Improved Efficiency: Off-chain aggregation enables complex calculations, filtering, and validation in a scalable environment.

3. Cost Effectiveness: By reducing both the volume of data and complexity of on-chain operations, gas fees and related expenses are significantly lowered, making oracle services more economically viable.

4. Enhanced Data Quality: Off-chain aggregation allows advanced validation and filtering methods, such as quartile-based techniques, ensuring accurate, consistent, and outlier-resistant data reaches the blockchain.

6.4 Use Cases and Growth Opportunities

Blockchain technology powered by smart contracts is expanding rapidly beyond cryptocurrencies into areas like digital identity, healthcare, and logistics, showcasing its ability to deliver secure, transparent, and trust-minimised interactions across a wide range of industries [3][25]. Praman Setu, as a decentralised oracle network, amplifies this potential by providing smart contracts and decentralised applications with reliable access to real-world data.

With the ability to deliver tamper-resistant, accurate, and timely information, Praman Setu is positioned as an indispensable component for a wide range of industries and applications [26], including:

- Decentralised Finance (DeFi)
- Insurance and Risk Management
- Decentralised Applications (dApps)
- Supply Chain and Trade Finance
- Gaming and NFTs
- Governance and Compliance
- Enterprise Integration

The breadth of these use cases reflects the significant market opportunity for oracle solutions. In essence, any application requiring non-deterministic, real-world data can leverage Praman Setu as its trusted bridge between off-chain information and on-chain execution.

6.5 Security

Safeguarding the integrity, accuracy, and reliability of external data is paramount, as compromised oracles can lead to severe exploits and substantial financial losses [9]. The security framework of Praman Setu is engineered to protect data across its entire lifecycle. Each request is uniquely identified with a request ID, ensuring both traceability and integrity. Off-chain computation minimises on-chain exposure, while the proprietary security architecture guarantees that data remains protected from the moment it is requested by a smart contract until it is delivered.

This layered approach not only provides strong safeguards against tampering but also ensures that oracle-delivered data remains consistently accurate and secure. For investors, this translates into a

resilient Web3 infrastructure capable of supporting mission-critical applications at scale, strengthening both adoption and long-term ecosystem value.

7. Conclusion

In conclusion, while the isolation of blockchain networks is fundamental to ensuring their security and immutability, it also poses a major obstacle for smart contracts that depend on real-world data to function effectively [4][27]. This limitation, widely recognised as the oracle problem, has long hindered the expansion and practical utility of decentralised applications. In the absence of trustworthy access to dynamic external information, smart contracts are unable to interact with real-world events, thereby restricting the scope of applications and services that can be developed on blockchain [3].

Praman Setu is addressing this problem head-on by establishing a decentralised oracle network that bridges the gap between blockchain's immutability and the need for timely, accurate and cost-effective data [28]. Through its innovative multi-source aggregation and Proof of Quartiles validation, Praman Setu ensures that developers and enterprises can access trusted data feeds while preserving transparency and security.

As development progresses, the focus will remain on scaling the system and expanding adoption across diverse sectors, enabling next-generation decentralised solutions. For those supporting its growth, Praman Setu represents a long-term opportunity to be part of the infrastructure that underpins the future of decentralised applications.

8. References

1. S. Ellis, A. Juels, and S. Nazarov, ChainLink: A Decentralised Oracle Network, White Paper, v1.0, Sept. 2017.

2. IBM, "Blockchain," IBM Think. Available: https://www.ibm.com/think/topics/blockchain

3. M. Javaid, A. Haleem, R. P. Singh, R. Suman, and S. Khan, "A review of Blockchain Technology applications for financial services," ScienceDirect. Available: https://www.sciencedirect.com/science/article/pii/S2772485922000606

4. Chainlink, "The Oracle Problem," Chainlink Education Hub. Available: https://chain.link/education-hub/oracle-problem

5. Ministry of Electronics & Information Technology (MeitY), "Smart Contracts," Blockchain Portal. Available: https://blockchain.meity.gov.in/index.php/articles/125-smart-contracts-2

6. Solidity Documentation, "Introduction to Smart Contracts". Available: https://docs.soliditylang.org

7. Stellar, "Smart Contract Basics: Oracles," Stellar Learn. Available: https://stellar.org/learn/smart-contract-basics-oracles

8. ChainUp, "Oracle Smart Contract Integration," ChainUp Blog. Available: https://www.chainup.com/blog/oracle-smart-contract-integration/

9. J. Emmanuel and J. Mayor, "Decentralised Oracle Networks and Data Integrity in DeFi," 2023.

10. Coinweb, "Understanding Deterministic Computations in Blockchain," Medium. Available: https://medium.com/@Coinweb.io/understanding-deterministic-computations-in-blockchain-c567164636e1

11. Y. Hu, M. Liyanage, A. Manzoor, K. Thilakarathna, G. Jourjon, A. Seneviratne, and M. Ylianttila, "The use of smart contracts and challenges," arXiv preprint, arXiv:1810.04699, 2018.

12. H. O. Ohize, A. J. Onumanyi, B. U. Umar, et al., "Blockchain for securing electronic voting systems: a survey of architectures, trends, solutions, and challenges," Cluster Comput., vol. 28, p. 132, 2025. doi: https://doi.org/10.1007/s10586-024-04709-8

13. D. Basile, V. Goretti, C. Di Ciccio, and S. Kirrane, "Enhancing Blockchain-based Processes with Decentralised Oracles," Lecture Notes in Computer Science, 2023.

14. Witnet, "Unveiling Common Misconceptions About Crypto Oracles and Understanding Their Role as Truth Engines," Medium. Available: https://medium.com/witnet/unveiling-

common-misconceptions-about-crypto-oracles-and-understanding-their-role-as-truth-engines-2a724b5f0a90

15. S. Lee et al., "Decentralised Oracles and Blockchain Security," Applied Sciences, vol. 15, no. 11, p. 5924, 2025. Available: https://www.mdpi.com/2076-3417/15/11/5924

16. B. Jay, "Decentralised Oracles vs. Centralised Oracles: A Deep Dive into Trust Models," Dev.to. Available: https://dev.to/bandojay/decentralised-oracles-vs-centralised-oracles-a-deep-dive-into-trust-models-bb2

17. Gate.io, "Types of Blockchain Oracle Attacks: Cases and Multi-layer Defense Strategies," Gate Learn. Available: https://www.gate.com/learn/articles/types-of-blockchain-oracle-attacks-cases-and-multi-layer-defense-strategies/5498

18. Prolitus, "Decentralized oracle networks vs centralized oracles: A comparative analysis," Medium. Available: https://medium.com/@Prolitus01/decentralized-oracle-networks-vs-centralized-oracles-a-comparative-analysis-ccce05e45c27

19. D. Sharma, "IQR (Interquartile Range) for anomaly detection," Medium. Available: https://medium.com/@divyansh9144/iqr-interquartile-range-for-anomaly-detection-f9c568d1195f

20. Kaleido, "How Chainlink Works: Under the Covers," Blockchain Blog. Available: https://www.kaleido.io/blockchain-blog/how-chainlink-works-under-the-covers

21. Chainlink, "Data Feeds," Chainlink Docs. Available: https://chain.link/data-feeds

22. S. Nakamoto, "Bitcoin: A Peer-to-Peer Electronic Cash System," 2008. Available: https://bitcoin.org/bitcoin.pdf

23. X. Gauci-Maistre Xynou, "Immutability in a smart contract: a blessing or a curse?," Lexology. Available: https://www.lexology.com/library/detail.aspx?g=9b0e1787-f6cc-428a-8d55-93e5994bf416

24. Chainlink, "Sustainable Oracle Economics," Chainlink Blog. Available: https://blog.chain.link/sustainable-oracle-economics/

25. G. Wood, Ethereum: A Secure Decentralised Generalised Transaction Ledger, Ethereum Project Yellow Paper, 2014.

26. S. B. Far, A. I. Rad, and M. R. Asaar, "Blockchain and its derived technologies shape the future generation of digital businesses: a focus on decentralized finance and the Metaverse," Data Science and Management, vol. 6, no. 3, pp. 183–197, 2023. doi: https://doi.org/10.1016/j.dsm.2023.06.002

27. Cimphony AI, "Smart Contract Oracles: A Complete Guide (2024)," Cimphony AI.

Available: https://www.cimphony.ai/insights/smart-contract-oracles-complete-guide-2024

28. J. Anglen, "Blockchain Oracles: Essential Guide to Connecting On-Chain and Off-Chain Data," Rapid Innovation. Available: https://www.rapidinnovation.io/post/blockchain-oracles-essential-guide-connecting-on-chain-off-chain-data