Router Cross-chain Intent Framework

Shubham Singh
shubham@routerprotocol.com

Abstract

The dynamic expansion of blockchain technology has catalyzed the emergence of decentralized applications (dApps) across diverse blockchain networks, ushering in a new era of digital interaction. However, the inherent complexity involved in navigating different decentralized protocols significantly hampers user engagement and the broader adoption of blockchain applications. Furthermore, the blockchain ecosystem is highly fragmented, impeding many users from harnessing the advantages offered by applications deployed across various blockchain networks. To this end, this paper introduces the Router Cross-chain Intent Framework (CCIF), a novel paradigm designed to abstract the processes involved in executing complex workflows, such as cross-chain staking. By deriving explicit, programmable intents from ambiguous user intents, disintegrating convoluted interactions into manageable sub-interactions, and instituting a standard interface for dApps through intent adapters, this framework endeavors to streamline user-dApp interactions significantly. The proposed framework will enhance user experience, expedite the adoption of Web 3.0 applications, and facilitate application development without necessitating a deep understanding of the underlying dApp complexities. In this paper, we will cover the entire architecture of the proposed framework, the workflows involved, how it serves as a viable solution to the identified problems, and its potential impact on the wider blockchain industry.

1 Introduction

1.1 Background

The blockchain technology has been a significant disruptor in the digital realm. Its ability to facilitate trustless transactions, ensure data integrity, and eliminate intermediaries has found a number of applications, ranging from finance to supply chain management. Decentralized applications (dApps) are at the forefront of blockchain adoption, offering users control over their data and transactions. However, as the ecosystem expands, so does the complexity of interactions, particularly across different blockchain networks. Cross-chain interactions are crucial for a seamless user experience and foster a unified blockchain ecosystem. Even though the current crop of bridging solutions has made significant strides in the domain of cross-chain asset transfer, they remain insufficient in offering a holistic resolution to the broader challenge at hand.

1.2 Problem Statement

Despite a high focus on cross-chain instruction/message transfers, the existing bridge solutions have yet to deliver a user-friendly solution for intricate cross-chain workflows, posing challenges for both developers and end-users. Even now, the intricate nature of cross-chain interactions necessitates users to have a nuanced understanding of underlying protocols and procedures. This complexity is a deterrent to mainstream adoption. Furthermore, developers find it challenging to build cross-chain applications due to the complexities involved. There is a dire need for a high-level abstraction to simplify user interactions and provide a conducive environment for developers to build applications seamlessly.

1.3 Proposed Solution

The Router Cross-chain Intent Framework (CCIF) is designed as a solution in the field of blockchain technology, with a focus on enhancing the efficiency and accessibility of decentralized applications (dApps) across various blockchain networks. It enables users to submit ambiguous intents, like "Stake my USDC," which are subsequently translated into executable cross-chain workflows optimized for speed, cost-effectiveness, and reliability.

1.3.1 Significance

The significance of the proposed solution extends across multiple dimensions of the blockchain ecosystem. As a framework that simplifies cross-chain in-

teractions, it stands to substantially enhance user experience, making blockchain applications more accessible to mainstream users. The establishment of standards for intent flows and intent adapters, along with fostering a common interfacing standard among dApps, is likely to provide a conducive environment for developers to build decentralized applications seamlessly. This, in turn, can spur innovation, catalyzing a more rapid transition towards a decentralized web. Moreover, by addressing the identified challenges, the proposed framework contributes to the broader discourse on improving interoperability and ease of interaction within the blockchain domain.

1.3.2 Scope

This paper establishes the meaning of "intent" in the context of blockchain applications, followed by a brief analysis of challenges in the manual actualization of these intents within the DeFi ecosystem. Subsequently, this paper explores the architecture and functionality of the Router CCIF. Additionally, discussions around potential future enhancements for various framework components are incorporated. The study also delves into the establishment of specifications for intent flows and intent adapters to promote a common interfacing standard among dApps. Furthermore, it analyzes the impact of the framework on user experience and developer ease-of-use, alongside the broader implications of the framework on Web 3.0 adoption and the transition towards a decentralized internet.

2 Understanding Intents in Blockchain

2.1 What is an Intent?

Broadly speaking, an intent denotes an individual's purpose or goal. Applied within the context of blockchain technology, the user's intent refers to their desired outcome when engaging with blockchain networks and decentralized applications (dApps), encompassing activities such as trade execution, token staking, or asset transfer. The execution of an intent through a framework of application programming interfaces (APIs) and smart contracts results in an action. By introducing the concept of intent-based applications, we aim to abstract the intricate technical procedures underpinning these actions and prioritize achieving the user's desired end result.

2.2 Abstract Intents

Abstract intents represent the initial, often incomplete, expressions of a user's goals. For example, consider a user who says, "I want to stake USDC." While this statement reveals the user's desire to stake USDC, it lacks specifics such as the blockchain network where the user holds the USDC and the staking platform designated for the USDC staking. This lack of detail renders the intent too broad to be actionable in its current form.

2.3 Executable Intents

As the name suggests, an executable or programmable intent is one that is actionable. Extending our previous example, the user's intent can be made actionable by determining the blockchain network holding their funds (e.g., Polygon), the amount to stake (e.g., 100 USDC), the chain for staking (e.g., Ethereum). Within the realm of executable intents, we define three distinct subcategories:

1. **Stochastic Intents:** In this category, both the path to the outcome and the outcome itself are not fixed. For instance, consider the intent, "Stake 100 USDC from my Polygon wallet on a liquid staking platform on Ethereum." While this intent is actionable, the ultimate result could be stETH (Lido), ETHx (Stader), or any another token.

2. **Semi-deterministic Intents:** Here, the path to the outcome is not fixed, although the outcome itself is predetermined. For example, "Stake 100 USDC from my Polygon wallet on Lido (Ethereum)." Discounting potential transaction errors, the result of this intent will be stETH, but the specifics of the DEX used to convert USDC to ETH or the bridge for fund transfer from Polygon to Ethereum are not predetermined.

3. **Deterministic Intents:** In this subcategory, both the path and the outcome are predefined. Consider the intent, "Use Nitro to transfer my USDC from Polygon to Ethereum, then use 1inch to swap USDC to ETH, and finally, stake the ETH on Lido." In deterministic intents, every step along with the final result is explicitly specified.

One important thing to note is that an executable intent is still not sufficient to generate a concrete, executable plan that a blockchain system can understand and act upon. The transformation from an executable intent to an executable workflow involves an actualization process.

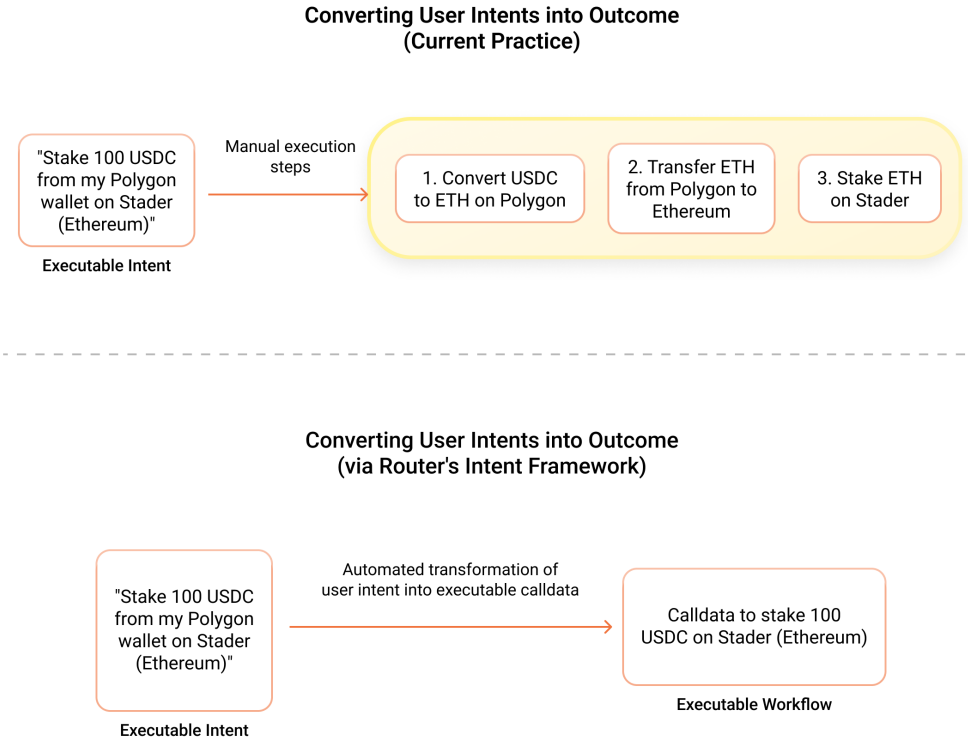

Figure 1: Manual vs Automated Actualization of Intents into Outcome

2.4 Navigating Manual Processes in Executing Intents

In the current system, the actualization of executable intents into executable workflows is a manual task. Building upon the aforementioned example, the manual actualization of the user's executable intent would involve several distinct steps. First, the user must decide where to swap USDC for the required cryptocurrency (e.g., ETH), considering factors like exchange rates and fees. If the USDC is on Polygon and the staking is on Ethereum, they must use a bridge service to transfer assets. Finally, the user navigates to the staking platform (e.g., Stader on Ethereum) to complete the staking process. This entire procedure entails interactions with multiple dApps, execution of various transactions, and engagement with different user interfaces.

The manual actualization of blockchain intents can be daunting for users. Staking in a liquid staking platform on a different blockchain typically involves around 27 clicks, interaction with 3 different dApps, 6 wallet interactions, and up to 15 minutes of effort. To accomplish a task like staking on Stader, users must be familiar with multiple blockchain networks, possess their respective gas assets, or know the processes for acquiring these assets. Furthermore, a comprehensive understanding of bridges for asset transfer and proficiency in navigating decentralized exchanges or aggregators is imperative. This intricacy and time investment significantly impede user experience and accessibility within the blockchain ecosystem.

2.5 Transitioning Towards a Streamlined Approach with Router's CCIF

Recognizing the intricacies and inefficiencies intrinsic to the existing manual process of executing blockchain intents, this paper introduces a novel paradigm: the **Router Cross-chain Intent Framework**. This framework is designed to simplify blockchain interactions via the automated actualization of user intents into precise, executable workflows. By streamlining this process, the framework aspires to enhance user experience, reduce the likelihood of manual errors, integrate robust security and efficiency standards and foster wider adoption of blockchain technology by making it more accessible and user-friendly. In the current version,

3 Framework Architecture and Working

3.1 Key Components

The current Router CCIF is primarily composed of the following:

1. **Intent Adapters:** Responsible for executing blockchain actions like swapping, bridging, or staking.
2. **Adapter Registry Module (ARM):** Maintains the standardization and integrity of intent adapters for effective usage by Intent Solvers.
3. **Intent Solvers:** Prepares and simulates call data for the executable flow, calculates slippages and fees, and determines the initial execution point.

Note: In its current version, the Router CCIF supports the actualization of semi-deterministic and deterministic user intents. We are working on introducing an Intent Resolution Module (IRM) within the framework that will allow us to support abstract and stochastic user intents. More about the IRM is mentioned in Section 5.

3.2 Intent Adapters

3.2.1 Overview

Intent adapters are smart contracts pivotal in abstracting complex actions of decentralized applications (dApps) and enabling streamlined interactions, whether within a single dApp or across multiple dApps. They serve as building blocks in constructing intuitive and efficient workflows.

Decomposing executable workflows into sub-tasks and utilizing adapters for these sub-tasks allows developers to focus on creating specialized adapters for their dApp's functionalities while leveraging generalized adapters for common tasks. This approach:

(a) reduces redundancy in development efforts, fosters reusability, and accelerates the deployment of new services and features;

(b) streamlines the integration process, making it easier for developers to contribute to the blockchain ecosystem.

3.2.2 Intent Adapter Components

To ensure secure and orderly operations, each adapter consists of the following components:

- **Head Registry:** This registry governs the initiation of the adapter, exclusively listing authorized preceding adapters. Only the adapters documented in this registry can invoke the current adapter, thereby guaranteeing regulated and secure execution sequences.
- **Tail Registry:** This registry dictates the potential succeeding actions by listing adapters that may be invoked following the current one. Just like the Head Registry, this registry ensures a secure progression of the execution flow.
- **Inbound Asset Registry:** This registry keeps track of all the acceptable incoming assets, ensuring that the adapter only processes predefined asset types.
- **Outbound Asset Registry:** It manages the types of assets that the adapter is allowed to output, thus controlling the flow's output and maintaining consistency.
- **Fee Handler Module:** A dedicated module that manages the fees associated with the utilization of the adapter. This module ensures a transparent and equitable remuneration structure for the rendered service.

3.2.3 Classification of Intent Adapters

1) Stateless Adapters: Stateless adapters are designed to execute blockchain actions that do not need knowledge of any user-specific state (e.g., past stake). A stateless adapter can efficiently execute any operation that does not require a persistent store within the contract.

2) Stateful Adapters: Unlike Stateless adapters, stateful adapters are designed to maintain and manage user states. They can handle complex interactions that require persistence of storage across different operations. These adapters can be utilized when the underlying dApp returns no proof of interaction, like a position NFT or an LP token. A good example of a stateful adapter is a farming adapter that can efficiently manage the process of depositing assets into liquidity pools of a particular protocol (say SushiSwap) and subsequently staking the received LP tokens into a yield farming platform. By ensuring that only authorized users can withdraw their assets, a farming adapter can maintain a high level of safety in its operations.

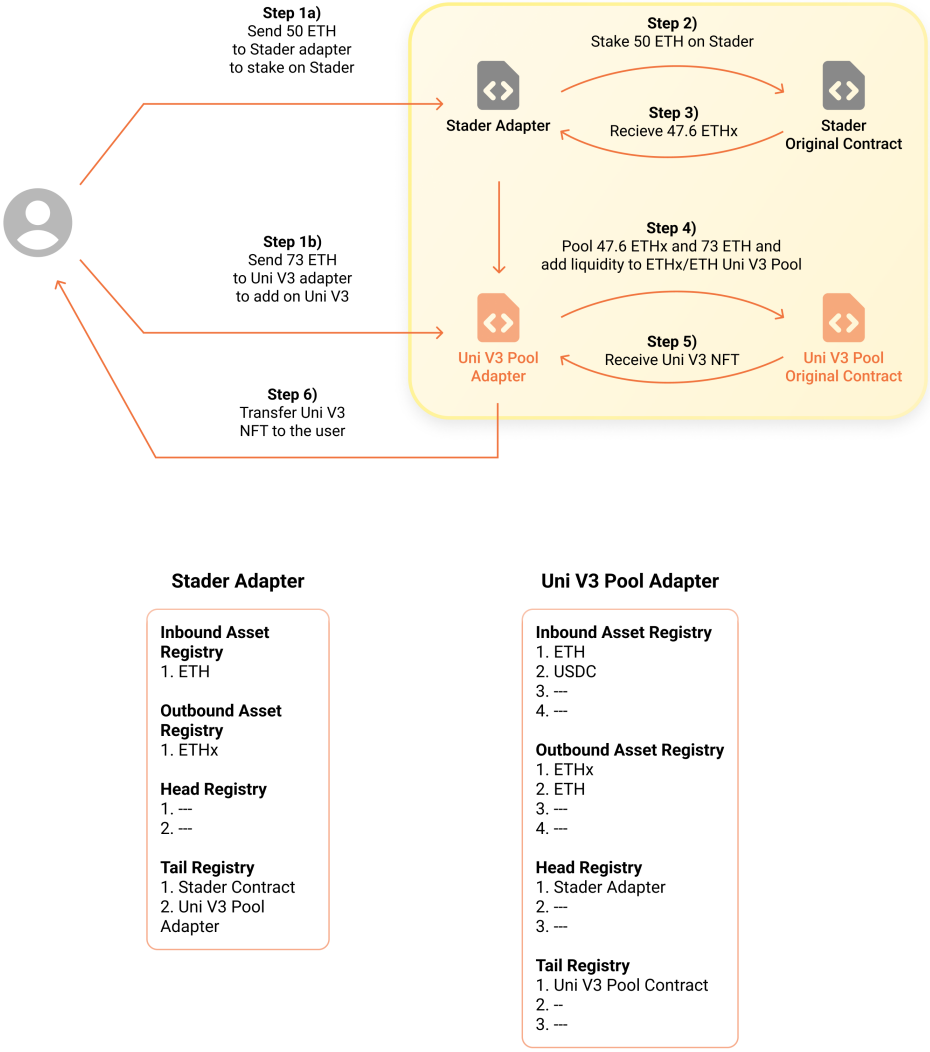

Figure 2: Adapter Specifications

3.3 Adapter Registry Module

3.3.1 Overview

The Adapter Registry Module (ARM) is a Web 2.0 module that serves as a repository for all the intent adapters, which developers can use to track and update their adapter's functionalities and constraints. It standardizes the way adapters are specified, utilized, and managed within the framework. This standardization is essential for seamless interaction between different components of the framework and the reliable execution of user intents at any given time.

3.3.2 Adapter Specifications

Each adapter within the ARM is uniquely defined by the following specifications:

- **adapter_id:** A unique identifier generated as a hash of the `chain_id` and `adapter_address`. This serves as the primary identifier for the adapter within the network.
- **adapter_description:** This field describes the function of the adapter and outlines how it interacts within the intent framework. It provides vital information about the adapter's operational mechanics, its interaction with blockchain elements, and its input and output processes. This

description helps users and developers understand the adapter's role in the overall execution flow.

- **adapter_function_selectors:** An array of function identifiers, each a hash of the `adapter_id` and the specific function selector. This array ensures unique and collision-free identification of each function within the adapter.
- **adapter_function_descriptors:** A mapping between the adapter function selectors and their descriptions, providing detailed insight into the behavior and purpose of each function.
- **whitelisted_head_set:** A set of `adapter_ids` that are allowed to precede an adapter, maintaining the integrity and order of the execution flow.
- **whitelisted_tail_set:** A set of `adapter_ids` that are allowed to succeed an adapter in an intent execution sequence, ensuring controlled and secure flow of execution.
- **chain_id:** The identifier of the blockchain network where the adapter is deployed, essential for cross-chain functionality and recognition.
- **inbound_asset_set:** A list of assets that an adapter is capable of receiving. This list specifies the adapter's scope of input handling capabilities.
- **outbound_asset_set:** A list of assets that an adapter can output or transfer. This list outlines the adapter's capacity in terms of deliverable assets.
- **adapter_type:** Specifies whether the adapter is stateless or stateful, indicating its capability to maintain state or execute discrete actions without retaining user-specific states.
- **tags:** A list of descriptive tags for the adapter, such as 'staking', 'lido', 'dex', 'swap', among others. This list aids in classification and easier identification within the framework.

3.3.3 Adapter Sync Module

The Adapter Sync Module plays a crucial role in maintaining the integrity and consistency of the Adapter Registry Module (ARM). Its primary functions include:

- **Verification Check:** Confirming the verification status of each adapter on its respective blockchain, thereby ensuring its authenticity.
- **ABI Synchronization:** Extracting and storing the adapter's ABI (Application Binary Interface) from the blockchain, providing a reliable interface for interaction.
- **Function Selector Validation:** Cross-referencing the adapter's declared function selectors with those available in the ABI, ensuring that the adapter's functionality aligns with its on-chain implementation.
- **Default Function Handler Generation:** Creating baseline function handlers for each declared function selector to streamline the integration process.

This module ensures that each adapter in the ARM not only meets the required specifications but also remains in sync with its on-chain version, upholding the framework's robust standards for security and reliability.

3.4 Intent Solver

3.4.1 Overview

Intent Solver is a crucial part of Router's Cross-chain Intent Framework Module. Its primary function is to find and scrutinize potential execution paths for the executable. After identifying potential execution paths along with the requisite inputs for the head adapter in each path, the Intent Solver employs advanced multi-criteria decision making algorithms to discern the most optimal path. Its responsibilities also encompass the generation and optimization of call data for each adapter in the selected path to ensure a seamless transaction process. Upon determination of the optimal path and preparation of the call data, the Intent Solver communicates the entry point contract address, along with the call data, back to the user. This step empowers the user to initiate the transaction with a well-defined execution path.

3.4.2 Intent Solver Components

The Intent Solver's effectiveness is anchored by its key components: Pathfinder, Simulator, and Calldata Composer. Pathfinder is responsible for determining the most optimal path based on different criteria. The Simulator performs real-time transaction simulation to ensure the feasibility of the paths prepared by the Pathfinder. The Calldata Composer assembles the necessary transaction data for the selected path. In synergy, these components collectively propel the Intent Solver towards efficient execution of user intents on the blockchain.

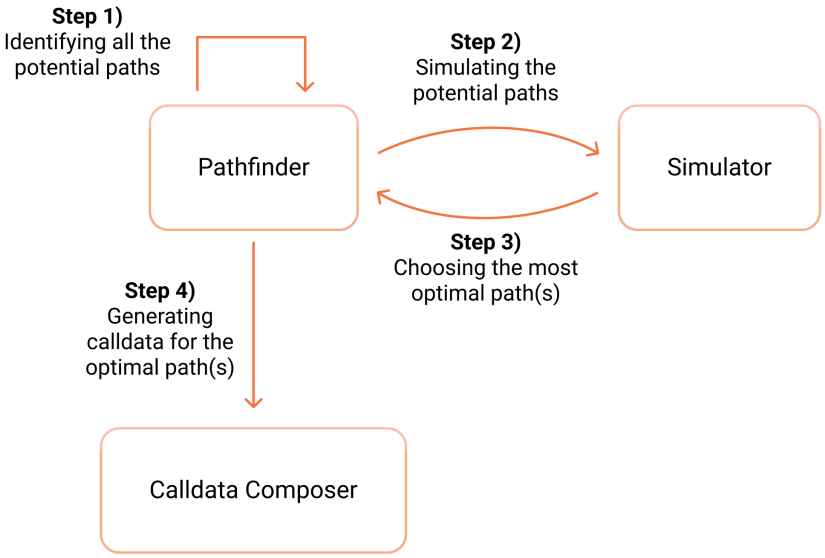

Figure 3: Intent Solver Workflow

1) Pathfinder: The Pathfinder is an algorithm that identifies potential adapters from the Adapter Registry Module to execute the user's intent and proposes executable paths. Leveraging advanced multi-criteria decision making techniques, it also evaluates these paths, giving precedence to factors such as cost-effectiveness, time efficiency, and the maximization of potential yields. This evaluation ensures that the chosen path balances cost savings and low-latency execution. In essence, the Pathfinder's role is pivotal in steering the user's intent towards execution in the most optimal manner, making it a key player in the decision-making process within the Intent Solver framework.

2) Simulator: As the name suggests, the Simulator is responsible for simulating transactions pertaining to each adapter in the proposed path. This component's primary responsibility is to guarantee the proper operation of every adapter in the proposed sequence. In instances where an adapter fails to yield a successful simulation, the Simulator dismisses that particular path. Besides being a verifier of adapter functionality, the Simulator also acts as a source for data-driven path optimization.

3) Calldata Composer: The Calldata Composer is responsible for preparing calldata for each adapter in a chosen path and synthesizing them into a unified calldata payload. This unified payload embodies the execution blueprint of the user's intent, as determined by the optimal path identified by the Pathfinder. It is this aggregated calldata payload that the user ultimately broadcasts on the blockchain to actualize their intent, making the Calldata Composer a key architect in the process of transforming user intents into actual blockchain transactions.

3.4.3 Intent Solver Workflow

This section delves into Intent Solver's operational workflow. To illustrate the intricate interplay between Intent Solver's components, let us consider the following executable intent: "Stake 100 USDC from Polygon to Stader on Ethereum."

Step 1) Recognizing all the Potential Paths The Pathfinder scans the ARM for potential adapters that can be used to execute the user intent and identifies all the potential paths to achieve user's desired output.

- **Potential paths proposed by the IRM:**
  1. `funds` → `DEX (Polygon)` → `Bridge 1` → `Stader (Ethereum)`
  2. `funds` → `Bridge 1` → `DEX (Ethereum)` → `Stader (Ethereum)`
  3. `funds` → `DEX (Polygon)` → `Bridge 2` → `Stader (Ethereum)`

Note: In case of deterministic intents, constraints can be placed on the Pathfinder to allow for only one execution path.

Step 2) Simulating the Potential Paths The Pathfinder engages the Simulator to assess all of the identified paths based on many factors, including execution cost, time taken for transaction completion, and the potential yield each path offers. For example, Path 2 requires 230K gas for execution, a duration of 6 minutes, and an estimated outcome of 0.13 ETHx. The Simulator's capability to identify and reject non-viable paths is a critical part of this process. For instance, if the Simulator encounters a failure during the simulation of `DEX (Ethereum)` adapter in Path 2, the Pathfinder promptly eliminates this path from consideration, ensuring that only viable paths are retained for further analysis.

Step 3) Choosing the Most Optimal Path(s) Using a custom A* algorithm, Pathfinder assesses each potential path in terms of its execution cost, time, and estimated outcome to calculate a composite score for each. Let us say that Path 3 (180K gas, 10 minutes, 0.136 ETHx) is identified as the best for net yield, but Path 2 (120K gas, 2 minutes, 0.134 ETHx) emerges as the recommended choice by striking a good balance for speed, cost and estimated outcome.

Step 4) Composing the Calldata Once one or more optimal routes are decided, the Calldata Composer creates a unified calldata payload by assembling the calldata for each adapter in the respective paths. Once the calldata is generated, the Intent Sovler returns it to the user along with the entry point contract address.

3.4.4 Challenges and Solutions

The Intent Solver faces several challenges in its operation. Addressing these challenges is crucial for enhancing its efficiency and reliability.

Challenge 1: Bridge Adapter Time Estimation Estimating the time for cross-chain communications through bridge adapters is inherently non-deterministic and can vary significantly.
Potential Solution: Implementing heuristic approaches to gauge execution time for bridge adapters can provide more reliable time estimates. Historical data and average time metrics can also be utilized for better prediction.

Challenge 2: External RPC Dependency The reliance on external RPC (Remote Procedure Call) services for blockchain data might cause problems, especially when these services experience network errors. RPC issues might lead to the rejection of valid paths.
Potential Solution: Integrating fallback mechanisms for RPC services can mitigate this risk. In case of network failures, the system could automatically switch to alternative RPC sources. Additionally, setting up dedicated full nodes, rather than relying on third-party RPC services, could provide more stability and control over data retrieval processes.

4 Use Cases

This section explores diverse use cases of Router's Cross-chain Intent Framework, demonstrating how it addresses challenges in blockchain adoption. Each example illustrates the framework's practicality, adaptability, and potential to enhance efficiency in blockchain interactions across various types of DeFi applications.

4.1 Cross-chain DeFi Applications

Router's Cross-chain Intent Framework can be used to create cross-chain DeFi applications including but not limited to cross-chain liquid staking, cross-chain lending/borrowing, and cross-chain liquidity position manager among others. Such applications allow users to put their idle funds to good use by allowing them to stake their funds in a platform of their choice, extending beyond the blockchain where their funds are situated initially. Consider the scenario of a user holding USDC on the Polygon network who wishes to stake on Benqi (a liquid staking platform on the Avalanche blockchain). Once the user states their intent to perform this action, the framework estimates and outputs a seamless execution flow. Initially, a bridge adapter facilitates the secure transfer of USDC from Polygon to Avalanche. Subsequently, a DEX adapter is used to convert the user's USDC to AVAX. Finally, a Benqi Intent Adapter is employed to stake the AVAX tokens on Benqi's smart contracts within the Avalanche network. This use case exemplifies the versatility of Router's intent framework, allowing users to capitalize on different DeFi opportunities across disparate chains.

4.2 Cross-chain NFT Marketplace

Router's Cross-chain Intent Framework can be used to enhance the accessibility of NFT marketplaces by

allowing users to bid and buy NFTs from a separate chain altogether. Imagine a user possessing an Ethereum-based NFT who wishes to trade it on a Polygon NFT marketplace. Leveraging Router's intent framework, this process unfolds as follows. The Ethereum-based NFT is transferred from Ethereum to Polygon via a bridge adapter. Once the NFT is bridged, a dedicated adapter to handle NFT platform interactions is engaged to facilitate the exchange or sale of the NFT on the Polygon marketplace. This use-case underscores the system's proficiency in managing non-fungible token transfers and facilitating interactions with diverse NFT platforms.

4.3 Automated Portfolio Rebalancing

Router's Cross-chain Intent Framework presents substantial benefits for a portfolio management dApp. Using the Intent Solver, the dApp can strategically identify and capitalize on platforms offering the highest yields. Following this, stateful portfolio management adapters can be used to dynamically manage user assets and optimize their yields. The inherent capability of the framework to facilitate such dynamic strategies demonstrates its relevance for stateful applications like portfolio management dApps.

5 Future Work

As mentioned above, the current iteration of Router's Cross-chain Intent Framework can only work on executable intents if they are semi-deterministic or deterministic. To enable support for abstract intents and stochastic executable intents, we are working on a dedicated Intent Resolution Module (IRM), a system based on a large language model (LLM) that converts abstract user intents into programmable ones. By prompting the user for clarifications and additional information, the IRM delineates a semi-deterministic (or deterministic) intent to achieve the desired result. Additionally, the IRM will reduce the load on Intent Solvers by using the Adapter Registry Module (ARM) data to understand each adapter's behavior and arrange their positions to create a direct pathway for executing user intents.

6 Industry Impact

The proposed framework stands to significantly impact the blockchain industry in various ways:

6.1 Simplified Blockchain Interactions

Users often struggle with managing different wallets and understanding different blockchain ecosystems. This complexity can be overwhelming and hinder efficient blockchain interaction. By providing an intuitive interface and a streamlined process, Router's Intent Framework makes it easier for users to interact with different applications, eliminating the need for in-depth knowledge of each chain or dApp. This simplification has the potential to widen the technology's appeal and increase its accessibility to a broader audience.

6.2 Improved DeFi Accessibility

By facilitating seamless cross-chain transactions and interactions, the framework encourages the adoption and development of cross-chain solutions. The framework's ability to integrate services across different blockchains could play a pivotal role in consolidating the currently fragmented blockchain ecosystems, leading to greater interoperability and functionality.

6.3 Improved Contract Reusability

The framework represents an advancement in making smart contracts more user-friendly and adaptable. This could open up new possibilities for diverse applications and use cases, making smart contracts more accessible and appealing to a wider user base.

6.4 Simplified dApp Development

Developing new applications on blockchain platforms presents significant complexity and technical barriers, particularly for new developers. To combat this issue, Router's Intent framework offers streamlined tools and pathways for creating derived applications. This reduces the technical challenges and complexities involved in blockchain integration, making it more feasible for businesses to develop and deploy blockchain-based solutions efficiently.

7 Efficiency and Environmental Impact

7.1 Gas Efficiency

Every transaction not sent to the blockchain translates to a direct saving of 21000 gas units. If we were to batch three transactions into one, it would result in a cumulative reduction of 42000 gas units. Furthermore, the strategic batching of transactions has

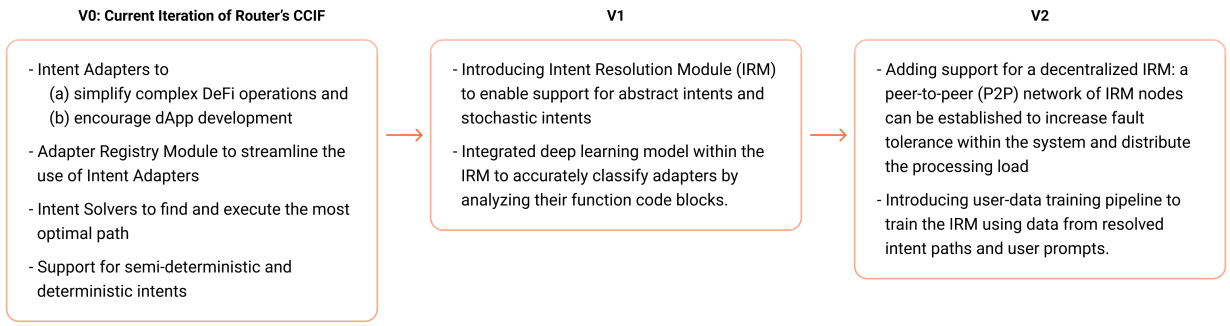

Figure 4: CCIF Product Roadmap

the potential to decrease the number of individual operations required, offering further gas optimization.

7.2 Mempool Optimization

Batching multiple transactions into one frees up mempool space, allowing more transactions to be a part of the mempool, significantly enhancing mempool efficiency. Furthermore, by reducing pressure on the mempool, the framework promotes a more streamlined inclusion process, lowering the competition for transaction inclusion and potentially moderating gas prices.

7.3 Time Efficiency

By consolidating multiple transactions into a single batch, the framework expedites workflow completion. Even after discounting the time for manual operations, the existing system will require a much longer time for completion in contrast to the completion time required for the same workflow via Router's Intent Framework. The existing system follows a sequential process: once the initial transaction is mined, the subsequent transaction enters the mempool. There, it undergoes validation, dissemination to peering nodes, and eventual inclusion in a block. This sequence repeats for each transaction in the workflow. Conversely, with a Router Intent transaction, only a single mining operation is necessitated, resulting in substantial time savings.

7.4 Reducing Carbon Footprint

Executing multi-step workflows using Router's Intent Framework can have significant ecological implications. According to an EY report, a solitary Ethereum transaction generates an emission of 0.01 kg CO2, equivalent to the environmental impact of 22 VISA transactions or two hours of YouTube viewership. Therefore, a reduction of 200,000 transactions could potentially mitigate emissions by 2000 kg CO2, corresponding to the environmental equivalent of four million hours of YouTube or 44 million VISA transactions.

8 Future Potential

The proposed framework represents more than a technical advancement; it is a stepping stone towards a fundamentally decentralized internet where user control and data autonomy are paramount. This vision encompasses several key aspects:

8.1 Universal Blockchain Accessibility

One of the framework's core goals is to create a seamless, interconnected blockchain environment. This means bridging the gaps between various chains, thus enabling smoother, more universal access to blockchain technology.

8.2 Decentralized Application Innovation

The framework is expected to catalyze a wave of innovation in decentralized applications across various sectors. This means new services and applications emphasizing user convenience, sovereignty, and decentralization.

8.3 Integration with Emerging Technologies

The framework is not static; it is designed to evolve and integrate with cutting-edge technologies like AI

and IoT. This integration could revolutionize a myriad of applications, from smart cities to personalized healthcare.

Conclusion

The Router Cross-chain Intent Framework marks a key innovation in the blockchain domain, offering an effective solution to the complexities of multi-chain interactions. Its distinct components – Intent Adapters, Adapter Registry Module, Intent Resolution Module, and Intent Solver – collaboratively ensure efficient and seamless cross-chain functionality. The framework remains open to contributions from any party, allowing for the deployment of custom intent adap ters by individuals seeking to enhance its capabilities. This framework's implications are far-reaching, potentially revolutionizing how users and developers interact with blockchain technologies. By simplifying cross-chain transactions and reducing the technical barrier to entry, it paves the way for broader blockchain adoption and novel application development. Future enhancements could involve algorithmic refinements and expansion into new application domains. This ongoing development will foster a more accessible, user-friendly, and versatile blockchain ecosystem, contributing significantly to the evolution of decentralized systems and digital interactions.

Acknowledgement

I would like to thank Vatsal Gupta, Abhishek Somani, Shivansh, Kunal Kotiyal for their contribution on the architecture and Aditi Shastry for her help in designing the figures. I would also like to thank Anish Mohammed, Kimberly Adams, and Zaki Manian for their valuable inputs that helped form the technical design of this framework.

References

[1] The Landscape of Blockchain Research: Impacts and Opportunities. *Springer*. https://link.springer.com.

[2] A Systematic Literature Review of Blockchain-Based Applications. *ScienceDirect*. https://www.sciencedirect.com.

[3] Research Papers by Blockchain@UBC. *Blockchain@UBC*. https://blockchain.ubc.ca.

[4] Blockchain for Development: A Guiding Framework. *Taylor & Francis Online*. https://www.tandfonline.com.

[5] A Look into the Future of Blockchain Technology. *PLOS ONE*. https://journals.plos.org.

[6] Toward More Rigorous Blockchain Research: Recommendations. *Frontiers*. https://www.frontiersin.org.

[7] Account Abstraction, Analysed. *arXiv.org*. https://arxiv.org.

[8] About Blockchain Interoperability. *ResearchGate*. https://www.researchgate.net.

[9] A Survey on Blockchain Interoperability: Past, Present, and Future. *ResearchGate*. https://www.researchgate.net.

[10] Interoperability Among Heterogeneous Blockchains: A Systematic... *Springer*. https://link.springer.com.

[11] Interoperability in Blockchain: A Survey. *IEEE Xplore*. https://ieeexplore.ieee.org.

[12] Inter-Blockchain Communication in Cosmos Blockchain Network. *MDPI*. https://www.mdpi.com.

[13] Modeling Cross-Blockchain Process Using Queueing Theory: The Case of Cosmos. *IEEE Xplore*. https://ieeexplore.ieee.org.

[14] How does the Ethereum Merge help the real and virtual world save energy? *EY* https://www.ey.com/en_ch/technology/how-does-the-ethereum-merge-help-the-real-and-virtual-world-save-energy