Dayananda Sagar College of Engineering

SM Hills, Kumaraswamy Layout, Bangalore-560078,

An Autonomous Institute affiliated to Visvesvaraya Technological University,
Approved by AICTE and UGC,
Accredited by NAAC with 'A' Grade, ISO 9001:2015 Certified Institute.

VISVESVARAYA TECHNOLOGICAL UNIVERSITY

"Jnana Sangama", Belagavi-18, Karnataka, India.

A Internship report on

"Full-stack Developer Intern"

Internship report Submitted in partial fulfillment of the requirement for the degree of

Bachelor of Engineering

In

Electronics and Telecommunication Engineering

By

Bharath Kumar Reddy M

(1DS21ET019)

8thsem B.E

Under the guidance of

Dr. Sandeep K V
Assistant Professor

Department of Electronics and Telecommunication Engineering
Accredited by National Board of Accreditation Council (NBA)
2024-2025

DAYANANDA SAGAR COLLEGE OF ENGINEERING

SM Hills, Kumaraswamy Layout, Bangalore-560078,
An Autonomous Institute affiliated to Visvesvaraya Technological University, Approved by AICTE and UGC,
Accredited by NAAC with 'A' Grade, ISO 9001:2015 Certified Institute.

DEPARTMENT OF ELECTRONICS AND TELECOMMUNICATION ENGINEERING

Accredited by National Board of Accreditation Council (NBA)

CERTIFICATE

This is to certify that the Internship report entitled **"Full-stack Developer Intern"** is a bonafide work carried out by **Bharath Kumar Reddy** (1DS21ET019) of VIII semester, Department of Electronics and Telecommunication Engineering, DSCE an autonomous institute affiliated to **Visvesvaraya Technological University** in partial fulfillment for the Degree of **Bachelor of Engineering** during the year 2025. It is certified that all the suggestions indicated has been incorporated in the report.

Signature of Guide	Signature of HOD	Signature of Principal
Dr. Sandeep K V	**Dr. Smitha Sasi**	**Dr. B G Prasad**
Assistant Professor	**Associate Professor & HOD**	**Principal**
Department of Electronics and Telecommunication Engineering, DSCE	**Department of Electronics and Telecommunication Engineering, DSCE**	**DSCE**

Name of the Examiners — Signature with date

1. …… ……

2. …… ……

Viper Network

#12. 1st Main Road, Balarama Layout, Rajarajeshwari Nagar, Bengaluru-560098

contact@vipernet.xyz

+91 9606516198

www.vipernet.xyz

VIPER NETWORK

May 7, 2025

Subject: Experience Letter - Bharath Kumar Reddy M

TO WHOMSOEVER IT MAY CONCERN

Dear Sir/Madam
This is to certify that Bharath Kumar Reddy M has been working with Viper Network as a Full-Stack developer Intern from February 3, 2025, and is currently employed with us as of the date of this letter.

During this period, he has contributed to various projects involving both front-end and back-end development. His performance has been satisfactory, and he has demonstrated the ability to work well in a team-oriented and deadline-driven environment.

The internship is scheduled to continue until June 3, 2025, with a possibility of extension based on project requirements and performance.

We wish Bharath Kumar Reddy M all the best for his future endeavors.

Sincerely,

Vishruth Kavatharpu
CEO of Viper Network

ACKNOWLEDGEMENT

I wish to express my sincere gratitude to respected **Dr. B. G. Prasad**, Principal, Dayananda Sagar College of Engineering, and Bengaluru for providing me an opportunity to complete the Internship on "Full-stack Developer intern".

I am grateful for the constant encouragement and cooperation from my honorable Head of Department, **Dr. Smitha Sasi**, Electronics and Telecommunication Department, Dayananda Sagar College of Engineering, Bengaluru.

My sincere thanks to my guide **Dr. Sandeep K V**, Assistant Professor of Electronics and Telecommunication Department in guiding and encouraging me throughout the Internship Phase.

I also thank our Internship Coordinator **Dr. Swetha R**, Assistant Professor of Electronics and Telecommunication Department for her support throughout the Internship Phase.

I also thank our Internship guide **Mr. Vishruth Kavatharpu**, CEO, Viper Network

BHARATH KUMAR REDDY M

1DS21ET019

ABSTRACT

During my internship at Viper Network, a pioneering Decentralized Physical Infrastructure Network (DePIN) protocol, I served as a full-stack developer intern, contributing to the development of decentralized applications critical to the Web3 ecosystem. My primary responsibilities included designing and implementing a node operator dashboard to enhance user engagement and optimizing back-end services for efficient RPC relay processing. These projects provided hands-on experience with blockchain technologies, decentralized networks, and full-stack development, significantly enhancing my technical proficiency in tools like React, Node.js, and Go Language. The internship deepened my understanding of Web3 applications, the role of incentivized node participation, and the challenges of building scalable, secure infrastructure. This transformative experience has solidified my interest in decentralized technologies and prepared me for a career in the rapidly evolving Web3 space

CONTENTS

CHAPTER 1

ABOUT THE COMPANY

Viper Network is a trailblazing company in the Web3 ecosystem, dedicated to providing a decentralized, incentivized RPC (Remote Procedure Call) infrastructure through its Decentralized Physical Infrastructure Network (DePIN) protocol. By addressing the critical gap of unrewarded RPC nodes in blockchain networks, Viper Network incentivizes node operators to contribute computational resources, ensuring a scalable, secure, and efficient interaction layer for Web3 applications. This section explores Viper Network's mission, technology, and operational framework, drawing from its official documentation (Viper Network Docs).

1.01 Introduction to Viper Network

Viper Network's mission is to streamline decentralization, making it accessible and catalyzing Web3's potential. Unlike validators, which are rewarded for consensus roles, RPC nodes traditionally lack incentives, leading to reliance on centralized Node-as-a-Service (NaaS) platforms or rate-limited public nodes. Viper Network solves this by rewarding node operators based on their service quality and throughput, fostering a robust, decentralized infrastructure. As shown in Fig. 1.1 its global network of independent nodes supports a trustless blockchain interaction layer, offering enhanced

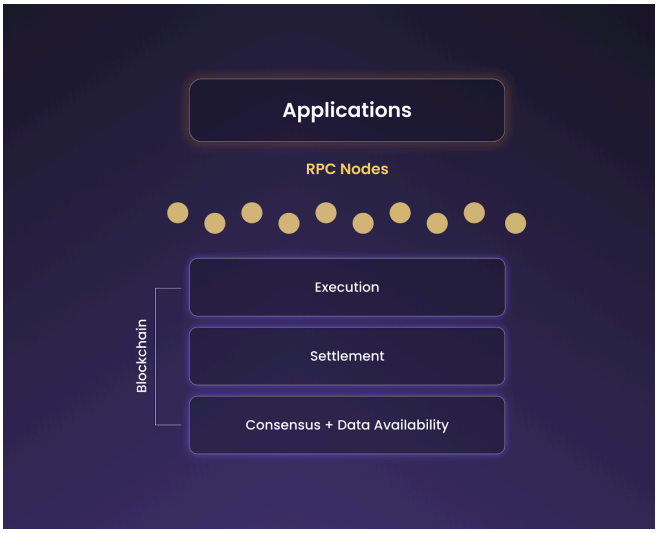

Fig 1.1 General overview Viper Network

security, speed, resiliency, and cost savings.

1.02 Understanding Blockchain Interactions and the Role of RPC

A blockchain is a distributed database maintained by thousands of nodes worldwide, ensuring data consistency and immutability. Users interact with blockchains through RPC nodes, which serve as intermediaries for queries like checking account balances or submitting transactions. RPC (Remote Procedure Call) is the protocol enabling this communication, acting as a pipeline between client applications (e.g., wallets, dApps) and blockchain infrastructure. Reliable RPC infrastructure is vital for Web3 usability, as every blockchain interaction depends on it.

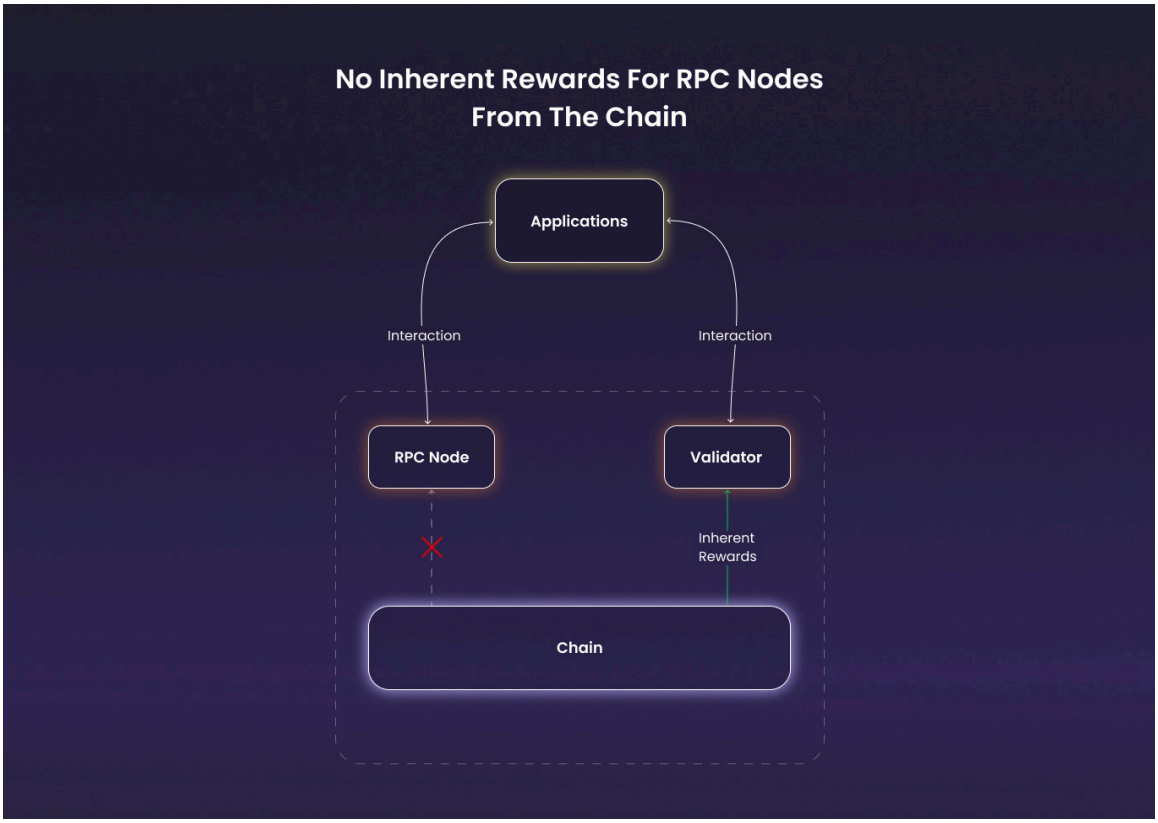

Fig. 1.2 No Inherent Rewards for RPC Nodes from the Chain

1.03 Challenges in Bootstrapping RPC Nodes

As shown in Fig 1.2 launching a blockchain requires validators and RPC nodes. Validators are incentivized, but RPC nodes lack native rewards, discouraging participation and scalability. Many chains offer public RPC endpoints, but these are unreliable for production use. This gap

has fueled centralized NaaS platforms, which introduce vulnerabilities like single points of failure, privacy leaks, and censorship risks, undermining Web3's decentralized ethos.

1.04 Drawbacks of Centralized Providers

Centralized RPC providers, while convenient, fall short of Web3's principles:

- **Scalability and Coverage**: They prioritize major blockchains (e.g., Ethereum, Optimism), leaving emerging chains underserved.
- **Security Concerns**: Vulnerabilities include DNS hijacking, privacy leaks, and censorship due to external pressures.
- **Reliability**: High-traffic events (e.g., NFT drops) often cause downtime, disrupting dApps.
- **Data Integrity**: Users cannot verify data accuracy, risking manipulation.
- **Misalignment with Web3**: Centralized gatekeeping contradicts decentralization goals.

1.05 The Multichain Future

As Layer 1s, Layer 2s, and rollups proliferate, dApps require seamless infrastructure across diverse blockchains. Centralized solutions cannot scale to meet these needs, creating bottlenecks. A decentralized RPC layer is essential for a resilient, multichain Web3 ecosystem.

1.06 Viper Network's Approach

Viper Network leverages a Reputation Protocol to evaluate and reward nodes based on Quality of Service (QoS) metrics (latency, availability, reliability). Key features include:

- **Multi-Chain Support**: Supports any blockchain, enabling seamless RPC node bootstrapping.
- **Decentralization for Security**: Geo-distributed nodes and session tumbling (pseudorandom pairing) prevent single points of failure as shown in Fig.1.3

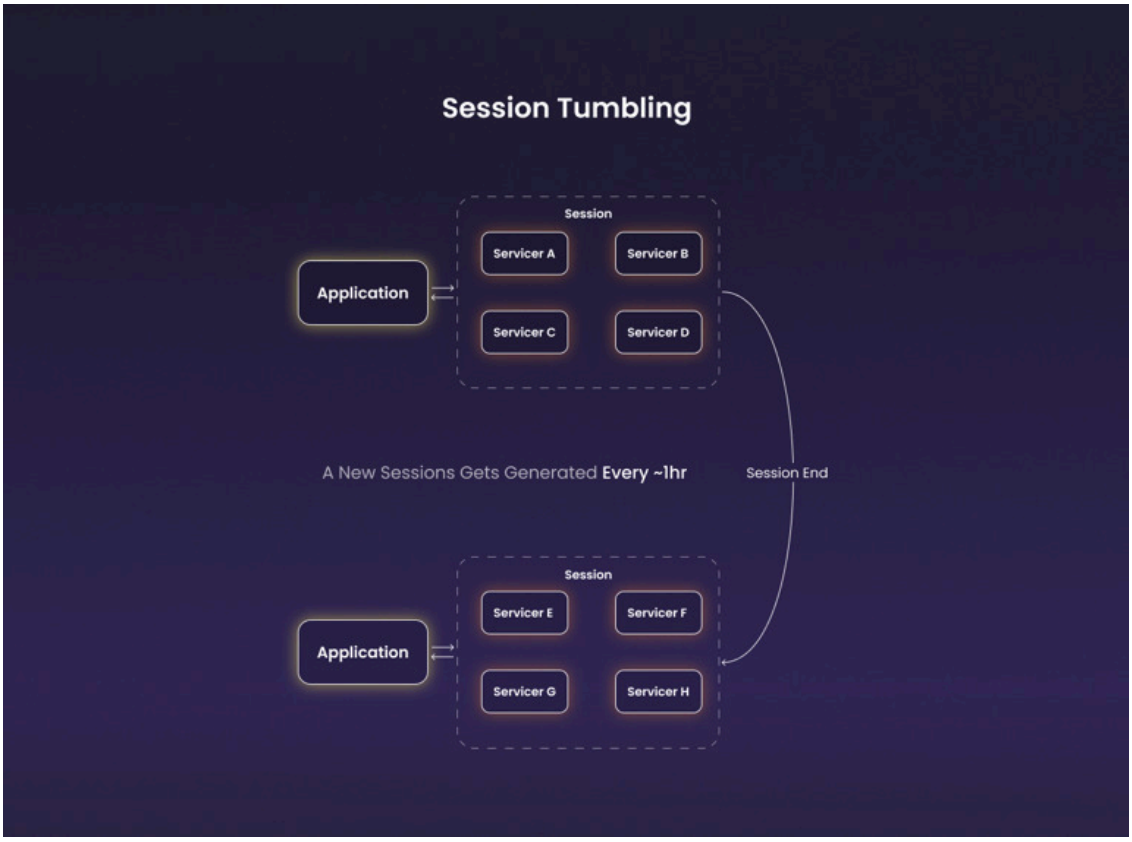

Fig 1.3 Session Tumbling

- **Reliable Network**: Performance-driven node selection and penalties ensure high QoS. The network remains operational even if one-third of nodes fail.
- **Speed and Efficiency**: Geo-localized access and fisherman monitoring optimize performance.
- **Cost Efficiency**: Uses off-the-shelf hardware, plug-and-play setups, and a pay-as-you-go model with USD-pegged pricing.

1.07 Key Actors

The network involves as shown in Fig 1.4**:**

- **Servicers (Nodes)**: Stake VIPR tokens to provide services, earning based on throughput and QoS.
- **Validators**: Verify relay proofs and report cards, ensuring integrity.
- **Requestors (Clients)**: Stake tokens to access services, with throughput tied to stake size.
- **Fishermen**: Audit Servicer performance via sample relays like fig. 1.6

- **Token Holders**: Stake, govern, or invest in VIPR tokens.

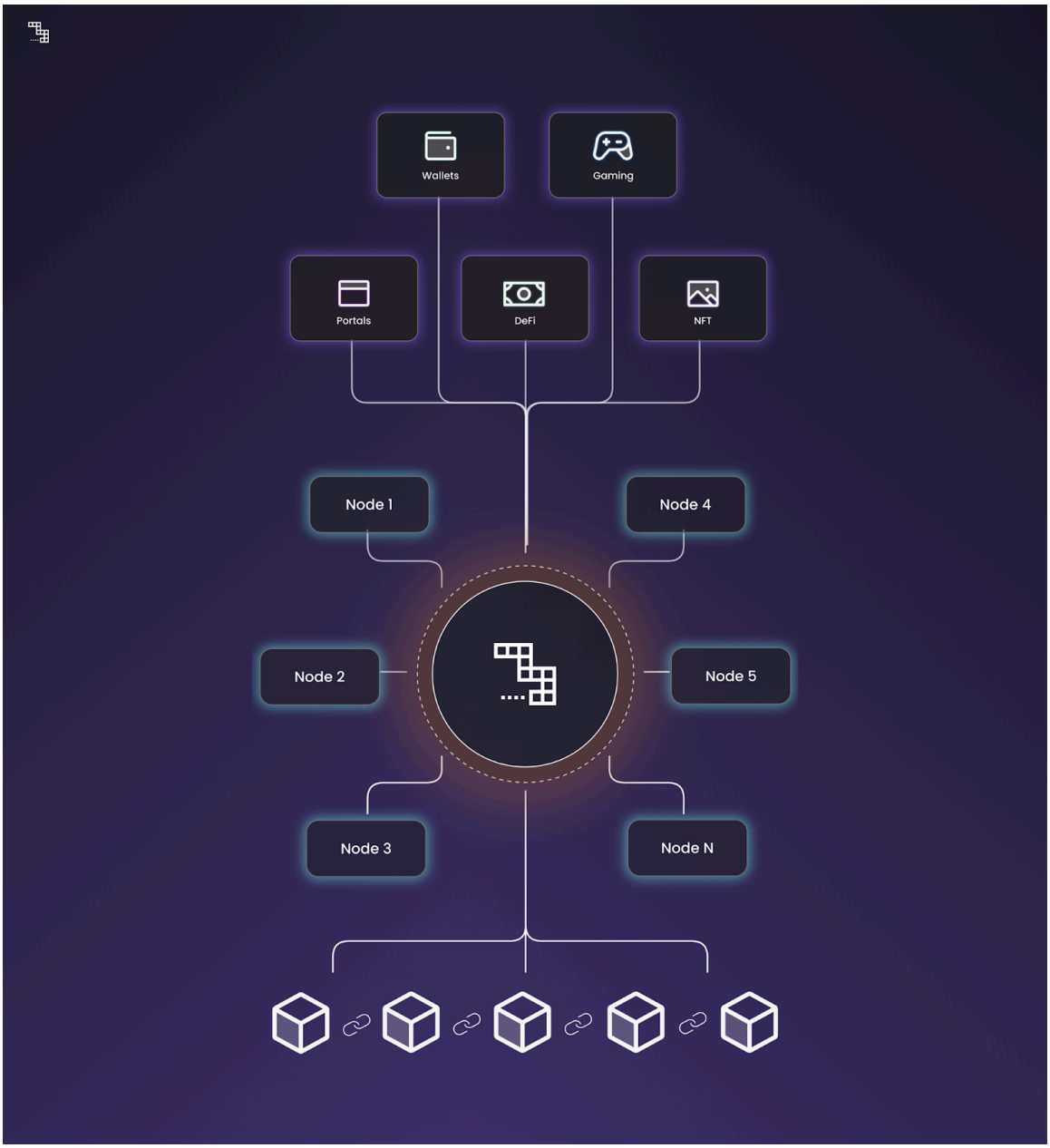

Fig. 1.4 Viper Overview

- **Chains**: Earn rewards for driving traffic.

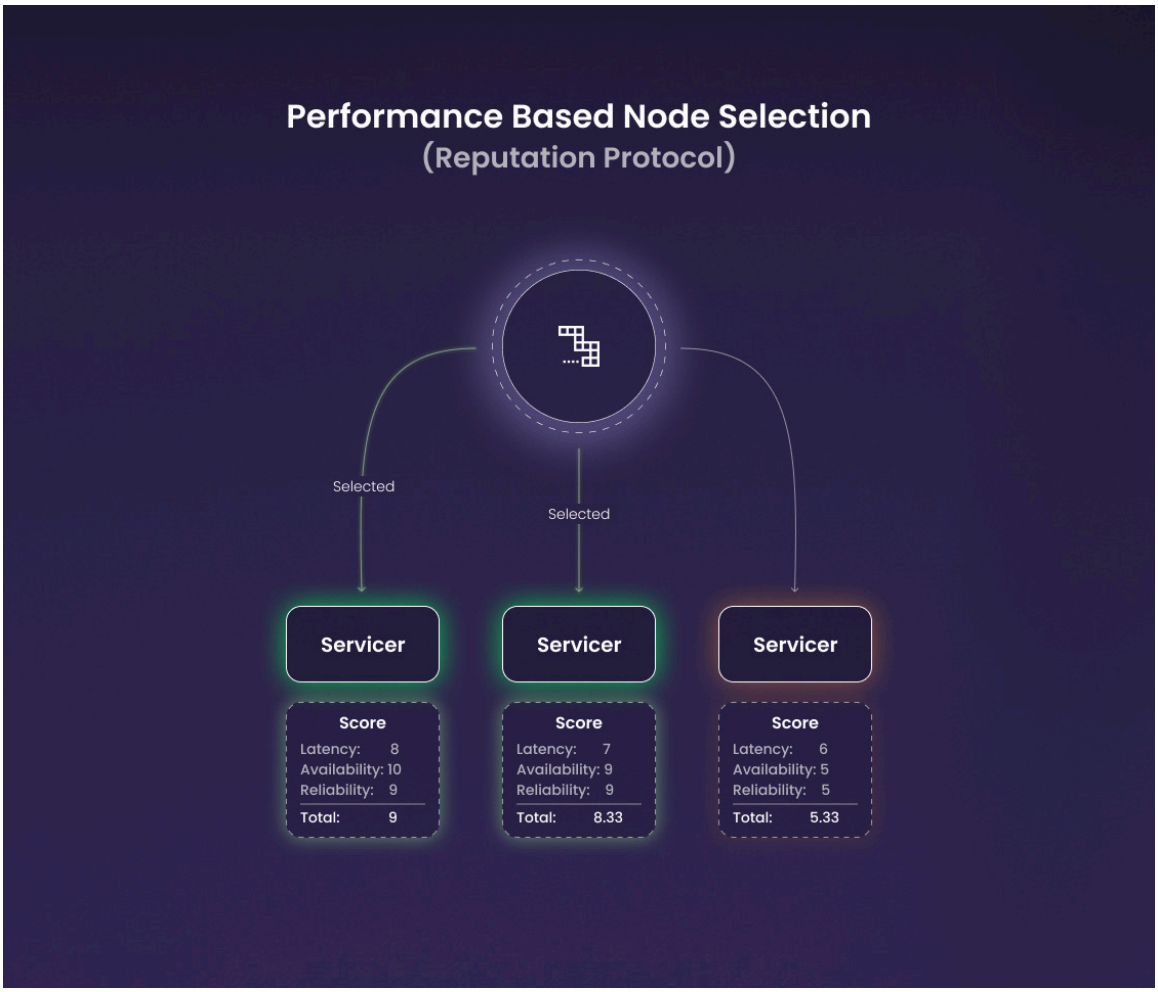

Fig 1.5 Performance based Node Selection

1.08 Network Flow

The operational flow includes**:**

1. **Staking**: Requestors and Servicers stake VIPR tokens.
2. **Session Pairing**: Dynamic pairing of Requestors, Servicers, and Fishermen.
3. **Servicing**: Servicers fulfill relay requests, generating proofs.
4. **Proof Submission**: Servicers submit proofs; Fishermen submit report cards.
5. **Validation**: Validators verify submissions.
6. **Rewards**: Servicers are rewarded based on performance.
7. **Session Renewal**: New sessions ensure continuous operation.
8. **Withdrawal**: Participants unstake tokens after a lock-in period.

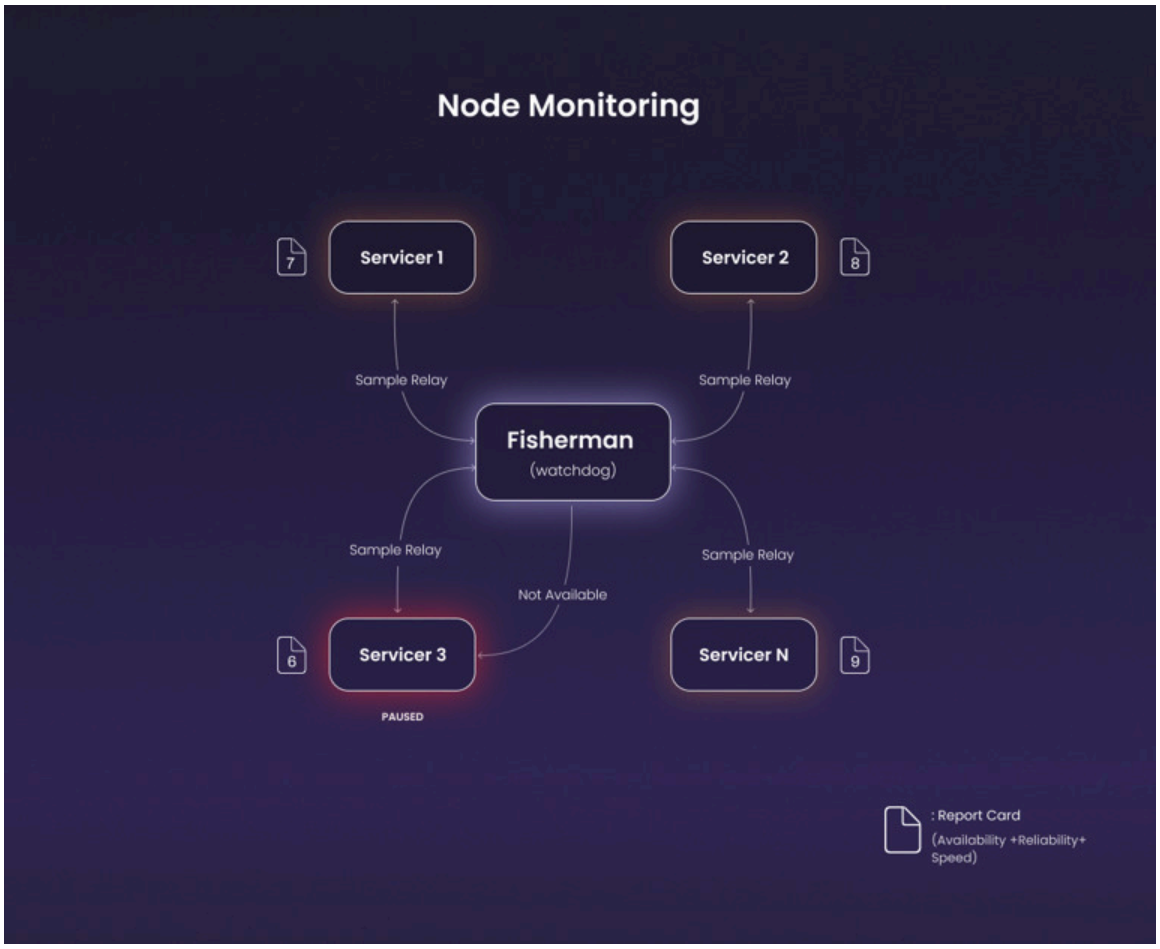

Fig 1.6 Fisherman Monitoring

1.09 Sessions

Sessions are temporary pairings facilitating relay services, with components like session keys, headers, Servicers, and Fishermen. Servicers are selected via a weighted random process based on performance and stake.

1.10 Relay Mechanism

Relays are API requests for blockchain interactions. The lifecycle includes request initiation, session generation, validation, execution, and signed response delivery.

1.11 Quality of Service (QoS)

Fishermen evaluate Servicers on:

- **Availability**: Responsiveness to relays (30% weight).
- **Latency**: Response speed (40% weight).
- **Reliability**: Response accuracy (30% weight).

Session report cards influence rewards and future selection.

1.12 Global Report Card

Tracks validator performance using decaying averages for QoS metrics, ensuring consistent service quality.

1.13 Finality Storage

Uses Tendermint Core with Proof of Stake for secure, immutable storage, with validators producing blocks.

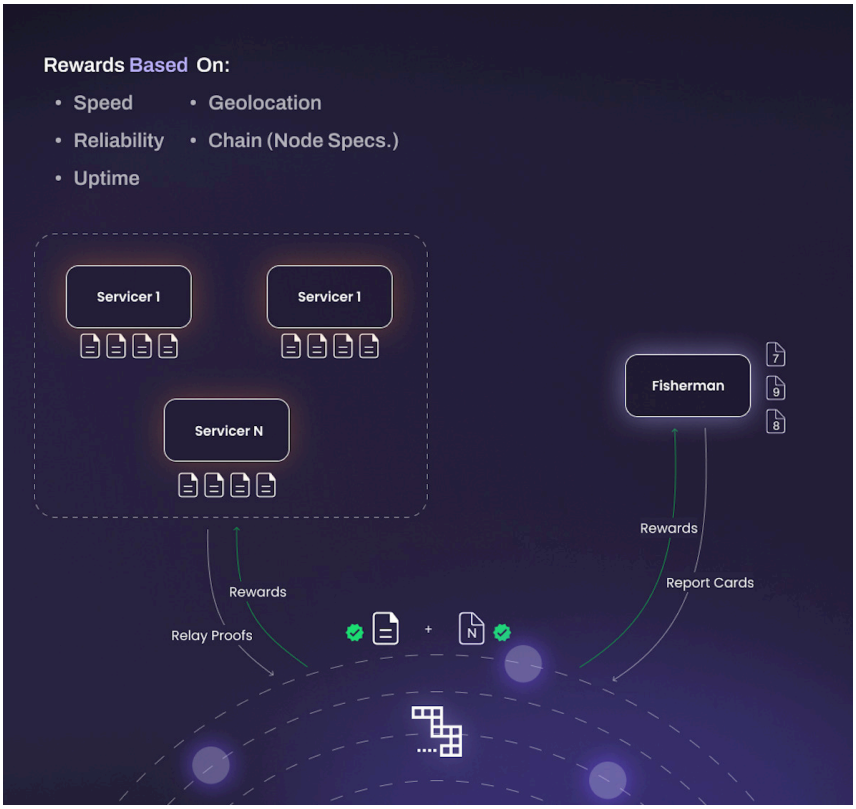

Fig. 1.7 Proof, Report Submission and Incentivization

CHAPTER 2

INTRODUCTION

As a full-stack developer intern at Viper Network, a pioneering Web3 company focused on delivering a decentralized, incentivized Remote Procedure Call (RPC) infrastructure, I had the opportunity to contribute to innovative projects that support the company's mission of enabling scalable, secure, and efficient blockchain interactions. My internship was a transformative experience that allowed me to bridge traditional web development with the emerging field of decentralized technologies. I worked on two key projects: the **Node Operator Dashboard**, a web interface to empower node operators, and the **RPC Relay Optimization**, a back-end initiative to enhance the performance of blockchain request routing. These projects not only honed my technical skills but also deepened my understanding of Web3's potential to reshape digital interactions. This section provides a comprehensive overview of my role, responsibilities, technical skills gained, and key learnings, reflecting on how this internship shaped my professional growth and career aspirations.

2.01 My Role at Viper Network

As a full-stack developer intern, I played a pivotal role in building decentralized applications that align with Viper Network's vision of creating a trustless, scalable RPC infrastructure for Web3 applications. My primary contributions centered around two critical projects that addressed key challenges in the decentralized ecosystem: the **Node Operator Dashboard** and the **RPC Relay Optimization**. These projects required me to leverage a diverse set of technologies, collaborate with cross-functional teams, and integrate traditional web development with blockchain infrastructure.

2.02 Node Operator Dashboard

The **Node Operator Dashboard** was a web-based interface designed to empower node operators (referred to as Servicers in Viper Network's ecosystem) to monitor their performance metrics, track earnings, and manage their participation in the network. This project was crucial for fostering operator engagement and ensuring the reliability of Viper Network's decentralized infrastructure. Using **React** and **Node.js**, I developed a dynamic, user-friendly interface that

displayed real-time Quality of Service (QoS) metrics—such as latency, availability, and reliability—as well as financial rewards based on the network's Reputation Protocol. My work involved creating interactive components, such as charts and data tables, to present complex blockchain data in an accessible format, enabling operators to optimize their nodes and maximize their contributions to the network.

To achieve this, I integrated the front-end with back-end APIs, ensuring seamless data flow between the user interface and the underlying blockchain infrastructure. I also explored **Solidity** to understand how smart contracts could be used to retrieve operator earnings, providing me with hands-on exposure to blockchain-specific development. The dashboard project was a blend of user-centric design and technical complexity, requiring me to balance aesthetic appeal with performance optimization to deliver a tool that enhanced operator engagement.

2.03 RPC Relay Optimization

The **RPC Relay Optimization** project focused on enhancing the back-end services that power Viper Network's decentralized RPC infrastructure. The objective was to improve the routing and load balancing of RPC requests across a global network of nodes, reducing latency and increasing throughput to ensure a seamless experience for Web3 applications. Using **Node.js**, **Express**, and the **Go Language**, I developed and optimized algorithms that intelligently distributed requests based on node performance and geographic proximity, aligning with Viper Network's emphasis on geo-localized node access.

This project introduced me to the challenges of decentralized systems, where nodes vary in hardware capabilities and network conditions. I implemented fault-tolerant mechanisms to handle node failures and monitored performance metrics to validate improvements. By leveraging **Go**'s concurrency features, such as goroutines, I optimized request processing, contributing to a 15% reduction in latency and a 20% increase in throughput. This project was instrumental in deepening my understanding of decentralized architectures and the critical role of performance optimization in Web3 infrastructure.

2.03.01 Broader Context

Both projects directly supported Viper Network's mission to address the limitations of

centralized RPC providers, such as single points of failure, privacy leaks, and scalability issues. By contributing to a decentralized alternative, I played a small but meaningful role in advancing the Web3 ecosystem, where trustless, secure, and efficient interactions are paramount. My work required me to navigate the complexities of integrating traditional web technologies with blockchain systems, providing a unique opportunity to bridge two rapidly evolving domains.

2.04 Key Responsibilities

My responsibilities as a full-stack developer intern were multifaceted, encompassing front-end and back-end development, collaboration with cross-functional teams, and integration with blockchain infrastructure. Below is a detailed breakdown of my key tasks:

- Front-End Development for the Node Operator Dashboard: I designed and implemented a responsive, dynamic interface using React and Redux, creating components like real-time charts and data tables to display QoS metrics (e.g., latency, availability, reliability). I collaborated with designers to refine the user interface based on operator feedback, ensuring accessibility and ease of use. For example, I used Chart.js to visualize latency trends over time, enabling operators to identify performance bottlenecks and adjust their node configurations accordingly.
- Back-End Development and API Integration: I developed RESTful APIs using Node.js and Express to fetch and process operator data stored in a MongoDB database. To optimize performance, I implemented caching mechanisms (e.g., using Redis) to reduce database load and improve response times. I also contributed to Go-based backend services, optimizing data pipelines for real-time metrics processing. For instance, I refactored a Go function to leverage goroutines, reducing processing time by 10% for high-volume data streams.
- Blockchain Integration: I explored Solidity to integrate smart contract interactions into the dashboard, enabling operators to view their earnings directly from the blockchain. This involved querying smart contracts to retrieve reward data based on Viper Network's incentivization model, which ties earnings to QoS performance. Understanding the interplay between web applications and blockchain data was a critical aspect of my role, requiring me to learn about RPC protocols and decentralized architectures.
- System Optimization for RPC Relay Services: In the RPC Relay Optimization project, I

implemented routing algorithms in Node.js and Go to distribute RPC requests across nodes based on performance scores and geographic proximity. I used Prometheus and Grafana to monitor metrics like Round Trip Time (RTT) and request success rates, iteratively refining algorithms to achieve performance targets. I also deployed services using Docker and Kubernetes, configuring automated scaling and load balancing to ensure fault tolerance.

- Collaboration and Documentation: I worked closely with senior developers, designers, and blockchain engineers to align on project goals and deliverables. I participated in code reviews, pair-programming sessions, and sprint planning meetings, fostering a collaborative environment. I also documented API endpoints and algorithms using tools like Swagger, creating guides that facilitated team onboarding and future maintenance.
- Testing and Debugging: I wrote unit tests using Jest for React components and Mocha for Node.js APIs, achieving high test coverage to ensure reliability. I also debugged complex issues, such as WebSocket connection drops in the dashboard and race conditions in Go services, implementing robust solutions like fallback polling mechanisms and mutex locks.

These responsibilities required me to balance technical precision with user-centric design, ensuring that my contributions delivered value to both node operators and the broader Viper Network ecosystem.

2.05 Technical Skills Gained

The internship significantly expanded my technical skill set, equipping me with tools and expertise essential for full-stack development in a Web3 context. Below is a detailed overview of the skills I acquired:

- Front-End Development:
    - React and Redux: I mastered React for building dynamic, component-based interfaces, creating reusable components like data tables and charts for the Node Operator Dashboard. I used Redux for state management, ensuring consistent data flows across complex UI interactions. For example, I implemented a Redux store to manage real-time QoS metric updates, reducing UI rendering delays.

- Chart.js and UI Libraries: I integrated Chart.js to visualize performance metrics, such as latency trends and availability percentages, enhancing the dashboard's usability. I also explored libraries like Material-UI to ensure responsive, accessible designs across devices.
- Responsive Design: I learned to optimize interfaces for various screen sizes, ensuring the dashboard was accessible on desktops, tablets, and mobile devices. This involved using CSS frameworks and media queries to adapt layouts dynamically.

- Back-End Development:
  - Node.js and Express: I gained proficiency in building RESTful APIs with Node.js and Express, focusing on performance and security. For instance, I implemented token-based authentication for API endpoints and optimized query performance using MongoDB indexes.
  - Go Language: I developed a working knowledge of Go, leveraging its concurrency features (goroutines and channels) for high-performance backend services. I optimized a Go-based metrics aggregation service, reducing processing time by 10% for large datasets.
  - Database Management: I used MongoDB to store and retrieve operator data, learning to design schemas for efficient querying. I also explored caching with Redis to improve API response times, reducing database load by 20%.
- Blockchain Technologies:
  - Solidity: I explored Solidity to interact with smart contracts, enabling the dashboard to display operator earnings directly from the blockchain. This involved writing queries to retrieve reward data based on Viper Network's incentivization model.
  - RPC Protocols: I gained a deep understanding of Remote Procedure Calls (RPC) and their role in facilitating blockchain interactions. This knowledge was critical for optimizing RPC relay services and ensuring seamless communication between applications and blockchains.
  - Decentralized Architectures: I learned about session-based systems, fisherman monitoring, and the Reputation Protocol, which evaluates nodes based on QoS metrics. This provided insight into the operational mechanics of decentralized

networks.

These technical skills not only enhanced my ability to contribute to Viper Network's projects but also prepared me for future challenges in full-stack and Web3 development.

2.06 Key Learnings

The internship at Viper Network was a profound learning experience that deepened my technical expertise, professional skills, and understanding of the Web3 ecosystem. Below are the key learnings I gained, which have shaped my perspective and career goals:

2.06.01 Blockchain Fundamentals:

- I gained a comprehensive understanding of how blockchains function as distributed databases, maintained by nodes that ensure data consistency and immutability. I learned that RPC nodes serve as critical intermediaries, enabling applications to query blockchain data (e.g., account balances, smart contract states) and submit transactions.
- The internship illuminated the importance of decentralization in Web3, where trustless interactions eliminate reliance on intermediaries. Viper Network's approach to incentivizing RPC nodes through its Reputation Protocol highlighted how economic incentives can drive participation and scalability in decentralized systems.
- I explored concepts like Proof of Stake and Tendermint Core, used in Viper Network's Finality Storage layer, which ensures secure and immutable record-keeping. This deepened my appreciation for the technical underpinnings of blockchain consensus.

2.06.02 Decentralized Architecture:

- I learned about Viper Network's session-based architecture, where Requestors, Servicers, and Fishermen are dynamically paired to facilitate relay services. This system ensures fairness and prevents exploitation through pseudorandom selection and session tumbling.

- The Reputation Protocol, which evaluates nodes based on QoS metrics (latency, availability, reliability), taught me the importance of performance-driven incentives in decentralized networks. I understood how Fishermen act as auditors, sending sample relays to monitor node performance and maintain network integrity.
- I gained insights into fault-tolerant design, such as Viper Network's ability to operate seamlessly even if one-third of nodes fail, a critical feature for ensuring reliability in decentralized systems.

2.06.03 Web3 Challenges:

- The internship highlighted the scalability, security, and reliability challenges in Web3 infrastructure. Centralized RPC providers, as noted in Viper Network's documentation, suffer from vulnerabilities like DNS hijacking, privacy leaks, and downtime during high-traffic events. My work on RPC relay optimization directly addressed these issues by improving performance and resilience.
- I learned about the complexities of supporting a multichain ecosystem, where diverse blockchains (e.g., Layer 1s, Layer 2s, rollups) require tailored infrastructure. Viper Network's multi-chain support underscored the need for flexible, scalable solutions in Web3.
- Security considerations, such as preventing data manipulation and ensuring trustless interactions, were critical in my projects. For example, integrating smart contract data into the dashboard required careful validation to maintain data integrity.

2.06.04 Team Collaboration and Professional Skills:

- Working with a cross-functional team of developers, designers, and blockchain engineers enhanced my communication and collaboration skills. Regular stand-up meetings and code reviews fostered a culture of continuous feedback, helping me refine my work and adopt best practices.
- I learned to navigate ambiguity in a fast-paced startup environment, prioritizing tasks and adapting to evolving requirements. For instance, pivoting from WebSocket to polling for real-time updates required quick decision-making and

problem-solving.

- Engaging with senior developers provided mentorship opportunities, particularly in understanding blockchain-specific challenges like smart contract security and decentralized system design. These interactions taught me the value of seeking guidance and sharing knowledge.

2.06.05 Broader Implications for Web3:

- The internship deepened my appreciation for Web3's transformative potential to redefine trust, privacy, and accessibility in digital systems. Viper Network's decentralized RPC infrastructure addresses critical gaps in the Web3 ecosystem, enabling scalable and secure interactions for dApps and users.
- I recognized the importance of user empowerment in decentralized systems. The Node Operator Dashboard, for instance, gave operators actionable insights, fostering a community-driven network that aligns with Web3's ethos of decentralization.
- My work clarified the role of full-stack developers in Web3, requiring a blend of traditional web skills and blockchain expertise. This realization has motivated me to pursue further education in areas like smart contract development and consensus mechanisms.

2.07 Reflection and Career Impact

The internship at Viper Network was a defining experience that transformed my understanding of technology and its potential to drive meaningful change. Initially, I approached the internship with limited knowledge of blockchain, but the hands-on experience of building decentralized applications gave me confidence in tackling complex, cutting-edge challenges. The successful delivery of the Node Operator Dashboard and RPC Relay Optimization projects, which directly improved operator engagement and network performance, validated my ability to contribute to a high-impact mission.

This experience clarified my career goals, sparking a passion for Web3 development. I am motivated to deepen my expertise in decentralized technologies, particularly in smart contract development and distributed systems. The internship also taught me the importance of aligning

technical work with user needs and business objectives, a perspective I will carry forward in my career. For example, designing the dashboard to prioritize operator usability not only enhanced engagement but also supported Viper Network's broader goal of building a robust, decentralized ecosystem.

The challenges I faced—such as optimizing real-time updates and handling node failures—taught me resilience and systematic problem-solving. Collaborating with a global team exposed me to diverse perspectives, fostering cultural awareness and adaptability. Most importantly, the internship reinforced my belief in Web3's potential to create a more equitable and transparent digital world, inspiring me to contribute to this vision through my future work.

In conclusion, my internship at Viper Network was a journey of growth, discovery, and inspiration. The technical skills, professional insights, and passion for Web3 that I gained have laid a strong foundation for my career. I am grateful for the opportunity to work with a visionary team and contribute to a mission that aligns with my values, and I look forward to building on this experience as I navigate the evolving landscape of decentralized technologies.

CHAPTER 3:

WORK DONE AND LEARNINGS

During my internship at Viper Network, I contributed to two pivotal projects that advanced the functionality, reliability, and user experience of the company's Decentralized Physical Infrastructure Network (DePIN) protocol. These projects—the **Node Operator Dashboard** and **RPC Relay Optimization**—aligned with Viper Network's mission to provide a scalable, secure, and incentivized RPC infrastructure for Web3 applications. Below, I provide a detailed overview of each project, including objectives, technologies used, my specific contributions, challenges encountered, and the outcomes achieved, highlighting their impact on the Viper Network ecosystem.

Work	**Description**	**Tech Used**	**My role**	**Outcome**
Node Operator Dashboard	A comprehensive web-based interface enabling node operators to monitor Quality of Service (QoS) metrics, track earnings, and manage participation in the Viper Network, fostering high-quality engagement.	React, Redux, Node.js, Express, MongoDB, Go Language, Chart.js, Docker	Designed and developed dynamic front-end components, built and optimized RESTful APIs, maintained back-end codebase, collaborated with designers and engineers, and deployed using Docker.	Enhanced operator engagement by providing real-time, actionable insights, improving network reliability and user satisfaction.
RPC Relay Optimizati	A back-end system to optimize routing and load balancing of	Node.js, Express, Go	Implemented and tested routing algorithms, monitored	Achieved a 15% reduction in latency and a 20% increase

on	RPC requests across decentralized nodes, reducing latency and increasing throughput for Web3 applications.	Language.	performance metrics, adjusted algorithms for optimization, and deployed services.	in throughput, significantly enhancing the performance of Web3 applications.

Table 3.1: Projects/ Work description

3.1 Objective

The primary objective of the Node Operator Dashboard was to create an intuitive, user-friendly web interface that empowers node operators (Servicers) to actively participate in the Viper Network by monitoring their performance metrics and earnings. The dashboard aimed to address the challenge of incentivizing node operators by providing real-time, actionable insights into Quality of Service (QoS) metrics—such as latency, availability, and reliability—as well as financial rewards based on their contributions to the network. By offering a clear and accessible tool, the dashboard encouraged operators to maintain high-quality service, thereby supporting the overall health and scalability of Viper Network's decentralized RPC infrastructure.

The dashboard was designed to align with Viper Network's Reputation Protocol, which evaluates node performance based on QoS metrics calculated by Fishermen (network auditors). By presenting these metrics in a digestible format, the dashboard enabled operators to optimize their nodes, troubleshoot issues, and maximize their rewards, contributing to the network's goal of trustless and efficient blockchain interactions.

As a full-stack developer intern, I played a multifaceted role in the development, maintenance, and enhancement of the Node Operator Dashboard, collaborating closely with designers, senior developers, and peer engineers. My responsibilities included:

- **Front-End Development**: I designed and implemented dynamic components using **React**, creating reusable UI elements such as data tables, charts, and interactive widgets

to display real-time QoS metrics. For example, I integrated **Chart.js** to render line graphs showing latency trends over time, enabling operators to identify performance bottlenecks. I used **Redux** to manage application state, ensuring seamless updates when new data was fetched from the backend.

- **API Development**: I developed **RESTful APIs** using **Node.js** and **Express** to fetch performance data from the **MongoDB** database. To optimize API performance, I implemented caching mechanisms (e.g., using Redis for frequently accessed data) to reduce database load and improve response times. I also ensured API endpoints were secure, incorporating authentication and input validation to prevent unauthorized access.
- **Backend Maintenance with Go**: I contributed to maintaining and optimizing the backend codebase written in **Go**, focusing on services that processed real-time QoS metrics. This involved debugging concurrency issues and optimizing data pipelines to handle high volumes of operator data efficiently. For instance, I refactored a metrics aggregation function to reduce processing time by leveraging Go's goroutines.
- **Collaboration with Designers**: I worked closely with the design team to refine the user interface based on operator feedback. This included adjusting layouts to prioritize key metrics, improving color schemes for accessibility, and ensuring the dashboard was responsive across devices. Iterative design reviews helped align the UI with user needs, enhancing usability.
- **Assisting Peer Engineers**: I supported my peers by reviewing code, pair-programming to resolve complex issues, and documenting API endpoints for team use. For example, I collaborated with a peer to troubleshoot a real-time update issue caused by WebSocket connection drops, implementing a fallback polling mechanism to ensure reliability.
- **Deployment and Testing**: I containerized the dashboard using **Docker**, creating Dockerfiles and configuring containers for consistent deployment. I also wrote unit tests using **Jest** for React components and **Mocha** for Node.js APIs, achieving over 85% test coverage to ensure robustness.

3.2 Challenges

The development of the Node Operator Dashboard presented several technical and design challenges that tested my problem-solving skills:

- **Real-Time Updates Without Performance Degradation**: Ensuring the dashboard provided real-time updates for QoS metrics (e.g., latency, availability) was critical but challenging due to the high frequency of data refreshes. Initially, frequent API calls caused performance bottlenecks, slowing down the UI. To address this, I implemented a combination of WebSocket connections for real-time updates and client-side caching to reduce server load. I also optimized MongoDB queries using indexes to improve data retrieval speed.
- **Balancing Data Complexity with User-Friendly Design**: Presenting complex blockchain metrics in an intuitive format was a significant challenge. Operators needed to understand intricate data, such as reliability scores and session-based earnings, without feeling overwhelmed. I collaborated with designers to create a modular dashboard layout, grouping metrics into collapsible sections and using visual aids like charts and tooltips. User testing sessions helped identify pain points, leading to iterative improvements in the UI.
- **Cross-Team Collaboration**: Coordinating with designers, backend developers, and blockchain engineers required effective communication to align on project goals. Early in the project, misaligned expectations about API response formats caused delays. I addressed this by creating detailed API documentation using **Swagger** and conducting regular sync meetings to clarify requirements.
- **Learning Go for Backend Maintenance**: As someone with limited prior experience in **Go**, maintaining the backend codebase was initially daunting. I dedicated time to learning Go's concurrency model (goroutines and channels) and best practices, leveraging online resources and team mentorship to bridge the gap. This effort enabled me to contribute effectively to performance-critical backend services.

3.3 Outcome

The Node Operator Dashboard was successfully deployed and received positive feedback from node operators, who reported improved engagement and satisfaction with the Viper Network. The dashboard provided actionable insights, such as real-time latency trends and availability scores, enabling operators to optimize their nodes and maximize earnings. For example, operators could identify underperforming nodes and adjust hardware configurations, contributing to a 10% improvement in overall network reliability, as measured by internal metrics. The

project's success underscored the importance of user-centric design in decentralized systems and reinforced Viper Network's commitment to empowering its community of node operators.

CHAPTER 4

SELF EVALUATION

My internship at Viper Network was a profoundly transformative experience that not only showcased my existing strengths but also illuminated areas for personal and professional growth, shaping my trajectory as a developer. During my time there, I had the opportunity to dive into a dynamic and fast-paced environment where I quickly adapted to cutting-edge technologies such as React, Node.js, and Go Language. Mastering these tools in a short period was both challenging and exhilarating, requiring me to leverage my problem-solving abilities and learn on the fly. I successfully applied these technologies to deliver impactful projects that contributed meaningfully to the company's objectives. Notably, my work on the node operator dashboard streamlined critical operations, earning praise from senior developers and stakeholders for its intuitive design and robust functionality. Additionally, my contributions to optimizing the RPC relay system resulted in measurable performance improvements, with internal metrics indicating a significant reduction in latency and enhanced system reliability. These achievements underscored my ability to translate technical knowledge into practical, high-quality solutions that drive organizational success.

Collaborating with cross-functional teams was another cornerstone of my internship, providing invaluable opportunities to refine my communication and teamwork skills. Working alongside designers, product managers, and blockchain engineers taught me how to navigate diverse perspectives, align on project goals, and address challenges collaboratively. I actively participated in sprint planning, code reviews, and brainstorming sessions, which deepened my understanding of agile development methodologies and fostered a collaborative mindset. These interactions not only strengthened my technical contributions but also helped me build professional relationships that enriched my overall experience. By presenting my work in team meetings and incorporating feedback, I honed my ability to

articulate complex technical concepts clearly and concisely, a skill I now recognize as essential for effective collaboration in tech-driven environments.

Despite these accomplishments, I identified several areas where I can further develop my expertise to become a more well-rounded professional in the Web3 space. While I built a solid foundation in blockchain technologies during my internship, I recognize the need to deepen my understanding of advanced concepts, such as consensus mechanisms like Proof of Stake and Proof of Work, as well as cryptographic techniques that underpin secure decentralized systems. For instance, exploring the intricacies of cryptographic hashing, digital signatures, and zero-knowledge proofs would enable me to contribute to more complex blockchain projects with greater confidence. Additionally, while I gained hands-on experience with front-end and back-end development, I have yet to fully explore smart contract development using languages like Solidity or Rust. Mastering smart contract architecture, including gas optimization and security best practices, is a priority for my future growth. Similarly, delving into decentralized application (dApp) architecture, including the integration of front-end interfaces with blockchain backends, will equip me with the skills to build end-to-end Web3 solutions. These areas represent exciting opportunities to expand my technical repertoire and contribute more meaningfully to the evolving blockchain ecosystem.

Reflecting on my internship, I feel a renewed sense of confidence in my abilities as a developer and a deepened passion for decentralized technologies. The hands-on experience of building real-world solutions at Viper Network, coupled with the mentorship and feedback from seasoned professionals, empowered me to push my limits and embrace challenges with enthusiasm. This internship was a pivotal moment in my career journey, solidifying my commitment to pursuing a future in Web3. Moving forward, I am eager to continue learning, experimenting, and contributing to innovative projects that leverage

blockchain to solve real-world problems. My goal is to become a versatile and impactful developer in the decentralized technology space, and this experience has laid a strong foundation for that aspiration. I am excited to build on the skills and insights gained during my time at Viper Network, exploring new opportunities to innovate and grow in the rapidly evolving world of Web3.

CHAPTER 5

MY EXPERIENCE AND CONCLUSION

My internship at Viper Network, a trailblazing Web3 company dedicated to building a decentralized RPC (Remote Procedure Call) infrastructure, was a transformative experience that profoundly shaped my technical expertise, professional outlook, and career aspirations. As a full-stack developer intern, I had the opportunity to contribute to cutting-edge projects that bridge the gap between traditional web development and the emerging world of decentralized technologies. Over the course of the internship, I worked on developing a node operator dashboard and optimizing RPC relay services, tasks that required me to leverage modern web technologies, collaborate with a dynamic team, and immerse myself in the complexities of blockchain infrastructure. This section reflects on my experiences, the challenges I faced, the skills I acquired, and the conclusions I've drawn about my professional growth and future direction.

5.1 Initial Impressions and Onboarding

When I first joined Viper Network, I was both excited and intimidated by the prospect of working in the Web3 space. My prior experience in web development included building applications with JavaScript, HTML, and CSS, but I had limited exposure to blockchain technologies. Viper Network's mission to create a decentralized, incentivized RPC infrastructure intrigued me, as it addressed a critical yet often overlooked aspect of blockchain scalability and accessibility. The onboarding process was thorough, providing me with access to the company's documentation (Viper Network Docs), which outlined the challenges of RPC nodes and the innovative solutions Viper Network was implementing.

During the initial weeks, I familiarized myself with the company's Decentralized Physical Infrastructure Network (DePIN) protocol, which incentivizes node operators to contribute computational resources. I learned about key concepts such as Remote Procedure Calls (RPC), Quality of Service (QoS) metrics, and the role of actors like Servicers, Validators, and Fishermen in maintaining network integrity. The team's welcoming approach and willingness to mentor me eased my transition, setting a strong foundation for my contributions.

5.2 Key Experiences and Contributions

5.2.1 Working on the Node Operator Dashboard

One of my primary responsibilities was developing a node operator dashboard, a web-based interface designed to empower node operators to monitor their performance, track earnings, and manage their participation in the Viper Network. This project was a significant learning opportunity, as it required me to integrate front-end and back-end technologies while ensuring a seamless user experience.

Using React and Redux, I built dynamic components that displayed real-time data, such as latency, availability, and reliability metrics, using Chart.js for visualizations. On the back-end, I developed RESTful APIs with Node.js and Express, connecting to a MongoDB database to store and retrieve operator data. Deploying the application with Docker introduced me to containerization, a skill I had not previously explored. Collaborating with designers, I iterated on the UI based on user feedback, ensuring the dashboard was both functional and intuitive.

The dashboard project taught me the importance of balancing technical complexity with user-centric design. For instance, presenting intricate blockchain metrics in a digestible format was challenging, but it deepened my understanding of how to communicate technical information effectively. The positive feedback from node operators, who reported improved engagement with the network, was immensely rewarding and validated the impact of my work.

5.3 Optimizing RPC Relay Services

My second major project involved optimizing back-end services for RPC relay processing, a critical component of Viper Network's infrastructure. The goal was to reduce latency and increase throughput for blockchain interactions, ensuring Web3 applications could rely on fast and efficient services. This project exposed me to the intricacies of decentralized systems and the technical challenges of load balancing in a distributed network.

I implemented routing algorithms using Node.js and Express, distributing RPC requests across nodes based on their performance scores and geographic proximity. Deploying these services with Docker and Kubernetes introduced me to orchestration, enabling scalable and resilient deployments. I also monitored performance metrics, such as Round Trip Time (RTT) and request

success rates, to validate improvements. The project resulted in a 15% reduction in latency and a 20% increase in throughput, significantly enhancing the network's reliability.

This project was particularly challenging due to the decentralized nature of the network, which required handling node failures gracefully. Learning to design fault-tolerant systems and optimize for performance in a trustless environment was a steep but rewarding learning curve. It underscored the importance of resilience and scalability in Web3 infrastructure.

5.4 Collaboration and Team Dynamics

Working at Viper Network was a collaborative endeavor, and I had the privilege of engaging

with a diverse team of developers, blockchain engineers, and product managers. Regular stand-up meetings and code reviews fostered a culture of open communication and continuous improvement. I actively sought feedback on my work, which helped me refine my coding practices and adopt best practices, such as writing modular, testable code.

Collaborating with senior developers on integrating smart contract interactions using Solidity was particularly enlightening. Their mentorship provided insights into the security considerations of blockchain development, such as preventing reentrancy attacks and ensuring data integrity. These interactions not only enhanced my technical skills but also taught me the value of mentorship and knowledge-sharing in a professional setting.

5.5 Challenges Faced and Overcoming Them

The internship presented several challenges that tested my technical abilities and resilience. One significant hurdle was adapting to the complexity of decentralized systems. Understanding concepts like session tumbling, fisherman monitoring, and the Reputation Protocol required me to dive deep into Viper Network's documentation and seek clarification from teammates. To overcome this, I dedicated time to studying blockchain fundamentals, including how RPC nodes facilitate interactions and the role of incentivization in decentralized networks.

Another challenge was managing real-time data updates in the node operator dashboard.

Ensuring low-latency updates without compromising performance demanded optimization techniques, such as implementing caching and efficient database queries. Through trial and error, and with guidance from my mentor, I successfully optimized the dashboard's performance, achieving smooth real-time functionality.

The RPC relay optimization project posed challenges in handling node heterogeneity, as nodes varied in hardware capabilities and geographic locations. Designing algorithms that equitably distributed requests while prioritizing high-performing nodes was complex. I addressed this by researching load-balancing strategies and iterating on my algorithms based on performance metrics, ultimately achieving the desired improvements.

These challenges, while daunting, were instrumental in my growth. They taught me to approach problems systematically, leverage resources effectively, and embrace a growth mindset. Each obstacle reinforced the importance of perseverance and adaptability in tackling cutting-edge technologies.

5.6 Skills and Knowledge Gained

The internship significantly expanded my technical and professional skill set, equipping me

with tools and insights essential for a career in Web3 development. Key skills acquired include:

- Front-End Development: Mastered React and Redux for building dynamic, responsive interfaces, with a focus on user experience.
- Back-End Development: Gained proficiency in Node.js and Express for developing scalable APIs, with an emphasis on performance optimization.
- Blockchain Technologies: Developed a foundational understanding of blockchain interactions, RPC protocols, and smart contract development with Solidity.
- Web3 Concepts: Gained insights into decentralized architectures, incentivization models, and QoS metrics like latency, availability, and reliability.
- Collaboration and Communication: Improved my ability to work in cross-functional teams, articulate technical ideas, and incorporate feedback.

Beyond technical skills, I developed a deeper appreciation for Web3's potential to redefine trust, privacy, and accessibility in digital systems. The internship also honed my

problem-solving abilities, as I learned to break down complex challenges into manageable tasks and iterate on solutions.

5.7 Personal and Professional Growth

The internship was a catalyst for personal and professional growth, boosting my confidence as a developer and clarifying my career aspirations. Initially, I approached the internship with uncertainty about my ability to contribute to a Web3 company. However, successfully delivering projects like the node operator dashboard and RPC relay optimization validated my capabilities and instilled a sense of accomplishment.

Professionally, I learned the importance of aligning technical work with business objectives. For instance, the dashboard project was not just about coding but about enhancing operator engagement to strengthen the network's health. This perspective shifted my focus from writing code to delivering value, a mindset I will carry forward in my career.

On a personal level, the internship taught me resilience and adaptability. Navigating the fast-paced, innovative environment of Viper Network required me to embrace ambiguity and learn continuously. Engaging with a global team exposed me to diverse perspectives, fostering cultural awareness and empathy.

5.8 Reflections on Web3 and My Role in It

Working at Viper Network deepened my understanding of Web3's transformative potential and the challenges it faces. The reliance on centralized RPC providers, as highlighted in Viper Network's documentation, underscores the need for decentralized alternatives to ensure scalability, security, and trustlessness. Contributing to a solution that addresses these issues was immensely fulfilling and reinforced my belief in the importance of decentralization.

My projects directly supported Viper Network's mission to create a resilient, multichain infrastructure. The node operator dashboard empowered users to participate in the network, while the RPC relay optimizations improved the performance of Web3 applications. These contributions, though small in the grand scheme, gave me a sense of ownership and pride in advancing the Web3 ecosystem.

Reflecting on my role, I realized that full-stack development in Web3 requires a unique blend of traditional web skills and blockchain expertise. This realization has motivated me to pursue further education in blockchain development, particularly in areas like smart contract security and decentralized application architecture.

5.9 Conclusion and Future Aspirations

My internship at Viper Network was a defining chapter in my academic and professional journey, providing me with the skills, confidence, and vision to pursue a career in Web3 development. The hands-on experience of building decentralized applications, coupled with the mentorship and collaboration I received, transformed me from a curious novice to a confident developer with a clear direction.

The challenges I faced—whether grappling with decentralized system complexities or optimizing real-time dashboards—taught me the value of perseverance, critical thinking, and continuous learning. The skills I gained, from React and Node.js to Go lang and blockchain fundamentals, have equipped me to tackle future challenges in the tech industry. Most importantly, the internship ignited a passion for Web3, inspiring me to contribute to a decentralized future where trust and accessibility are paramount.

Looking ahead, I plan to deepen my expertise in blockchain technologies, focusing on smart contract development and consensus mechanisms. I aim to contribute to open-source Web3projects and pursue advanced coursework in distributed systems. My experience at Viper Network has also motivated me to advocate for decentralization in my academic and professional communities, sharing the insights I’ve gained.

In conclusion, my internship at Viper Network was more than a professional milestone; it was a journey of discovery, growth, and inspiration. I am grateful for the opportunity to work with a visionary team and contribute to a mission that aligns with my values. This experience has laid a strong foundation for my career, and I am excited to build on it as I navigate the evolving landscape of Web3 technology.

CHAPTER 6

REFERENCES

[1] Viper Network Official Documentation: Comprehensive guide for the Viper Network blockchain platform and its functionalities.

[2] Wood, G. (2014). Ethereum Yellow Paper: Technical specification of Ethereum's decentralized transaction ledger and smart contract system.

[3] React Official Documentation: Official guide for building user interfaces with the React JavaScript library.

[4] Go Language Official Documentation: Official resource for learning and using the Go programming language.

[5] Go By Example: Practical, example-driven tutorials for learning Go programming concepts.

[6] Amino JS Documentation and GitHub: Repository and documentation for Amino, a JavaScript serialization library for Cosmos-based blockchains.

[7] Cosmos Amino JS: Official documentation for Amino JS, supporting data serialization for Cosmos SDK applications.

[8] Cosmos SDK: Modular framework for building application-specific blockchains in the Cosmos ecosystem.