Nimble Case Study - The Marketplace of Intents

A Permissionless and Extensible Protocol for an Intent Economy

October 10, 2023

Abstract

Today's blockchain systems are obscure. Users of dApps must perform low-level operations to complete basic tasks. Developers seeking to launch a new dApp must learn a new technical stack. Such system complexity is preventing mainstream adoption of Web3 technologies by users and developers alike.

This paper presents a Nimble case study, the *ecosystem's first* marketplace of intents. The protocol solves this notorious problem while improving Web3 composability and decentralization. User interactions with dApps (e.g. GUI clicks and natural language queries) are modeled using intents. Intents are mapped to a standard DSL using LLMs and other intent understanding techniques. Dispatchers route intent DSL to specialized solvers, which execute the instructions on-chain. Solvers compete for intents, minimizing harmful MEV. The protocol is open, decentralized, and extensible.

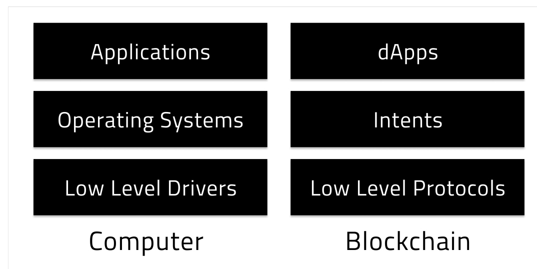

Figure 1: The protocol abstracts low-level blockchain operations away from developers, serving as an operating system for Web3. It is an abstraction layer on top of L1s & L2s.

1 Introduction

Today's blockchain user experience is comparable to the early days of computing. Computer users had to understand hardware configurations, install drivers, and interact with applications through command lines or minimal user interfaces. Modern Web3 users face similar technical overhead. Low-level operations like transactions, swaps, and bridging are part of the basic vocabulary of Web3 users. Developers cannot build platform-agnostic applications, which forces users to understand the differences between blockchains, possess multiple wallets, and manually track their assets.

The protocol's vision is to enable a marketplace of intents with a large action model built on top of the Nimble AI protocol. As a Web3 portal protocol, it sits on top of blockchains, L2s, and bridges. It hides the underlying complexity of blockchain systems, providing a unified interface for all compatible Web3 systems. Developers interact with the large action model directly without the need to understand fundamental operations. Such developers can easily build cross-platform dApps, and low-level operations can be completely hidden from end users.

The intent marketplace consists of the interpretation, dispatching and settlement markets. Dispatching routes intents to the correct solvers. Settlement markets are for intent executions with a modular auction layer. Both markets can leverage existing network design of the Nimble AI protocol, while the interpretation market is built as AI models on the Nimble network.

- **The Interpretation Market.** The intent recognizer accepts general format intents and translates them into concrete intent operations. Intents are represented as an intent taxonomy forest for ontology. Each intent category (e.g., trade and token) is an independent tree defined by a standard DSL. Specific intent operations are leaves on the trees. LLMs and rule engines are adopted for natural language intent understanding (i.e., map user intents into tree leaves).

Section 2 discusses related work and existing intent solutions. Section 3 details the marketplace infrastructure design. Conclusions follow this in Section 4.

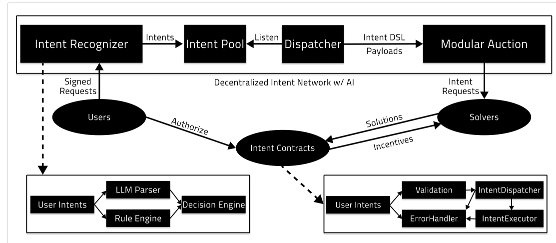

Figure 2: The intent protocol consists of three essential modules that work together seamlessly to facilitate decentralized user interactions with blockchain technologies. Intent recognizer understands user intents as machine-readable intent operations. Intent operations are published to intent pools. Such pools are a critical component of the intent dispatcher. The intent dispatcher runs a second-price auction for truthful solver bids. Solvers interact with the auction protocol within the dispatcher for intent competitions. Intent contracts provide permissionless onchain execution.

2 Intents in Web3

The protocol is the first to focus on building a trustless marketplace infrastructure for Web3 intent applications. Anoma builds a privacy blockchain with zero knowledge proofs [3]. This is a significant departure from intents. SUAVE by Flashbots focuses on MEV decentralization with a trusted execution environment (TEE) for privacy via SGX hardware support. However, a) SGX can be hacked [14]; and b) SGX hardware is highly specialized and difficult to acquire as the network scales [18]. Cowswap, 1inch Fusion, and UniswapX provide dApps which focus on trading [1, 4, 19]. Essential is building an intent contract standard [5] emphasizing intent contracts.

3 The Marketplace Infrastructure

The protocol's marketplace infrastructure consists of interpretation, dispatching, and settlement markets. Since dispatching and settlement markets are built in functionalities in the Nimble AI protocol. The interpretation market is detailed in this section.

3.1 The Interpretation Market

The Intent recognizer serves as the gateway to the Nimble protocol, acting as the interpreter of user intentions. Its primary function is to bridge the gap between the user's natural language expressions and the protocol's machine-readable understanding. Leveraging Natural Language Processing (NLP) techniques for entity recognition, this layer deciphers user input, extracting the essence of their blockchain-related requests together with rule-based approaches. It translates these intents into a structured format, such as a list of user operations, using an intuitive Intent Query-Based DSL [6,9,10,13,17]. This layer is essential in ensuring that user interactions are comprehensible to the subsequent layers, creating a seamless and user-friendly entry point to the world of Web3.

- This foundational layer serves as the entry point for user interactions with Web3 components.

- It utilizes Natural Language Processing (NLP) algorithms and smart contract interfaces to decipher and extract user intent from textual or voice-based inputs.

- Within the Intent Recognizer, a key component is the utilization of an Intent Query-Based DSL, which bridges users' natural language intents and machine-readable operations.

- This DSL allows users to express their blockchain-related intentions in a structured and declarative manner, resembling SQL or YAML for ease of use.
- Users can formulate complex queries or declarative statements that specify the desired blockchain actions, parameters, and conditions.
- It abstracts the complexities of blockchain interactions into intuitive and human-readable syntax, including smart contract interactions, token transfers, staking, and more.
- It employs advanced parsing and pattern recognition techniques to translate DSL statements into a list of user operations comprehensible to the protocol.

Intent Representation. Intents are outcome-driven messages signed by users. Intents are desired user-specified states rather than instructions like transactions. They focus on user preferences, the product journey, and the desired protocol outcomes. For extensibility, intents are standardized in both the definitions and understanding. Example intents are:

- Stake ETH for best yields,
- Trade USDC for ETH at the optimal price, and
- Swap USDC for ETH with minimum slippage of 0.1%.

To provide unambiguous definitions, tree-based ontology is adopted. Ontology, as a branch of philosophy, is the science of what is, of the kinds and structures of objects. In simple terms, ontology seeks the classification and explanation of entities.

The intent taxonomy is a forest formed by a set of trees. In computer science, a tree is a widely used abstract data type representing a hierarchical structure with a set of connected nodes. Each node in the tree can be connected to many children (depending on the type of tree) but must be connected to exactly one parent, except for the root node, which has no parent (i.e., the root node as the top-most node in the tree hierarchy).

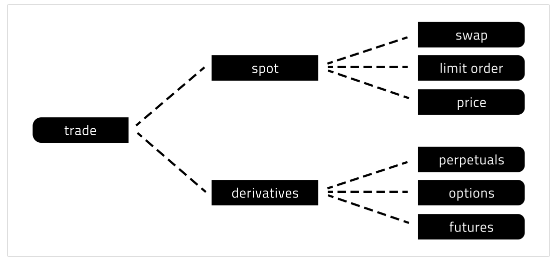

Figure 3: An intent taxonomy is a forest of intent trees each of which represents an independent intent category.

Each tree defines an independent intent category. The root is the largest concept for that category. The leaf is a particular operation defined by DSL. The forest is extensible by adding new trees for new operation categories with the evolution of the Web3 concepts. Trees can be extended by adding new operations. In the example above, new derivative types can be added constantly as nodes and leaves.

The forest and leaf operations can be defined as a set of configuration files or specific code languages. The specific implementation does not matter too much, only if it is accurate. For example, operations can be defined as EVM, Move, or Rust smart contract functions, while the forest is defined as configurations. An alternative can be defined as contract programs. As a result, our definition of intent is extensible, accurate, and flexible.

Intent Understanding. The outcome of intent recognizers is intent operations. Each intent operation is a leaf on a taxonomy tree. Before deep-diving intent understanding, the specification of particular intent operations is discussed. A Domain Specific Language (DSL) is a programming language with a higher level of abstraction optimized for a specific class of problems. A DSL uses the concepts and rules from the field or domain. In Web3, different intents are defined in a structured manner as a configuration or a piece of code. Below is a simple example intent *swap 10 BTC for USDT at maximum* 0.5% *slippage instantly*.

```
intentType: swap
  from: BTC
  to: USDT
  slippageThreshold: 0.5
  amount: 10
  delay: 0
```

Despite the extensible, accurate, and flexible definitions provided by DSL, end users should not be expected to learn technical languages before using dApps. Additionally, without sophisticated knowledge of blockchain infrastructure, users may submit unsupported or irrelevant intents. Thus, developers structure intents and present them to users as familiar interactive elements like buttons, drop-down menus, etc. Once intents are submitted to the network, intent understanding aims to parse the arbitrary user inputs into specific leaf nodes in our intent taxonomy.

The intent recognizer module combines LLMs and a rules engine as illustrated in Fig. 2. There is a wide literature on this topic. Basically, rule engines define specific patterns to parse user input queries, while LLMs are a large model of neural networks used for pattern identification.

- **Rule Engine.** Rules engines define a set of patterns to process the intents and map intents to leaf nodes in the taxonomy forest (i.e., specific intent operations).
- **LLM Parser.** LLMs are used for intent natural language understanding with named entity tagging by identifying salient entities, followed by specific operation classifications.
- **DSL Mapper.** The DSL mapper combines the above parsing results for DSL formulation and intent precision boosting.

The Intent Execution Layer serves as the engine that transforms user intentions into tangible blockchain actions, and it introduces an innovative element known as the Solver Network to provide extensibility. Within this layer, the protocol's primary function is to leverage the Solver Network - a decentralized group of nodes specializing in different dApp and blockchain operations - to solve user's intents while introducing competition at every stage.

Let us use the interpretation market consensus as a case study. Network validators own one single interpretation model on the Nimble AI protocol, as standardized by intent DSL. It of different model stages such as spelling corrections, query segmentation, query expansion, named entity tagging and structured intent DSL mapping. At the inference stage, validators apply weighted voting for the binary classification example. For the initial launch, the network only decentralizes inferences and updates models with a voting governance mechanism by evaluating the model performance against a predefined validation intent set. In the Nimble AI network, the training can also be decentralized with decentralized learning [11,12,16] and the model parameters are compressed for efficient network transmission [2,7,8,15,20,21].

4 Conclusion

In conclusion, the *intent marketplace infrastructure* we are diligently crafting represents a pivotal leap forward in the realm of Web3 and blockchain technology. This innovative framework bridges the communication chasm between users and decentralized networks, offering a seamless and intuitive means to articulate complex intentions within this intricate digital landscape. By encapsulating the essence of human intent within a structured query language, we empower individuals to navigate the Web3 ecosystem with unparalleled ease and precision.

The primary objective is to significantly lower the bar for users to enter the Web3 world. Our intent protocol paves the way for a harmonious collaboration between human cognition and the decentralized world through natural language processing, LLMs, and blockchain expertise. It promises a future where blockchain interactions are as effortless as conversing in plain language, all while upholding the security and integrity of blockchain transactions.

With the intent marketplace solution built on top of the Nimble AI protocol, the Web3 world ventures into uncharted territory, shaping the future of Web3 by democratizing access to the blockchain. This groundbreaking initiative is a testament to our commitment to enhancing user experiences, catalyzing innovation, and ultimately transforming how we interact with the digital frontier. As we continue to refine and expand upon this vision, we invite you to join us on this remarkable journey into the future of Web3. Together, we shall redefine the possibilities of human-machine collaboration in the blockchain era, making the Web3 world more accessible and inclusive than ever before.

References

[1] 1inch. 1inch fusion: A new dex standard. safe. efficient. gasless execution., 2023.

[2] Zhuotong Chen, Qianxiao Li, and Zheng Zhang. Self-healing robust neural networks via closed-loop control. *arXiv preprint arXiv:2206.12963*, 2022.

[3] Coindesk. Anoma foundation plans to launch privacy-focused namada blockchain, 2023.

[4] Cowswap. Cowswap docs, 2023.

[5] Essential. Asset based intent standard: An intent standard for the world of digital assets, 2023.

[6] Martin Fowler. *Domain-specific languages*. Pearson Education, 2010.

[7] Cole Hawkins, Xing Liu, and Zheng Zhang. Towards compact neural networks via end-to-end training: A bayesian tensor approach with automatic rank determination. *SIAM Journal on Mathematics of Data Science*, 4(1):46–71, 2022.

[8] Cole Hawkins and Zheng Zhang. Bayesian tensorized neural networks with automatic rank selection. *Neurocomputing*, 453:172–180, 2021.

[9] Paul Hudak. Domain-specific languages. *Handbook of programming languages*, 3(39-60):21, 1997.

[10] Tomaž Kosar, Pablo E Martı, Pablo A Barrientos, Marjan Mernik, et al. A preliminary study on various implementation approaches of domain-specific language. *Information and software technology*, 50(5):390–405, 2008.

[11] Anusha Lalitha, Shubhanshu Shekhar, Tara Javidi, and Farinaz Koushanfar. Fully decentralized federated learning. In *Third workshop on Bayesian Deep Learning (NeurIPS)*, 2018.

[12] Tian Li, Anit Kumar Sahu, Ameet Talwalkar, and Virginia Smith. Federated learning: Challenges, methods, and future directions. *IEEE Signal Processing Magazine*, 37(3):50–60, 2020.

[13] Marjan Mernik, Jan Heering, and Anthony M Sloane. When and how to develop domain-specific languages. *ACM computing surveys (CSUR)*, 37(4):316–344, 2005.

[14] Kit Murdock, David Oswald, Flavio D Garcia, Jo Van Bulck, Daniel Gruss, and Frank Piessens. Plundervolt: Software-based fault injection attacks against intel sgx. In *2020 IEEE Symposium on Security and Privacy (SP)*, pages 1466–1482. IEEE, 2020.

[15] Yuji Roh, Kangwook Lee, Steven Whang, and Changho Suh. Sample selection for fair and robust training. *Advances in Neural Information Processing Systems*, 34:815–827, 2021.

[16] Felix Sattler, Simon Wiedemann, Klaus-Robert Müller, and Wojciech Samek. Robust and communication-efficient federated learning from non-iid data. *IEEE*

transactions on neural networks and learning systems, 31(9):3400–3413, 2019.

[17] Bran Selic. A systematic approach to domain-specific language design using uml. In *10th IEEE International Symposium on Object and Component-Oriented Real-Time Distributed Computing (ISORC'07)*, pages 2–9. IEEE, 2007.

[18] Kristoffer Myrseth Severinsen. Secure programming with intel sgx and novel applications. Master's thesis, 2017.

[19] Uniswap. Introducing the uniswapx protocol, 2023.

[20] Zi Yang, Junnan Shan, and Zheng Zhang. Hardware-efficient mixed-precision CP tensor decomposition. *arXiv preprint arXiv:2209.04003*, 2022.

[21] Kaiqi Zhang, Cole Hawkins, Xiyuan Zhang, Cong Hao, and Zheng Zhang. On-FPGA training with ultra memory reduction: A low-precision tensor method. *arXiv preprint arXiv:2104.03420*, 2021.

